A Constructive Approach to Constrained Hexahedral Mesh Generation

Carlos D. Carbonera[1] and Jason F. Shepherd[2,3]

[1] Gauss Research Laboratory, University of Puerto Rico, Rio Piedras, PR carbonera@uprr.pr
[2] Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT jfsheph@sci.utah.edu
[3] Computational Modeling Sciences Dept., Sandia National Laboratories, Albuquerque, NM jfsheph@sandia.gov

S. Mitchell proved that a necessary and sufficient condition for the existence of a topological hexahedral mesh constrained to a quadrilateral mesh on the sphere is that the constraining quadrilateral mesh contains an even number of elements. S. Mitchell's proof depends on S. Smale's theorem on the regularity of curves on compact manifolds.

Although the question of the existence of constrained hexahedral meshes has been solved, the known solution is not easily programmable; indeed, there are cases, such as Schneider's pyramid, that are not easily solved.

D. Eppstein later utilized portions of S. Mitchell's existence proof to demonstrate that hexahedral mesh generation has linear complexity. In this paper, we demonstrate a constructive proof to the existence theorem for the sphere, as well as assign an upper-bound to the constant of the linear term in the asymptotic complexity measure provided by D. Eppstein.

Our construction generates $76{*}n$ hexahedra elements within the solid where n is the number of quadrilaterals on the boundary. The construction presented is used to solve some open problems posed by R. Schneiders and D. Eppstein. We will also use the results provided in this paper, in conjunction with S. Mitchell's Geode-Template, to create an alternative way of creating a constrained hexahedral mesh. The construction utilizing the Geode-Template requires 130*n hexahedra, but will have fewer topological irregularities in the final mesh.

1 Introduction

Hexahedral mesh generation has been subject to active research during the past twenty years. And, while some progress has been made in the area of push-button hexahedral mesh generation for fairly specialized classes of complex domains, a generalized hexahedral mesh generation algorithm does not exist which will support all types of domains and applications.

The existence theorem for hexahedral meshes provided by S. Mitchell [6] states that a solid homeomorphic to the sphere, whose boundary is tessellated by an even number of quadrilaterals, can be partitioned into a hexahedral mesh using interior surfaces whose boundaries are the dual cycles of the quadrilateral mesh. The solid partition is referred to as a constrained hexahedral mesh, and the partition of the boundary is known as the constraining quadrilateral mesh.

The problem of constructing constrained hexahedral meshes has proven very difficult to address. The techniques based on S. Mitchell's proof to the existence theorem are difficult to implement; in a few cases, seemingly simple problems are difficult to solve.

D. Eppstein [5] presented a complexity analysis on the generation of hexahedral meshes constrained to a bipartite quadrilateral mesh. Part of his construction depends on adding a layer of cells that have sixteen and eighteen faces; the problem of constructing the hexahedral solution to these cells of quadrilaterals is left open to the reader, and, instead, S. Mitchell's proof is invoked to prove existence of a solution to those cells. In his paper, D. Eppstein focuses on the analysis of the complexity of the generation of constrained hexahedral meshes.

In this paper, a constructive proof is given based on adding four basic transitional cells of hexahedral elements to a quadrilateral mesh: 1) a transition of paired hexahedra, 2) a transition to four-split hexahedra, and 3) a transition from four-split hexahedra to a closed mesh. The rules of how to build the transitional layers of hexahedra using these basic cells will be given. The result presented in this paper is a constructive, easily-programmable, solution that provides a precise, *a priori*, count on the number of hexahedral elements that will be generated.

Additionally, S. Mitchell [7] introduced the Geode-Template to interface a four-split quadrilateral mesh to a diced tetrahedral mesh. In his paper, Mitchell relies on splitting a hexahedral mesh to create a four-split, or diced, quadrilateral boundary. In this paper, we will show how to transition to a four-split mesh without modifying the original boundary.

The remainder of this paper will outline the concepts, definitions, and proofs which ultimately result in a constructive proof of S. Mitchell's existence theorem. The proof of the theorem presented in this paper can be summarized as follows:

1. We introduce the notions of a **Paired Partition** and **Transitions** between quadrilateral meshes. It is shown that every quadrilateral mesh that admits a Paired Partition has a transition to a quadrilateral mesh whose dual has no self-intersecting loops.

2. Given a quadrilateral mesh whose dual has no self-intersecting loops, we introduce a method for transitioning the quadrilateral mesh to a **Four-Split Quadrilateral Mesh**. The transition is created by inserting layers of elements that divide a quadrilateral in two along the each of the two dual cycles that compose the quadrilaterals.
3. It is shown that a **Four-Split Quadrilateral Mesh** on the sphere[4] is the boundary of a hexahedral mesh.
4. We demonstrate a **Four-Split Pyramid** cell to close the hexahedral mesh.
5. Finally, we show that any quadrilateral mesh on a sphere with an even number of quadrilaterals is the constraining boundary of a hexahedral mesh.

While topologically valid, the resulting quality of the hexahedral mesh created by this construction will not provide solutions for practical applications and is presented merely to provide a concrete measurable construction of a solution to the problem of constrained mesh generation.

The solution presented for the sphere can be extended to the case of the torus and compact 2-dimensional manifolds in general by using the Geode-template coupled with a constrained tetrahedral mesh. (If every loop in a quadrilateral mesh on a 2-dimensional compact manifold has an even number of quadrilaterals, it is possible to apply all the results of this paper to transition to a **Four-Split Quadrilateral Mesh**. This, then, will permit the use of the Geode-template and reduce the problem to the existence of a constrained tetrahedral mesh.)

Finally, a few solutions to open problems in mesh generation are presented including: a new solution to Schneider's open problem [11], the eight-sided quadrilateral octahedron [5], and Eppstein's cube [5]. Additionally, a question by M. Bern, et al, [2] on the existence of a hexahedral decomposition with linear edges for a convex polyhedron is solved by the construction provided in this paper.

2 Basic Terminology

The terms quadrilateral and hexahedral mesh follow the definition given by S. Mitchell in [6]. The dual of a quadrilateral mesh on a compact manifold is a graph where every vertex is connected to four other vertices (i.e. a 4-regular graph). A structure referred to as the Spatial Twist Continuum or STC for short is associated with this graph [9]. In this definition, the notion of chord is introduced. A chord is a chain of quadrilaterals that is constructed by traversing the adjacent quadrilaterals through opposite edges. A loop is

[4]Technically, the construction requires a four-split quadrilateral mesh with a 'star-shaped' boundary (i.e. there must be a point which can be seen by all nodes on the four-split boundary simultaneously.)

a closed chord. In particular, for a quadrilateral mesh on a closed compact manifold, every chord belongs to a loop.

Loops may self-intersect. T. Suzuki, et al [12] gave a detailed description of how to untangle self-intersecting loops to create the interior surfaces necessary to generate the hexahedral mesh. Their results are used to resolve R. Schneider's pyramid [11].

3 Element Representation

The quadrilateral and hexahedral elements to be referenced throughout the paper will follow the conventions used in finite-element analysis. A quadrilateral is represented by an ordered set of vertices {v1, v2, v3, v4}, and bounded by the four edges {v1, v2}, {v2, v3}, {v3, v4}, and {v4, v1}. The edges in the quadrilateral that do not share vertices are called opposite edges of the quadrilateral. A hexahedron is represented by an ordered set of vertices {v1, v2, v3, v4, v5, v6, v7, v8}, and bounded by the six faces {v1, v4, v3, v2}, {v5, v6, v7, v8}, {v5, v6, v2, v1}, {v8, v7, v3, v4}, {v6, v2, v3, v7}, and {v1, v5, v8, v4}.

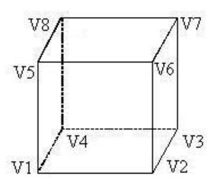

Fig. 1. Element configuration

Additional requirements of a valid quadrilateral mesh are that each edge in the mesh must contain exactly two distinct vertices, and each interior edge must be shared by exactly two quadrilaterals. Similarly, for a valid hexahedral mesh the faces of a hexahedron must contain exactly four distinct vertices, and each interior face of the hexahedral mesh must be shared by exactly two hexahedra.

4 Hexahedral Transitions of Quadrilateral Meshes

Definition 1 *Two distinct quadrilateral meshes are* ***transitions*** *of each other if there is a hexahedral mesh whose boundary contains the union of both meshes.*

By solving the hexahedral mesh of the transition of a given quadrilateral mesh, the original hexahedral problem is resolved, because the union of the hexahedral mesh with the layer of transition elements gives the solution to the original quadrilateral mesh.

4.1 Paired Partition Transition

Definition 2 *A* ***Paired Partition*** *of a quadrilateral mesh, Q, is a partition PQ of Q such that each element in the partition is a pair of quadrilaterals that share at least an edge.*

In other words, a quadrilateral mesh Q admits a paired partition if there exist a set

1. PQ = { {p, q}, such that p and q are quadrilaterals in Q},
2. Any two distinct elements in PQ {p, q} and {p', q'} are disjoint,
3. Q is the union of PQ , and,
4. For each element {p, q} in PQ, p and q share an edge or, equivalently, p and q are neighbors.

Since the dual of a quadrilateral mesh on a closed manifold is a 4-regular graph, a **Paired Partition** also corresponds to the graph-theoretic problem known as a perfect matching, or a 1-factor, of a 4-regular graph. We utilize the following theorem (a proof is given in [3]):

Theorem 1. *Every quadrilateral mesh on a 2-Dimensional manifold in* $\Re^3$ *with an even number of quadrilaterals admits a* ***Paired Partition****.*

Removal of Self-intersecting Loops

Theorem 2. *Every quadrilateral mesh on the sphere with n elements that admits a* ***Paired Partition*** *transitions to quadrilateral mesh with no self-intersecting loops. The total number of hexahedral elements within the transition between the original quadrilateral mesh and the new quadrilateral mesh with no self-intersecting loops is n.*

Proof: Construct the **Paired Partition** of the quadrilateral mesh. Let {p, q} be in PQ.

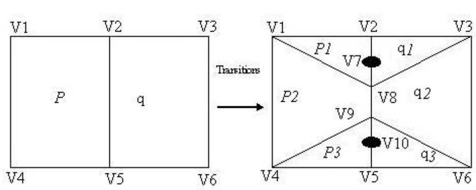

Fig. 2. The Paired Partition Transition. The image on the right is the 'Cell of Six Quadrilaterals'.

The two quadrilaterals with vertices {{v1, v2, v5, v4}, {v2, v3, v6, v5}} (see Figure 2) transition to the quadrilateral mesh with six quadrilaterals {{v1, v2, v7, v8}, {v7, v2, v3, v8}, {v1, v8, v9, v4}, {v8, v3, v6, v9}, {v4, v9, v10, v5}, {v9, v6, v5, v10}}. The boundary of the hexahedral mesh comprised of the two hexahedra with vertices {v1, v2, v5, v4, v8, v7, v10, v9}, and {v2, v3, v6, v5, v7, v8, v9, v10} mesh below is the exclusive union of the two sets of quadrilaterals. For any two paired quadrilaterals, p and q, with vertices {v1, v2, v3, v4}, and {v2, v5, v6, v3}, construct the two hexahedral elements with vertices {v1, v2, v3, v4, v7, v8, v9, v10}, and {v2, v5, v6, v3, v8, v7, v10, v9}.

This transition is applied to each set of paired quadrilaterals in the **Paired Partition**. The boundary of the transition mesh minus the original quadrilateral mesh is composed of cells of six quadrilateral elements (see Figure 2). Thus each paired element in the **Paired Partition** is mapped to a unique set of quadrilaterals {p1, p2, p3, q1, q2, q3} in the transitioned mesh.

There is a natural partition induced by mapping each element {p, q} in the **Paired Partition** to a unique subset of quadrilaterals {p1, p2, p3, q1, q2, q3} of the transitioned mesh. We will call the newly introduced partition of the mesh '**Cell of six quadrilaterals**'. As a consequence of a peculiar property of these new cells, it will be shown that all the loops in the new quadrilateral mesh that result from the transition are non-self-intersecting.

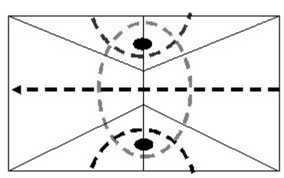

Fig. 3. Dual chords on the Paired Partition Transition boundary

The fact that the new mesh will not contain any self-intersections can be seen by taking any quadrilateral in the transitioned quadrilateral mesh, and finding the Cell of six quadrilaterals that contains the quadrilateral. Notice that there are two types of loops that go through a **Cell of six quadrilaterals**: there is one loop that is fully contained inside a **Cell of six quadrilaterals**, and three others that are not (see Figure 3). Notice, also, that the only intersections in the cell take place between the fully contained loop in the cell and one of the loops that is not fully contained in the cell (see also Figure 4). Therefore, for any given quadrilateral, the intersection must take place between two distinct loops. Hence, there cannot be any self-intersections.

A total of n hexahedral elements were used to transition to a quadrilateral mesh with no self-intersecting loops. We should also note here, that it may be possible that changing only a subset of the pairs may be required to remove all self-intersections in the mesh. That is to say that n is an upper bound on the minimum transition layer to obtain no self-intersections.

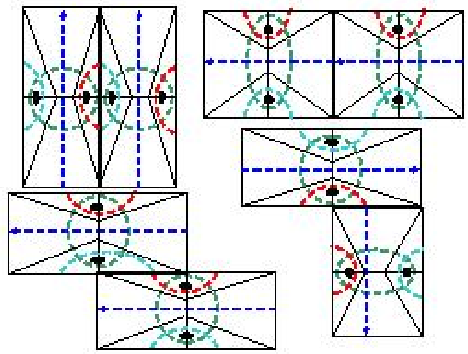

Fig. 4. Various arrangements of dual chords with adjacent **Paired Partition** transitions.

4.2 The Four-Split Transitions

The Four-Split Transition

In this section, we will describe a transition from a non-self-intersecting quadrilateral mesh to a **Four-Split Quadrilateral mesh**. A **Four-Split Quadrilateral mesh** is the collection of four cells that result from splitting a single quadrilateral into four quadrilaterals; five points are added: four at the mid-edges, and one at the center as shown in Figure 5. Any quadrilateral mesh that results from a transition which splits one quadrilateral into four is called a **Four-Split Quadrilateral Mesh**.

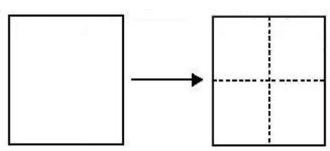

Fig. 5. Four-Split Transition

A **Four-Split Quadrilateral Mesh** has the following properties:

1. There are four vertices labeled as corner vertices.
2. There is a unique vertex labeled as interior.
3. There are four vertices labeled as mid-edges.

4. Adjacent cells share corner vertices with corner vertices, and mid-edge vertices with mid-edge vertices.

The fourth property is essential to the proof of the next theorem to ensure that the faces of the **Four-Split Pyramid** given below will match appropriately.

We now define Theorem 3 which provides a sufficient condition for the existence of a transition to a **Four-Split Quadrilateral Mesh**.

Theorem 3. *If each of the dual loops of the quadrilaterals on the sphere does not self-intersect, there is a transition of the mesh to a four-split quadrilateral mesh.*

Proof: A given oriented loop in the dual of the quadrilateral mesh which does not self-intersect splits the mesh into three disjoint sets of quadrilaterals: 1) the set of quadrilaterals that lie to the right of the loop, 2) the set of quadrilaterals that compose the loop, and 3) the set of quadrilaterals that lie to the left of the loop. The quadrilaterals that compose the loop are a boundary between the quadrilaterals labeled left and right. This partition of the mesh exists because there are no self-intersections that change the orientation of the curve.

By utilizing these three sets of quadrilaterals, we can begin to add layers of hexahedra onto the quadrilaterals which will result in a four-split quadrilateral transition. There are three cases to consider when adding a layer of hexahedra (shown in Figure 6): case 1 is used when a quadrilateral lays in the regions labeled as right of the oriented loop, case 2 is utilized when a quadrilateral belongs to the loop, and case 3 is used when a quadrilateral lies to the left of the loop. We will discuss each of these cases separately.

- **Case 1:** If a quadrilateral lies to the right of an oriented loop, a single hexahedron is added on top of the quadrilateral towards the center of the sphere.
- **Case 2:** When a quadrilateral belongs to a loop, a cell of three quadrilaterals is placed as illustrated in Figure 6.
- **Case 3:** When a quadrilateral lies in the region labeled as left of the loop, two hexahedra are added on top of the quadrilateral towards the center of the sphere.

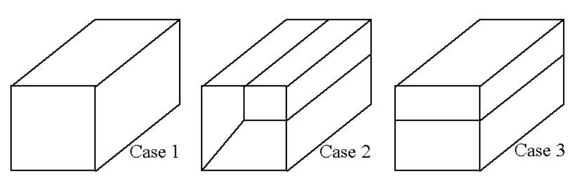

Fig. 6. Transition cell for the Four-Split transition

Thus, three types of transition cells placed at each quadrilateral of the mesh: one is a single hexahedra, another is a pair of hexahedra one on top of the other, and the third one is the one illustrated in Figure 6.

By placing the set of cells on top of each quadrilateral as described above for each of the cases, we will now show that the cells will match appropriately at the interface with each of the other sets of cells for each quadrilateral. For a single loop, take any two quadrilaterals sharing an edge with the cells placed on top of the quadrilateral as described above, and the result is one of six possible scenarios:

- *Scenario A* -Both quadrilaterals are in the set of 'right' quadrilaterals: In this case, two hexahedra are placed next to each other, and will match up appropriately.
- *Scenario B* - Both quadrilaterals are in the set of 'left' quadrilaterals: In this case two pairs of hexahedral elements are palaced next to each other, and the hexahedra will match up appropriately.
- *Scenario C* - One quadrilateral is labeled 'left', and a neighbor is labeled 'right': This scenario is not possible, because the oriented loop with no self-intersections is the boundary that divides the 'left' quadrilaterals from the 'right' quadrilaterals by definition.
- *Scenario D* - One quadrilateral is labeled 'right' and the other belongs to the loop: The quadrilateral labeled 'right' must lie in the region to the right of the loop. In this case, there is one hexahedral element on the right side of the loop that is matched with an appropriately aligned hexahedra from case 2 template.
- *Scenario E* - One quadrilateral is labeled 'left' and the other belongs to the loop: Using similar reasoning to case D, the 'left' quadrilateral is to the left of the quadrilateral in the loop, and there are two hexahedral elements matching two hexahedral elements from the case 2 template, and the hexes match up appropriately.
- *Scenario F* - Both quadrilaterals belong to the loop: Label the two quadrilaterals p and p' having vertices v1, v2, v3, v4, and v3, v2, v5, v6 respectively. The quadrilaterals p and p' share the edge v2, v3.
  There are two possible cases: i) one quadrilateral is a successor of the other quadrilateral with respect to the loop, or ii) the two quadrilaterals are not successors of each other.
  - *Case i* - The quadrilaterals are traversed successively in the loop: Suppose the loop traverses quadrilateral p through edges v4, v1 and v2, v3. If the loop traverses quadrilateral p' through v2, v5 and v3, v6 (see Figure 7), the loop must also traverse quadrilateral p' through two additional edges v3, v2 and v5, v6. If the loop traverses quadrilateral p' at four edges, the loop is self-intersecting at p', which is contradictory to the assumption of no self-intersections. Therefore, the loop will traverse the adjacent quadrilateral p' through edges v3, v2 and v5, v6, and the hexes match up appropriately.

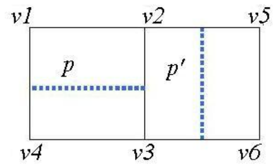

Fig. 7. This configuration is will occur only if there is a self-intersecting loop in the quadrilateral mesh.

- *Case ii* - The quadrilaterals are not successively traversed in the loop: In this case, the case 2 templates must face in opposite directions with respect to the loop; that is, the left edge of one template must be sharing the left edge of the adjacent template, or the right edge of one template is sharing the right edge of the adjacent template, as shown in Figure 8).
  If the loop is traversing p through edges v1, v2 and v3, v4, the loop cannot traverse at p' through the shared edge v3, v2. Otherwise, p could be considered a successor or predecessor of p'; this would lead to the loop being self-intersecting at p and contradicting our assumptions on the loop. Hence, the loop must traverse p' through the edges v2, v5 and v3, v6.. Therefore, we need to show that p cannot be to the 'left' of p' while p' is to the 'right' of p. We can demonstrate that this is impossible by using basic properties of simple Jordan curves.
  We create an oriented, closed loop of segments by connecting the midpoint of each edge traversed by the loop from all the quadrilaterals in the loop. This oriented, closed loop of segments is a simple Jordan curve because the loop that induced it is not self-intersecting.
  Construct an oriented, open poly-segment with vertices v1, v2, v5. This poly-segment must intersect the oriented, closed curve induced by the loop at two different points. As a direct result of the property of simple Jordan Curves,one intersection point the tangent of the oriented closed loop must be pointing to left of the oriented poly-segment, and, at the other intersection point, the tangent must be pointing to the right.
  We are, therefore, left with two possibilities: the loops are either to the right of each other, or to the left of each other. In either case, the hexahedra from the case 2 template will match up appropriately as illustrated in Figure 8.

We now return to the **Paired Partition** transition described earlier, and we notice an interesting pattern which can be exploited to reduce the number of elements in the four-split transition. Each **Cell of Six Quadrilaterals** (as shown in Figure 3) contains two types of loops: the inner loops, and the exterior loops (as described earlier). None of the inner loops intersect each other; hence, it is possible to orient all loops so that their direction is consistent. Similarly, none of the exterior loops intersect each other, and we can

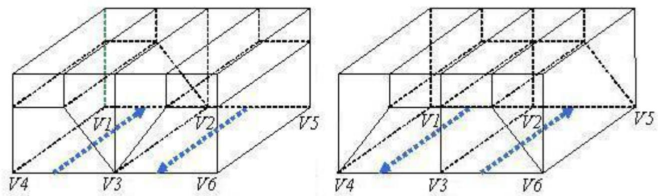

Fig. 8. Possible loop orientations in Scenario F, case ii

therefore orient these loops, as well. Thus, by adding only two layers of two splits, it is possible to transition to a four-split quadrilateral mesh. Each of the cells has two outer loops oriented to the left of each other, and the remaining loop to the right of the center loop, and placing each element within the cell distinctly into one of the scenarios described above.

As a result of this ability to distinctly orient each of the loops, we can calculate the total number of transition elements needed to convert from a **Paired Partition** to a **Four-Split Quadrilateral Mesh** as:

*number of transition elements = number of quadrilaterals to the right of the loop + 2*number of quadrilaterals to the left + 3*number of quadrilaterals from the loop*

The hexahedral elements transition the original quadrilateral elements to one where the quadrilaterals to the left and right of the mesh remain identical to the originals, but the ones along the loop are split into two quadrilaterals. The set of originals chords is transferred to the transition quadrilateral mesh as follows:

1. The loop just processed is discarded from the set of chords,
2. The remaining chords are mapped onto the faces of the transition quadrilateral mesh. In particular, when a loop is projected onto one of the pair of quadrilaterals resulting from the split along the loop, it must be transverse to the loop.

The operation of adding two split transition layers described above is repeated for another loop of the remaining loops projected onto the new quadrilateral mesh that results from the transition elements. Each time a loop is processed the set of remaining loops diminishes by one. The process continues iteratively until no more loops remain from the original set. Wherever two loops intersect, a group of **Four-Split Cell** is created, because two quadrilaterals resulted from the loop processed at an earlier stage in the process, and the secondarily processed loop adds two more quadrilaterals along the transversal direction. Since each loop is processed only once, and exactly two loops cross each quadrilateral cell in traversal directions; hence, the resulting transition of quadrilaterals will be a **Four-Split Quadrilateral Mesh**. Figure 9 illustrates the results of two layers from two different loops that intersect.

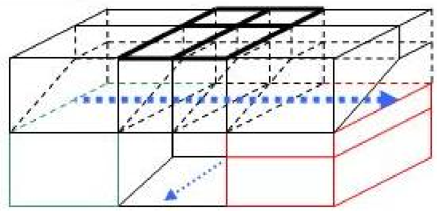

Fig. 9. Two layers result from two different loops intersecting.

An Alternative Four-Split Transition

Scott Mitchell suggested an alternative construction [10]. The alternative transition is accomplished by adding one layer of hexahedra on the elements that are either on the loop or to the left of the loop. All other quadrilaterals are left as they are. The proof of the theorem is very similar to the one above. Using his approach, the number of transition elements added at each layer is given by the formula:

number of transition elements = number of quadrilaterals to the left of the loop + number of quadrilaterals on the loop

It is critical for both four-split transitions that the quadrilateral mesh is on the sphere. A torus, for example, may have loops that do not split the domain in two regions as required by the proof. In general, non-intersecting loops could be processed simultaneously by the approach given above if the chords are carefully oriented to reduce the number of layers.

4.3 Four-Split to Closure Transitions

Once a **Four-Split Quadrilateral Mesh** is in place it is possible to transition to a constrained hexahedral mesh utilizing one of several other transitions. We now demonstrate how the remainder of the sphere can be filled using a Four-Split Pyramid, or the Geode-template to obtain an all-hexahedral mesh.

The Four-Split Pyramid Transition

Theorem 4. *A* ***Four-Split Quadrilateral Mesh*** *is the boundary a hexahedral mesh containing 16*n hexahedra, where n is the number of quadrilaterals before four-split division.*

Proof: This construction is done by utilizing a hexahedral decomposition of a pyramid into sixteen hexahedra (a detailed construction of the pyramid is given in [4].) This pyramid is characterized by having a **Four-Split Quadrilateral Mesh** at the base of the pyramid (the pyramid is illustrated in Figure 10). The **Four-Split pyramids** are placed inside the sphere with their bases aligned at each of the four-split cells with the apex of pyramids being connected to the center of the sphere, and the midpoints to the corresponding midpoints of the faces to the adjacent cells as illustrated in Figure 10. The connectivity to adjacent **Four-Split Pyramids** is guaranteed by the fundamental property of the **Four-Split Quadrilateral Mesh** that ensures that corner vertices meet with comer vertices, and mid-edge vertices meet with mid-edge vertices.

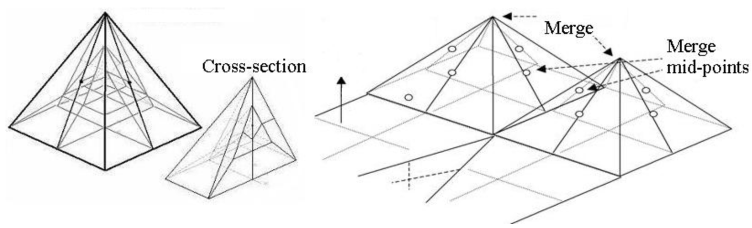

Fig. 10. A cross-sectional view of the Four-Split Pyramid. The mid nodes and the tips of each of the Four-Split pyramids are merged to ensure conformal meshes.

S. Mitchell's Geode Template

The **Four-Split Pyramid** contains several hexahedra which share two faces or edges with neighboring elements (also known as "doublets" [8]). therefore, a very attractive alternative to the **Four-Split Pyramid** is S. Mitchell's Geode-Template [7]. The Geode-template contains more elements than the **Four-Split Pyramid**, but reduces the number of doublets in the resulting mesh. In this section, we will show how the Geode-template can be utilized in place of the **Four-Split Pyramid**.

Fig. 11. The Geode-template by S. Mitchell

The Geode-Template contains 26 hexahedral elements, and contains a **Four-Split Quadrilateral Mesh** at the base. The template was designed to match a four-split quadrilateral cell to a diced tetrahedral constrained mesh. The interior of the mesh is filled by two four element hexahedral dicing of a tetrahedral element where the apex is connected at the center of the sphere.

If we cap the geode template with a pyramid split into two diced tetrahedra, the resulting hexahedral decomposition can be used in place of the **Four-Split Pyramid** described earlier.

5 A Constructive Hexahedral Existence Theorem

5.1 Solutions using the Four-Split Pyramid

Theorem 5. *A quadrilateral mesh on the sphere with an even number of quadrilaterals is the boundary of a hexahedral mesh of the interior of the sphere. The total number of hexahedra is 76*n where n is the number of quadrilaterals on the boundary.*

Proof: By Theorem 1, every even-parity quadrilateral mesh on a 2-dimensional compact manifold admits a **Paired Partition**. By Theorem 2, a transition to a quadrilateral mesh can be constructed with no self-intersecting loops. By Theorem 3, the quadrilateral mesh transitions to a four-split, and, by Theorem 4, the quadrilateral mesh is the boundary of a hexahedral mesh.

The total number of hexahedral elements is given by

*number of hexahedra = number of elements to resolve self-intersecting loops + number of elements to transition to a four-split + 16 * number of four-split pyramids*

1. The number of hexahedra added by applying the transition that removes self-intersecting loops illustrated in Figure 3 is n.
2. For each of the six quadrilaterals of each cell, the total number of hexahedra needed to transition to a four-split is 9; hence the total number of hexahedra needed to transition the mesh to a four-split is $9 * 6 * n/2$.
3. The total number of hexahedral elements per cell needed to solve the four-split is $16 * 6 * n/2$.

Hence, the total number of hexahedra to fill the interior is $n + 48*n + 27*n$ which equals $76*n$ total elements.[5]

5.2 Solutions using the Geode-Template

Utilizing the **Paired Partition** transition and the **Four-Split** transition, but replacing the **Four-Split Pyramid** with the modified geode-template results in a solution with fewer 'doublet' entities in the mesh. This solution requires:

1. $n + 9*6*n/2$ hexahedra needed to transition to the **Four-Split Quadrilateral Mesh**

[5]Using Mitchell's alternative Four-Split transition in the section titled **An Alternative Four-Split Transition**, the number of hexahedra required for the constrained solution is $54*n$.

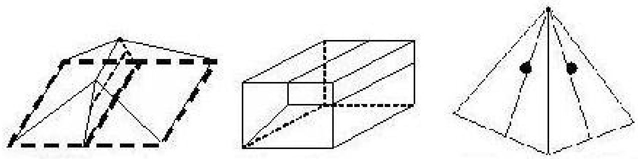

Fig. 12. The various transition layers used in the constructive proof.

2. plus, *26*6*n/2* elements from the Geode-Template
3. plus, *4*2*6*n/2* elements for the diced tetrahedra with apex at the center of the mesh.

This new solution of the constrained hexahedral mesh contains a total of ***130*n*** elements,[6] but will contain significantly less doublets than a solution produced by the **Four-Split Pyramid**.

5.3 Solutions Using a 'Pillowed' Four-Split Pyramid

Another slightly different approach replaces the transitions from the paired partition and the **Four-Split Pyramid** by transition cells but do not contain the doublet elements identified earlier. The doublet elements can be removed in the transition layers by applying the doublet-pillowing technique described by S. Mitchell and T. Tautges [8]. The base of the new **Four-Split Pyramid** will still be a four-split quadrilateral cell, and the resulting transition cells still have no self-intersecting loops, and will not contain any doublets. However, by removing the doublets the resulting solution to the constrained hexahedral problem requires 5396*n hexahedral elements.

6 Applications

In this section we demonstrate constrained hexahedral solutions to the quadrilateral meshed boundaries described as Schneider's Pyramid, the Quadrilateral Octahedron, and Eppstein's Cube. In all cases, the solution is a direct result from Theorem 5. The solutions are presented by conveniently numbering the quadrilaterals such that the quadrilateral admits a **Paired Partition** of the form **PQ = {{q1, q2},...,{q2k-1, q2k},...,{qn-1, qn}}** where n is the number of quadrilaterals for each case; the total number of hexahedra used to solve the constrained hexahedral mesh will be ***76*n*** or ***130*n*** depending on which solution is used. The source code used in generating these solutions is available at [1].

[6] Using Mitchell's alternative Four-Split transition in the section titled **An Alternative Four-Split Transition**, the number of hexahedra required for the constrained solution using the Geode-template is 112*n.

6.1 Schneider's Pyramid

Figure 13 contains an open view of Schneider's pyramid with each of the boundary quadrilaterals being numbered. There are 16 quadrilaterals on the boundary of the pyramid resulting in a total of 1216 hexahedral elements being generated using the **Four-Split Pyramid**, or 2080 hexahedral elements if the Geode-template is utilized.

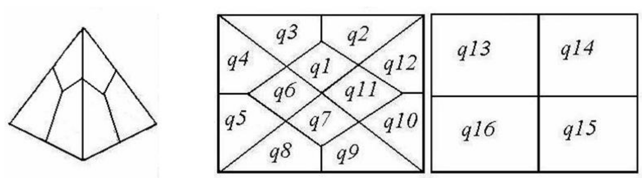

Fig. 13. Schneider's pyramid

6.2 Quadrilateral Octahedron

Figure 14 contains an open view of the Quadrilateral Octahedron with each of the boundary quadrilaterals being numbered. There are 8 quadrilaterals on the boundary of the pyramid resulting in a total of 608 hexahedral elements being generated using the **Four-Split Pyramid**, or 1040 hexahedral elements if the Geode-template is utilized.

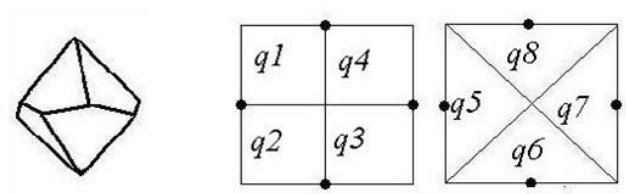

Fig. 14. Quadrilateral Octahedron

6.3 Eppstein's cube

In Eppstein's original construction, there are two sets of quadrilateral cubes used to transition to the tetrahedral based mesh. One had sixteen elements, and the other eighteen. These cubes contain quadrilateral doublets (i.e faces that share two edges with an adjacent neighbor). There is another version of Eppstein's cubes with 22 and 20 quadrilateral non-degenerate elements respectively. We focus on the 16-quadrilateral cube shown in Figure 15.

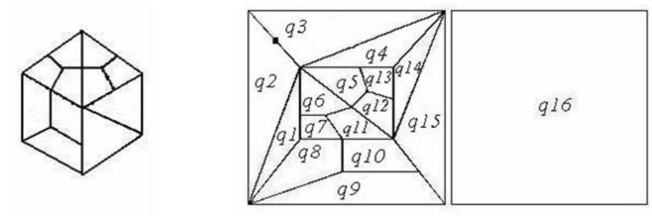

Fig. 15. Eppstein's Cube

The value of **n** for Eppstein's cube shown in Figure 15 is 16. A total of 1216 hexahedral elements are needed to resolve the constrained mesh utilizing the **Four-Split Pyramid** solution, or 2,080 hexahedra if the Geode-template solution is utilized.

7 Conclusions

The construction presented in this paper demonstrates a solution of S. Mitchell's existence theorem. However, this construction is different from the constructions utilized in S. Mitchell's proof. Indeed, his solution cannot lead to the construction given in this paper. In the approach outlined in the original proof of S. Mitchell's existence theorem, a mesh that contains a loop with an odd number of intersections will have to be connected through an interior surface to another loop with an odd number of intersections on the boundary. The **Paired Partition** transition has the very interesting property of locally connecting all of the loops defined by the initial quadrilateral boundary mesh through interior surfaces, and simultaneously creating a transition to a quadrilateral mesh which contains no self-intersecting loops.

We have given results for Schneider's pyramid and Eppstein's cube and Quadrilateral Octahedron, with element counts needed to generate a hexahedral topology in these solids using the construction outlined in this paper. The solution presented for the sphere can be extended to the case of the torus and compact 2-dimensional manifolds in general by using the Geode Template coupled with a constrained tetrahedral mesh. From the construction, it follows that it is possible to generate a hexahedral decomposition with linear edges for a quadrilateral mesh of a convex region. The question of finding a general construction with a minimal number of elements with linear edges is open.

8 Acknowledgements

The authors would like to express appreciation to Scott Mitchell for his review of the paper, and the many additional insights provided. We would also like to thank Cynthia Phillips, M. Gopi, and David Eppstein for their guidance and expertise in graph theory which enabled us to formulate a proof for perfect matching of closed quadrilateral meshes.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

1. C. D. Carbonera, 'A Constructive Approach to Hexahedral Mesh Generation: Algorithm Implementation', source code available from http://carbonera.uprr.pr/.
2. M. Bern, D. Eppstein, P. K. Agarwal, N. Amenta, P. Chew, T. Dey, D. P. Dobkin, H. Edelsbrunner, C. Grimm, L. J. Guibas, J. Harer, J. Hass, A. Hicks, C. K. Johnson, G. Lerman, D. Letscher, P. Plassmann, E.Sedgwick, J. Snoeyink, J. Weeks, C. Yap, , and D. Zorin. Emerging challenges in computational topology. In *NSF-funded Workshop on Computational Topology*, pages –, 1999.
3. C. D. Carbonera and J. F. Shepherd. On the existence of a perfect matching for 4-regular graphs derived from quadrilateral meshes. *SCI Institute Technical Report*, UUSCI-2006-021, 2006.
4. Carbonera Pyramid, available from http://www-users.informatik.rwth-aachen.de/~roberts/SchPyr/index.html.
5. D. Eppstein. Linear complexity hexahedral mesh generation. In *12th ACM Symposium on Computational Geometry*, pages 58–67. ACM, 1996.
6. S. A. Mitchell. A characterization of the quadrilateral meshes of a surface which admit a compatible hexahedral mesh of the enclosed volumes. In *13th Annual Symposium on Theoretical Aspects of Computer Science*, volume Lecture Notes in Computer Science: 1046, pages 465–476, 1996.
7. S. A. Mitchell. The all-hex geode-template for conforming a diced tetrahedral mesh to any diced hexahedral mesh. In *Proceedings, 7th International Meshing Roundtable*, pages 295–305. Sandia National Laboratories, October 1998.
8. S. A. Mitchell and T. J. Tautges. Pillowing doublets: Refining a mesh to ensure that faces share at most one edge. In *Proceedings, 4th International Meshing Roundtable*, pages 231–240. Sandia National Laboratories, October 1995.
9. P. J. Murdoch and S. E. Benzley. The spatial twist continuum. In *Proceedings, 4th International Meshing Roundtable*, pages 243–251. Sandia National Laboratories, October 1995.
10. Scott Mitchell to Jason Shepherd, email entitled 'constructive approach to constrained hex mesh generation', dated March 31, 2006.
11. Schneiders Pyramid Open Problem, http://www-users.informatik.rwth-aachen.de/~roberts/open.html.
12. T. Suzuki, S. Takahashi, and J. F. Shepherd. Practical interior surface generation method for all-hexahedral meshing. In *Proceedings, 14th International Meshing Roundtable*, pages 377–397. Sandia National Laboratories, September 2005.