SAND2008-1880C

Surface Splicing: A Dual Based Approach to Surface Mesh Modification and Generation

Benjamin T. Grover[1]
Steven E. Benzley[2]

Department of Civil and Environmental Engineering
Brigham Young University
Provo, UT 84606 USA
[1]benjy@byu.edu
[2]seb@byu.edu

Jason F. Shepherd

Parallel Computing Sciences Department
Sandia National Laboratories
Albuquerque, NM 87185 USA
jfsheph@sandia.gov

Abstract

The effective generation of high quality quadrilateral surface meshes is an area of important research and development for the finite element community. This paper presents a new and unique procedure named "Surface Splicing," that allows for the generation of all quadrilateral surface meshes as well as the ability to edit these meshes via the dual. The dual contains the same data as the mesh but, unlike the mesh, the dual directly allows a unique visualization of how surface and volume elements interrelate and connect with one another. The dual also provides mesh connectivity information that is crucial in forming an all-quadrilateral surface mesh that can form the basis of a valid all hexahedral volume mesh.

Introduction

The effective generation of high quality quadrilateral surface meshes is an area of important research and development for the finite element community. Quadrilateral elements generally lead to more efficient and accurate finite element results. In addition, some all hexahedral volume-meshing algorithms are based on an initial quadrilateral surface mesh that has specific connectivity requirements [7].

This paper presents a new and unique mesh generation procedure that provides for the creation and editing of quadrilateral surface finite elements based on the dual of the mesh. The dual, which will be explained in detail later in this paper, contains the same data as the mesh. The dual, however, helps to visualize how all elements on a surface or in a volume interrelate and connect to one another. Being able to visualize this interrelationship provides a powerful means to edit surface and volume meshes.

Frequently in finite element mesh generation, automatically created meshes need to be modified in some areas so that the mesh will be of higher quality and in turn give more accurate analysis results. By utilizing the dual to edit the mesh, one can easily see what changes will propagate thorough a mesh if one element is deleted or moved. Without utilizing the dual, monitoring these changes is difficult.

The Dual for 2-D Quadrilaterals

A 2-D dual for a surface quadrilateral mesh as defined by Murdoch[6] is a collection of centroids and chords. A "centroid" is the dual representation of quadrilateral element, and a "chord" is a dual representation of row of quadrilateral elements. A centroid is formed at the intersection of two dual chords, and this centroid corresponds to a specific quadrilateral element.

Figure 1 shows the mesh and dual entities of a quadrilateral element. Note that each 2-cell in dual space corresponds to a specific node of the mesh. A 2-cell is an n-sided polygon whose edges are composed of segments of dual chords. A 2-cell is the smallest closed dual loop on a surface. Figure 1 also shows a centroid (i.e. the intersection of two dual chords) in dual space. Note that each centroid will correspond to a specific quadrilateral element of the mesh.

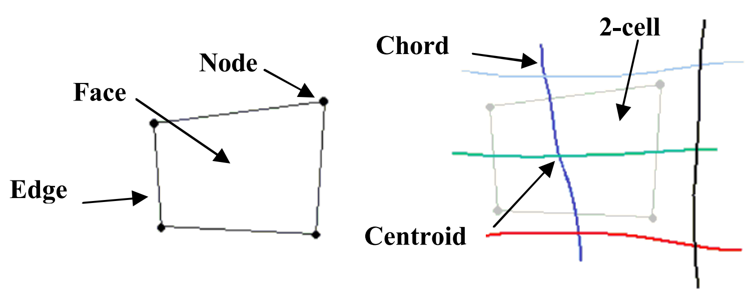


Figure 1. Mesh and dual entities

Dual chords can begin and terminate at surface boundaries. This is the analogue of an all-quadrilateral mesh having an even number of surface boundary intervals. In addition to beginning and ending on a boundary, dual chords can also close on themselves to make a closed loop on the interior of a surface.

When creating the dual, the chords can be created in any order. All chords are totally formed when two chords intersect at the face of every quadrilateral element. As the dual is created for a given surface mesh, three rules must be obeyed. These rules are:

1. Dual chords can be nowhere tangent.
2. Two chords always intersect in the interior of a quadrilateral element
3. Only two chords can intersect at any given point.

These rules must be followed both when creating the dual from an existing surface mesh, and creating a dual with no existing mesh. Table 1 shows a summary of the mesh and dual entities explained in this section.

Table 1. Mesh and Dual Entities for the 2-D Quadrilateral Dual

Mesh Entity	Dimension	Dual Entity	Dimension
Face	2	Centroid	0
Edge	1	Chord	1
Node	0	2-Cell	2

The Surface Splicing Algorithm

The Surface Splicing Algorithm [5] will be briefly explained here. The algorithm is used to create an all-quadrilateral surface mesh from the dual of the mesh. The dual can either be created manually or can be generated directly from an existing all-quadrilateral surface mesh.

The steps of the algorithm are in given the following list. If the dual is being created manually, the boundary edges on the surface must be meshed before these steps take place.

Steps of the Algorithm:

1. Form the chords and centroids, either created manually or generated from an existing all-quadrilateral surface mesh.
2. Traverse the chords and centroids and create the 2-cells.

3. If the dual is being created manually: find existing boundary nodes and create new interior nodes.
4. Connect the nodes to create quadrilateral elements.

By following the steps above the mesh is created quite easily. The previous section explained that each 2-cell corresponds to exactly one node. Since the 2-cells are created by the algorithm then a node can be created in each 2-cell. Also as the dual is created the chords that attach into each centroid are stored in a counter clockwise order so that the nodes can be connected correctly.

To use Surface Splicing for mesh editing the dual is generally created automatically from an existing quadrilateral mesh. The dual can be manipulated, moved or even re-created in some sections. Surface Splicing then recreates the mesh by following the same steps as given above. The next section gives some examples of meshes that have been edited using this procedure.

Examples

This section provides examples using the dual representation and surface splicing to manipulate and edit quadrilateral surface and hexahedral volume meshes.

Example 1-Removal of Self-Tangencies on a Surface Mesh

The first example shows how the removal of self-tangent chords on a surface can improve the quality of the mesh. The mesh in Figure 2 was created using the whisker weaving algorithm [7].

The current CUBIT[4] whisker-weaving algorithm automatically removed initial surface self–intersections in Figure 2, however it modified the surface mesh in a manner that has an adverse effects on both the surface mesh and volume mesh by creating element self tangencies shown in Figure 2. An element self-tangency is created when a row of elements that are on the same dual chord also have shared edges. The self-tangency is shown in the darker gray color.

The CUBIT mesh generation toolkit [2] provides for the manual deletion of dual sheets. The self-tangent dual sheet of the hook object is shown in Figure 3. Such sheets result in poor quality meshes. In general, smooth sheets with little curvature produce high quality elements.

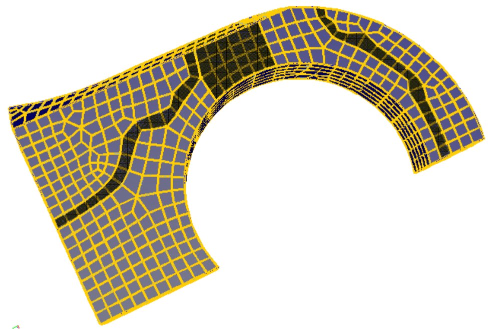


Figure 2. Hook shaped object meshed with the whisker weaving algorithm

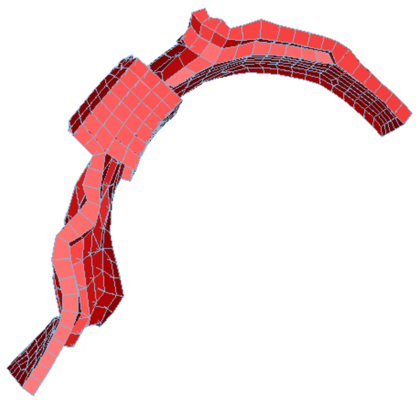


Figure 3. Dual sheet in the hook object shown in Figure 3.

The dual sheet shown in Figure 3 was deleted from the volume mesh in Figure 2. Additional sheets were deleted until the new mesh given in Figure 4 was created. Table 2 compares the mesh quality metrics for Figures 2 and 4. This data shows that if meshes can be edited or manipulated in dual space then the metrics that measure mesh quality can be improved.

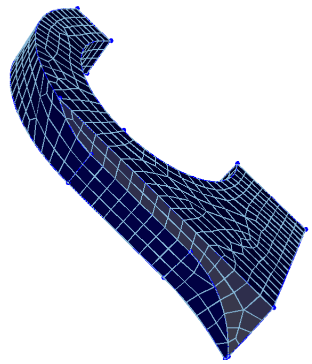


Figure 4. Hook figure with dual sheets removed

Table 2. Mesh quality metrics

Metric	Average		Std. Deviation		Minimum		Maximum	
	Before	After	Before	After	Before	After	Before	After
Scaled Jacobian	0.5372	0.6208	0.3155	0.1812	-0.9864	0.08475	0.9987	0.9968
Skew	0.5922	0.5336	0.2381	0.2133	0.007467	0.03209	1.00	0.9176
Aspect Ratio	2.2	1.92	1.759	0.6042	1.014	1.008	46.78	5.067
Shape	0.5721	0.6267	0.2749	0.1533	0	0.1634	0.9890	0.9844

The quality metrics used are scaled Jacobian, skew, aspect ratio and shape [4]. The scaled Jacobian metric is found by dividing the minimum Jacobian by the lengths of the three edge vectors. The skew metric is defined as the maximum cosine of the angle between edges at the hex center. The aspect ratio is the maximum edge length ratios at the hex center. Lastly, the shape is found by 3/mean ratio of the weighted Jacobian matrix. The acceptable ranges for these metrics are, scaled Jacobian(0.5-1), skew(0-0.5), aspect ratio(1-4), and shape(0.3-1) [4]. The optimum values for these metrics are scaled Jacobian(1.0), skew(0.0), aspect ratio(1.0), and shape(1.0).

Example 2- Changing the Interval Count Across a Surface Mesh

Figure 5 shows an example of a paved[1] mesh that does not maintain a consistent number of intervals across a surface. Here we use the capability to edit a mesh by using the dual. Figure 6 shows the deleted mesh region.

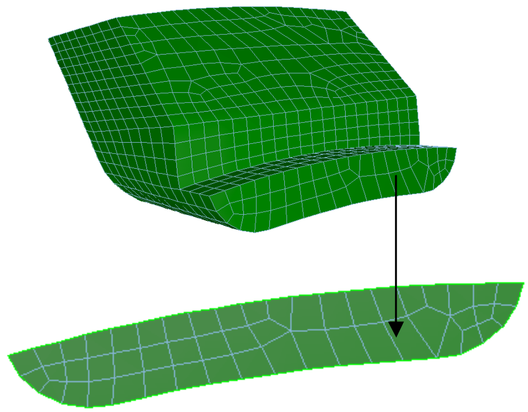


Figure 5. Undesirable mesh interval on the surface

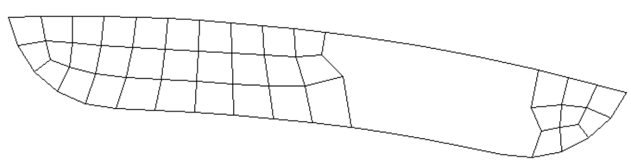


Figure 6. Deleted mesh in a specified region

Figure 7 depicts a new dual created and finally Figure 8 shows the new mesh. Notice that only the mesh in the region specified was modified and the remainder of the mesh was left completely unchanged.

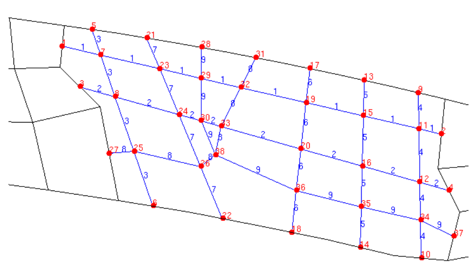


Figure 7. Newly created dual

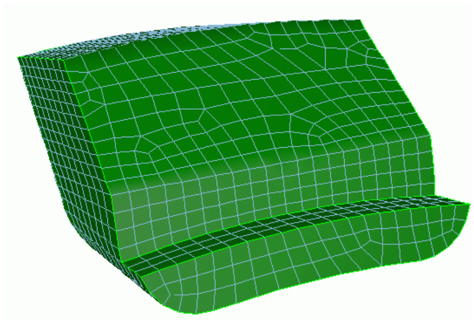
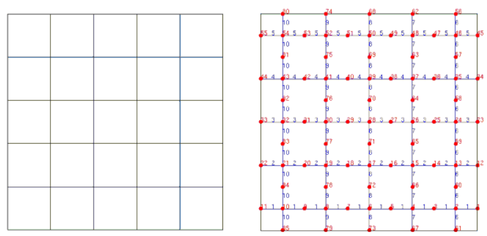


Figure 8. Newly created surface mesh from edited dual

Example 3- Editing of a Mapped Surface Mesh

The next example begins with a mapped[3] five by five surface mesh. Figure 9 shows the original mesh and the dual that was automatically created from the mesh.



(mesh) (dual)

Figure 9. 5 by 5 mapped mesh and its dual representation

With the dual of the mesh available it can be manipulated in many different ways. In this example, chord 3 (the third horizontal chord from the top) was deleted and then a new chord was created by connecting existing centroids on the surface. Figure 10 shows this new chord that was created, and Figure 11 shows the final mesh that was created from the new dual.

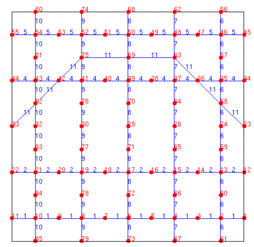


Figure 10. New chord created in the existing dual.

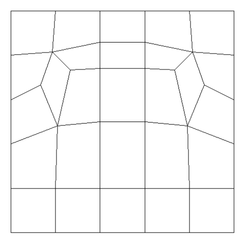


Figure 11. Mesh created from the edited dual in Figure 8.10

Conclusion

Many quadrilateral meshing algorithms in use today create good quality surface meshes. However, once the surface mesh is created, the ability to edit this mesh effectively is a challenge. This paper defines a new algorithm; Surface Splicing that allows for the editing of quadrilateral surface meshes in a manner that higher quality elements can be formed. The Surface Splicing Algorithm uses a dual based approach, which facilitates visual interpretation of how changes in one part of the mesh can propagate across the surface. This ability to visualize interrelationships and changes is not readily apparent in the mesh. Using the dual to edit and change meshes presents a powerful new way to modify and create all quadrilateral surface meshes.

References

1. Blacker, T.D., Stephenson, M.B., “Paving: A New Approach to Automated Quadrilateral Mesh Generation”, *International Journal for Numerical Methods in Engineering*, Vol. 32, pp. 811-847, 1991.
2. Borden, M., Master’s Thesis in Progress, Brigham Young University
3. Cook, W.A., Oaks, W.R., “Mapping Methods for Generating Three Dimensional Meshes”, *Computers in Mechanical Engineering*, pp.67-72, 1982

4. Cubit Online Users Documentation, version 4.0, http://endo.sandia.gov, accessed October 23, 2001.
5. Grover, B. T., *Surface Splicing: A Method of Quadrilateral Mesh Generation and Modification for Surfaces by Dual Creation and Manipulation,* Published Master's Thesis at Brigham Young University, 2002
6. Murdoch, P., *The Spatial Twist Continuum: A Dual Representation of the all Hexahedral Finite Element Mesh*, Published Doctoral Dissertation at Brigham Young University, 1995.
7. Tautges, T.J., Blacker, T.D., Mitchell, S.A., "The Whisker Weaving Algorithm: A Connectivity-Based Method for Constructing All-Hexahedral Finite Element Mehes", *International Journal for Numerical Methods in Engineering*, Vol. 39, pp.3327-3349, 1996