Mesh Matching - Creating Conforming Interfaces Between Hexahedral Meshes

Matthew L. Staten[1,2] Jason F. Shepherd[1], and Kenji Shimada[2]

[1] Sandia National Laboratories[†], P.O. Box 5800, Albuquerque, NM 87185, {mlstate,jfsheph@sandia.gov}
[2] Carnegie Mellon University, Pittsburgh, PA {mstaten,shimada@cmu.edu}

Summary. This paper presents a new method for handling nonconforming hexahedral-to-hexahedral interfaces. One or both of the adjacent hexahedral meshes are locally modified to create a one-to-one mapping between between the mesh nodes and quadrilaterals at the interface allowing a conforming mesh to be created. In the finite element method, nonconforming interfaces are currently handled using constraint conditions such as gap-elements, tied contacts, or multi-point constraints. By creating a conforming mesh, the need for constraint conditions is eliminated resulting in a smoother, more precise numerical solution. The method presented in this paper uses hexahedral dual operations, including pillowing, sheet extraction, dicing and column collapse operations, to affect the local mesh modifications. In addition, an extension to pillowing, called sheet inflation, is introduced to handle the insertion of self-intersecting and self-touching sheets. The quality of the resultant conforming hex mesh is high and the increase in number of elements is moderate.

1 Introduction

The finite element method is an indispensable part of the design through analysis process. Mesh generation is often a key bottleneck preventing broader use of the finite element method. The method utilized to handle interface conditions between assembly components often has a dramatic impact on the quality of the solution. Often two spatially adjacent geometric volumes must behave as a single component. Ideally, a conforming mesh will be created between components. A conforming mesh ensures a smooth and accurate interpolation of the solution to the governing equations over the interface, and also improves solution efficiency by minimizing the number of equations that

[†]Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000

must be solved. Although conforming meshes are preferable, nonconforming meshes are regularly encountered for a variety of reasons including:

1. Different engineers created the mesh on the different components.
2. The meshing algorithm used on the model did not honor boundary meshing constraints [1, 2].
3. Difficulties in generating the mesh required a different mesh topology on the interfacing surfaces for the two components. For example, during sweeping [3, 4], one interface surface may be required to be a linking-surface requiring a mapped mesh, while the other interface surface may be a source surface allowing a paved mesh.
4. The desired density of elements is different in the two components.

The current state of the art is to artificially constrain the nonconforming meshes with multi-point constraints, tied contacts, gap elements, etc. [5, 6, 7, 8] to maintain solution continuity across the interface. However, these methods typically result in solution quality degradation, disjoint solution fields, and/or adverse effects on solution convergence. Thus, these nonconforming interface conditions should only be used in non-critical regions of the model. A conforming interface would be preferred whenever possible.

In this paper, we present a new computational method, called mesh matching, which makes nonconforming hexahedral-to-hexahedral interface conditions conforming. This new method locally modifies the topology of the hexahedral elements in one or both of the adjacent hexahedral meshes to create a one-to-one pairing of nodes and quadrilaterals on the interface surfaces so that the meshes can be merged into a conforming mesh across the interface. As with any mesh modification procedure, the quality of the modified elements may be reduced from the initial mesh quality; however, assuming the element quality remains above prescribed element quality thresholds, the benefits of having a conforming mesh may compensate for the reduction in element quality.

This paper is organized as follows: Section 2 reviews existing hexahedral mesh topology and modification theory used during mesh matching. In Section 3 a new mesh topology operator is defined. In Section 4, the mesh matching algorithm is presented. In Section 5, two examples of mesh matching are provided. Finally, in Section 6, we provide some concluding remarks along with some areas of current and future efforts.

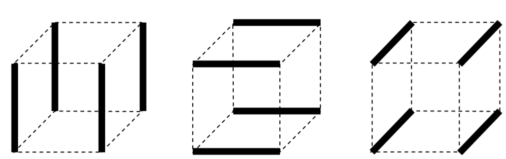

Fig. 1. A hexahedral element has 12 edges organized as 3 sets of 4 parallel edges.

2 Previous Research - The Hexahedral Mesh Dual

The dual of a hexahedral mesh [9, 10] is an alternate representation of the mesh composed of sheets and columns of hexahedral elements in the interior of the mesh, and chords and vertices on the boundary of the mesh.

Dual Sheets and Columns

Figure 1 illustrates how the 12 edges on a hexahedral element can be divided into three sets of four edges. The four edges in each set are topologically parallel to each other (i.e. do not share any nodes, but have one or more common adjacent hexahedra). Given one edge, the other three topologically parallel edges in each adjacent hexahedra can be identified. From each of these edges a similar set of topologically parallel edges can be recursively gathered from each of the adjacent hexahedra extending through the mesh. Thus, a dual sheet, S_i, can be defined as a set of topologically parallel edges. Alternatively, S_i can also be defined as the set of hexahedral elements traversed to build this set of edges. Figure 2b shows a single dual sheet uniquely defined by traversing starting from edge A in Figure 2a. A dual sheet is self-intersecting if any hexahedron in the sheet has more than one of its three edge sets in the definition of the sheet (Figure 2c). A dual sheet is self-touching if two or more edges defining the sheet use the same mesh node (Figure 9d and Figure 10c).

A hexahedral element contains six quadrilateral faces, grouped into three pairs of topologically opposite quadrilaterals. From a single quadrilateral, a column of hexahedra is defined by traversing adjacent hexahedra through their topologically opposite quadrilaterals. Thus a dual column, C_i, is defined as the set of topologically opposite quadrilaterals of adjacent hexahedra. Alternatively, C_i is defined as the set of hexahedral elements traversed to locate this set of quadrilaterals. Figure 3b illustrates the dual column defined by quadrilateral face A specified in Figure 3a. An important link between sheets and columns is that a column defines the intersection of two sheets (Figure 3c). A column is self-intersecting if any hexahedron in the column has more than one of its quadrilateral pairs in the definition of the column (Figure 3d).

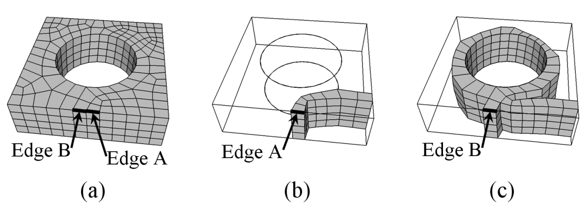

Fig. 2. Hexahedral Dual Sheets: (a) A simple mesh with two edges, A and B, identified. (b) The dual sheet uniquely identified by edge A. (c) A self-intersecting sheet identified by edge B.

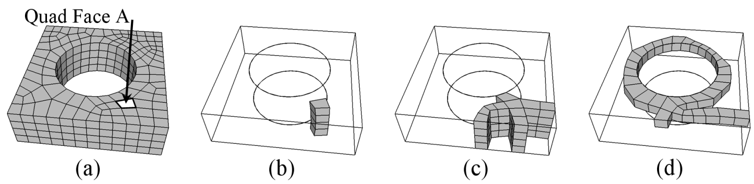

Fig. 3. Hexahedral Dual Columns: (a) Hex mesh with highlighted quadrilateral A. (b) The column of hexahedra defined by quadrilateral A. (c) The sheets which intersect to form the column in (b). (d) A self-intersecting column.

Dual on the Boundary of a Hexahedral Mesh

The boundary of a mesh is the set of quadrilaterals which have exactly one adjacent hexahedron. These quadrilaterals can be grouped based on their associated geometric surface. Quadrilateral meshes have a dual representation of dual chords and vertices. The four edges on a quadrilateral are grouped into two pairs of topologically opposite edges. A dual chord is uniquely defined starting from a single edge and traversing adjacent quadrilaterals through opposite edges (Figure 4). This process is repeated until every edge in the quadrilateral mesh has been associated with a dual chord. Thus, a dual chord, c_i, can be defined as a set of the topologically opposite edges. Alternatively, c_i can also be defined as the quadrilaterals that were traversed to build this set of edges. Finally, a dual chord c_i can also be defined as the collection of line segments connecting the centroids (dual vertices, v_i) of this set of quadrilaterals. A dual chord is self-intersecting if any quadrilateral in the chord has all four of its edges in the definition of the chord. Associated with each dual chord, c_i, is the dual sheet, S_i, defined by traversing topologically parallel edges from any edges in c_i. Likewise, associated with each dual vertex, v_i, is a dual column, C_i, defined by traversing topologically opposite quadrilaterals from the quadrilateral associated with v_i.

2.1 Dual Topological Operators

The matching procedure described in Section 4 performs a series of topological operations on hexahedral dual sheets and columns. Sheet extraction [11] removes a dual sheet by collapsing all edges that define it, reducing it to a continuous set of quadrilateral faces (Figure 5). Any sheet topology, including self-intersecting and self-touching sheets, can be extracted. Sheet extraction is not always possible due to geometric nodal associativity. That is, when collapsing the edges that define a sheet, the two nodes on each edge are merged. If two edge nodes have conflicting geometric associativities, the edge cannot be collapsed. However, the sheet can be extracted if preceded by a sheet insertion to add sufficient mesh topology. In addition, low node valency in the region of the sheet extraction can sometimes lead to doublets [12], resulting in ill-shaped elements with zero of negative scaled Jacobians [13].

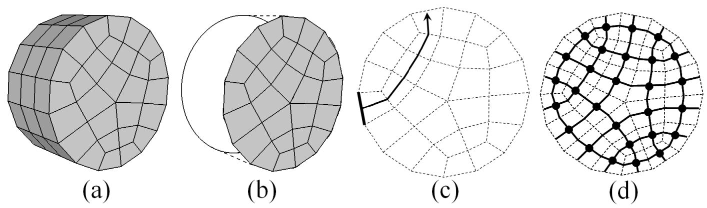

Fig. 4. Boundary quad meshes and their dual: (a) A hexahedral mesh. (b) One boundary surface mesh. (c) A single dual chord. (d) The complete surface dual.

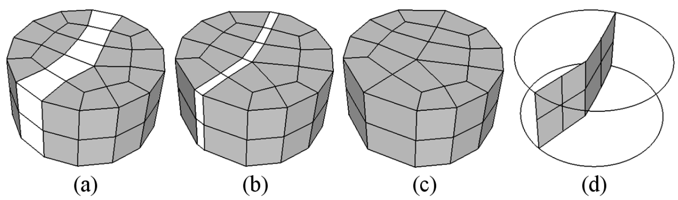

Fig. 5. Hexahedral sheet extraction: (a) A hexahedral mesh with one dual sheet. (b) The edges are collapsed to extract the sheet. (c) The sheet is extracted. (d) The sheet becomes a continuous set of quadrilaterals.

Pillowing [10, 12, 14] is a method of inserting new dual sheets into a mesh. Pillowing is performed by identifying a set of hexahedral elements as the *shrink set* (Figure 6a). The hexahedra in the shrink set are separated from the remainder of the mesh by a 'shrink' distance, allowing the placement of a new sheet between the shrink set and the other hexahedra in the mesh (Figures 6b and 6c). The new sheet is always non-self-intersecting and non-self-touching. In contrast to sheet extraction, pillowing is always possible given a well-defined shrink set. Since its introduction by Mitchell [12], it has been applied in many mesh modification procedures [10, 15, 16, 17, 18].

Dicing [19] is another method of inserting dual sheets into a mesh. Dicing is performed by splitting the edges that define the sheet. Dicing can insert multiple sheets at once by splitting each edge multiple times (Figure 7). The new sheets inserted with dicing are duplicates of the input sheet; if the input sheet is self-intersecting, the new sheets will be also. The disadvantage of dicing is that it can only copy existing sheets; it cannot create new sheets that did not already exist in the mesh. In addition, dicing cannot create self-touching sheets.

The column collapse operation is also an important dual operation (Figure 8). A column is collapsed by merging one pair of opposite nodes of each quadrilateral defining the column. As described in Section 2, a dual column defines the intersection of two dual sheets. This intersection is removed by collapsing the column. In addition, the paths of the two sheets is altered. Collapsing self-intersecting columns creates doublets [12] and should be avoided.

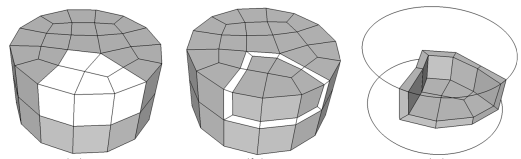

Fig. 6. Pillowing: (a) A hexahedral mesh with a shrink set of five hex elements identified. (b) A pillow is inserted. (c) The pillow sheet.

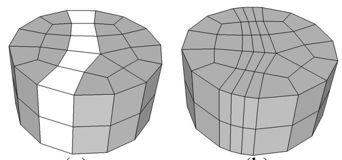

Fig. 7. Dicing: (a) A hexahedral mesh with one dual sheet indicated. (b) The sheet is diced into three sheets.

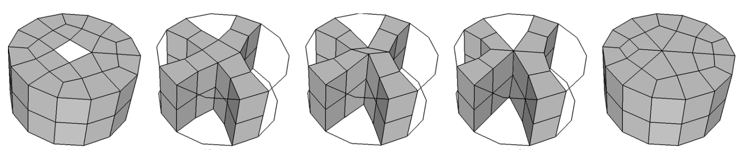

Fig. 8. Column collapse: (a) A hexahedral mesh with one dual column indicated. (b) The sheets that intersect to define the column drawn separate from the mesh. (c) Opposite nodes are merged to collapse the column. (d) The sheets defining the column no longer intersect. (e) The entire mesh after the column collapse.

3 Sheet Inflation - Generalized Sheet Insertion

The mesh matching algorithm presented in Section 4 requires the ability to insert any kind of sheet including both self-intersecting and self-touching sheets. Previous research allows the insertion of sheets through pillowing and dicing. However, neither pillowing nor dicing can insert self-touching sheets, and pillowing is unable to insert self-intersecting sheets. Dicing can insert self-intersecting sheets, but only if an existing self-intersecting sheet exists in the correct location of the mesh. Hence, a sheet insertion operator which inserts both self-intersecting and self-touching sheets is required.

Sheet inflation can be thought of as the reverse of sheet extraction. In sheet extraction, a dual sheet is reduced to a continuous set of quadrilaterals. This process can be reversed by *inflating* the quadrilaterals to re-introduce the extracted sheet. Knupp [14] introduced a similar operator with the inflate hex ring which inflates a set of quadrilaterals into a new dual column. For sheet inflation, the boundary of the quadrilateral set must lie on the boundary of the hex mesh. Self-intersecting and self-touching sheets are inserted by inflating quadrilateral sets with non-manifold edges. A set of non-manifold edges with four adjacent quadrilaterals (i.e. 4NMEsets) can be inflated as either a self-intersecting (Figure 9c) or a self-touching sheet (9d). A a set of non-manifold edges with three adjacent adjacent quadrilaterals (i.e. 3NMEsets) can be inflated as either self-touching (Figure 10c), or self-touching and self-intersecting (Figure 10d). Thus each non-manifold edge set can be inflated in two different ways. The input to sheet inflation requires each non-manifold edge set have a flag indicating which option should be performed. 3NMEsets must appear in the quadrilateral set in pairs, or be paired with a boundary.

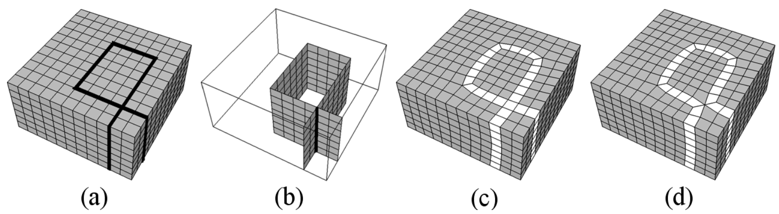

Fig. 9. Sheet inflation example #1: (a) A simple hex mesh with boundary edges indicated. (b) A non-manifold continuous set of quadrilaterals bounded by the indicated boundary edges in (a). The highlighted edges form a 4NMEset. (c) A self-intersecting sheet inflation option. (d) A self-touching sheet inflation option.

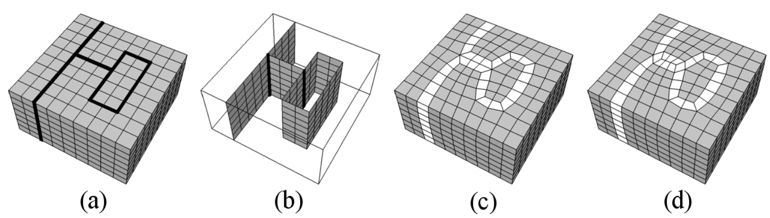

Fig. 10. Sheet inflation example #2: (a) A simple hex mesh with boundary edges indicated. (b) A non-manifold continuous set of quadrilaterals bounded by the indicated boundary edges in (a). The highlighted edges form two 3NMEsets. (c) A self-touching sheet inflation option. (d) A self-touching and self-intersecting sheet inflation option.

For manifold sets of quadrilaterals, sheet inflation is the same as pillowing the hexahedra on one side of the quadrilaterals set. Thus, sheet inflation can be implemented in a manner similar to pillowing, with the following three differences caused by the non-manifold edge sets:

1. Multiple shrink sets are required, partitioned from each other by the non-manifold edge sets.

2. Nodes along non-manifold edge sets must be duplicated either twice for self-touching sheets, or three times for self-intersecting sheets.
3. Quadrilaterals which lie between two 3NMEsets must be duplicated twice, and the resulting gap is filled with two hexahedra instead of one.

4 Hexahedral Mesh Matching

4.1 Mesh Matching Input Requirements

The input requirements of the mesh matching algorithm are:

1. Two geometric surfaces, A and B, that are:
   a) Topologically identical (The number of boundary curves, loops, and vertices defining the two surfaces must be the same),
   b) Geometrically similar (each boundary curve/vertex on Surface A must have a corresponding boundary curve/vertex on Surface B that is within a tolerance, β), and
   c) Both adjacent to hexahedral mesh elements (could be different surfaces on a single conforming mesh, or surfaces on separate meshes).
2. An integer value for a *depth* parameter indicating how many layers into the adjacent hexahedral meshes the modifications can propagate.
3. A flag indicating which (or both) of the surfaces can have their mesh topology modified. The simplest case is that both meshes can be modified, but all changes can be done on one side of the interface if necessary.

If input requirement 1a or 1b are not met by the initial hexahedral meshes, the Graft Tool [18] can be used to imprint the boundaries of the interface surfaces onto each other.

4.2 Mesh Matching Procedure

Figure 11a illustrates a two-volume model positioned such that Surface A on Volume A overlaps exactly with Surface B on Volume B meeting the input requirements in Section 4.1. However, as seen in Figures 11b and 11c, the quadrilateral meshes on Surfaces A and B do not match. In this case, the nonconforming mesh was created because the topology of Volume B requires Surface B to be a linking surface for sweeping, while the topology of Volume A requires Surface A to be a source surface for sweeping. The resulting mesh is nonconforming as shown in Figure 11d and 11e. The objective is to modify the mesh topology of one or both of the adjacent hexahedral meshes such that the quadrilateral meshes on Surfaces A and B match, node-for-node and quad-for-quad. The node pairs can then be merged resulting in a conforming mesh across the interface.

We make the following assertion:

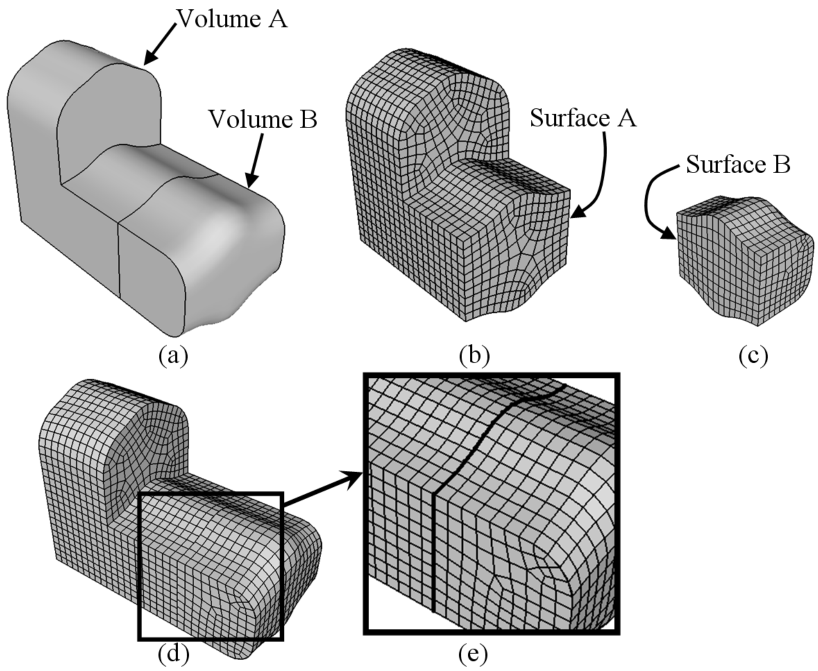

Fig. 11. Nonconforming mesh example: (a) Geometric model composed to two volumes, A and B. (b) The mesh on Volume A. (c) The mesh on Volume B. (d) The complete mesh showing the discontinuity in the mesh. (e) Zoom-in of mesh discontinuity.

Assertion 1: *Given two geometric surfaces, A and B, which meet the input requirements stated in Section 4.1, the topology of the two adjacent hexahedral meshes can always be modified to create identical quadrilateral mesh topology on both Surfaces A and B, which can then be merged to create a conforming hexahedral mesh across the interface.*

Rationale: Two quadrilateral meshes will be topologically identical *iff* the duals of the two quadrilateral meshes are identical. Let Ω_{cA} and Ω_{cB} be the sets of chords, c_i, in the quadrilateral meshes on Surfaces A and B respectively. Initially $\Omega_{cA} \neq \Omega_{cB}$. However, through sheet insertion and extraction, dual chords can be inserted and extracted from boundary quadrilateral meshes. Thus, one or both of Ω_{cA} and Ω_{cB} can be modified such that they do match. The algorithm of mesh matching is then:

1. For each $c_i \in \Omega_{cA}$:
   a) Search for a $c_j \in \Omega_{cB}$ such that $c_j = c_i$ within a tolerance, δ. If an equal c_j is found, insert c_i and c_j into $\Omega_{c-pairs}$, else, insert c_i into $\Omega_{cA-unmatched}$.
2. For each $c_j \in \Omega_{cB}$:
   a) If $c_j \notin \Omega_{c-pairs}$ insert c_j into $\Omega_{cB-unmatched}$.

3. For each $c_i \in \Omega_{cA-unmatched}$:
   a) Use the following rules to determine if c_i should be inserted into Ω_{cB} or extracted from Ω_{cA}:
      i. No c_i should be extracted if doing so violates geometric associativity (see Section 2.1).
      ii. No c_i should be extracted if doing so creates doublet topology.
      iii. Sheet insertions should be done if mesh density has already been decreased bordering upon acceptability thresholds.
      iv. Sheet extractions should be done if mesh density has already been increased bordering upon acceptability thresholds.
      v. If insertion is done:
         A. Project c_i onto Surface B, and find a continuous set of mesh edges, E_B, on Surface B which approximate this projection.
         B. Use dicing if $\exists\ c_j \in \Omega_{cB}$ such that $c_j = E_B$ with a tolerance, γ, where $\gamma >> \delta$.
         C. Use pillowing if a well-connected shrink set can be defined behind the quadrilaterals on one side of E_B.
         D. Otherwise, use sheet inflation on a continuous quadrilateral set which contains E_B on its boundary.
      vi. If extraction is done, consider doing a column collapse (see example 1 in Section 5.1) in order to avoid global changes.
   b) If c_i is to be extracted, remove c_i from Ω_{cA} and $\Omega_{cA-unmatched}$
   c) If c_i is to be inserted, remove c_i from $\Omega_{cA-unmatched}$, and insert it along with the newly inserted chord into $\Omega_{c-pairs}$.
4. Repeat Step 3 for each $c_j \in \Omega_{cB-unmatched}$
5. Smooth all nodes local to the interface surface modifications to improve element quality [21].

The Chord Equals Operator

In order to perform Step 1a and Step 3avB, $c_i = c_j$ must be defined:

Definition 1. *$c_i = c_j$, within a tolerance, δ, iff the maximum distance between c_i and all v_k in c_j is less than δ AND the maximum distance between c_j and all v_k in c_i is also less than δ.*

$c_i = c_j$ does not require c_i and c_j to have the same number of dual vertices. Rather, $c_i = c_j$ if the two chords are spatially close to each other. For example, in Figure 12 the two indicated chords have 16 and 15 chord vertices. However, when the two surfaces are overlaid, the maximum distance between the two chords is less than δ. Thus, these two chords are equal.

In short, mesh matching implements Shepherd's [20] 2^{nd} assertion which states: *There exists a transformation that converts one set of fundamental sheets into an alternative set of fundamental sheets.* This assertion can be generalized as follows:

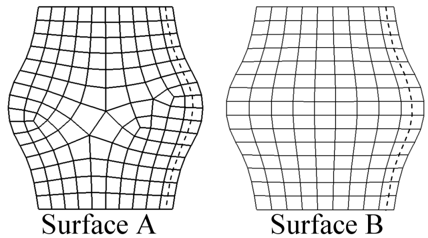

Fig. 12. The mesh topology of example Surfaces A and B. Although the indicated chords have different number of dual vertices, they are still spatially within δ when overlaid.

Assertion 2: *There exists a transformation that converts one hexahedral mesh into any other hexahedral mesh on a given geometry.*

Rationale: Any sheet can be extracted from a mesh. If geometric associativity would be violated by sheet extraction, the sheet can be extracted after a sheet insertion to adequately add the appropriate mesh topology. In addition, any sheet can be inserted, including self-touching and self-intersecting sheets. Thus any sheet not matching the goal topology can be extracted, and any missing sheet can be inserted. In the case of mesh matching, the hexahedral mesh to convert to is a mesh which matches across the interface.

5 Examples

5.1 Simple Example

We now illustrate mesh matching on the simple example from Figure 11. Figure 13a identifies a chord, c_i, in Surface A, which has no pair in Surface B. In Figure 13b, a string of edges, E_B, on Surface B is identified which roughly matches the projection of c_i. E_B partitions the surface quadrilaterals into two sets, of which, one is chosen (normally the smaller set). A pillow shrink set is then defined as the hexahedral elements behind the chosen quadrilateral set. The input *depth* parameter is used to determine how far into the volume to propagate to build the shrink set. Figures 13c and 13d show the mesh after the pillow is inserted using depth=2, followed by appropriate smoothing [21]. The resulting new chord in Surface B is then paired with the identified unpaired chord in Figure 13a. Figure 14 repeats this process for another unpaired chord in Surface A.

Figure 15 illustrates the use of dicing to introduce topology required for mesh matching. In Figure 15a, three topologically parallel chords are indicated. One of these three chords is paired with the chord indicated in Figure 15b, which is diced, followed by smoothing, introducing the required topology to match all three chords. After these three operations, the topology on the left side of Surface B is beginning to match the topology on the left side of Surface A. Additional sheets are inserted and extracted until the mesh topology on these two surfaces matches.

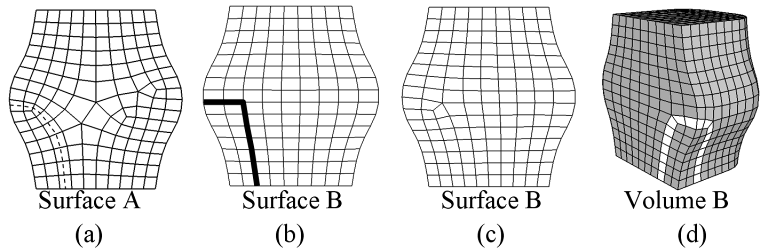

Fig. 13. Pillowing during mesh matching: (a) The mesh topology on Surface A with one non-paired chord, c_i, indicated. (b) The mesh topology on Surface B with a string of mesh edges, E_B, indicating where pillowing will be performed. (c) Surface B after pillowing is performed. (d) Volume B after pillowing is performed.

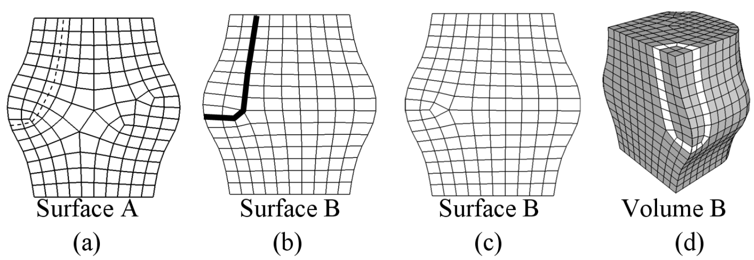

Fig. 14. Pillowing during mesh matching: (a) The mesh topology on Surface A with one non-paired chord, c_i, indicated. (b) The mesh topology on Surface B with a string of mesh edges, E_B, indicating where the 2nd pillow will be inserted. (c) Surface B after pillowing is performed. (d) Volume B after pillowing is performed.

Sheet Extraction for Mesh Matching

Although all required topology can be introduced with sheet insertion, doing so will have the potentially undesirable side-effect of increasing the density of the mesh local to the interface surfaces. Sheet extraction is useful in reducing or eliminating the increase in mesh density. For example, Figure 16a shows the mesh topology on Surface A with one unpaired chord indicated. Figure 16b shows the mesh topology of Surface B; clearly the chord indicated in Figure 16a has no match in Surface B. Figure 16c shows the mesh in Volume A. The indicated sheet is extracted from the mesh as shown in Figure 16d which removes the unpaired chord in 16a from Surface A as shown in Figure 16e.

One potentially undesirable side-effect of sheet extraction is that the entire sheet must be extracted in order to maintain a conforming all-hexahedral

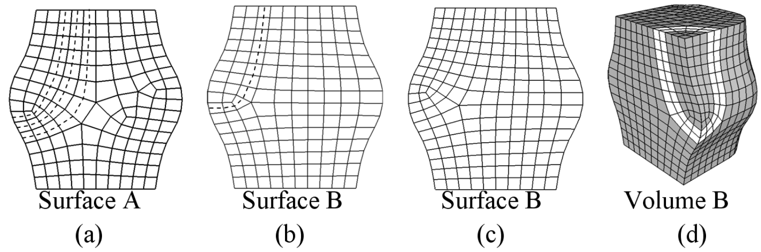

Fig. 15. Dicing during mesh matching: (a) The mesh topology on Surface A with three chords indicated. (b) The mesh topology on Surface B with a single chord to be diced indicated. (c) Surface B after dicing is performed. (d) Volume B after dicing is performed.

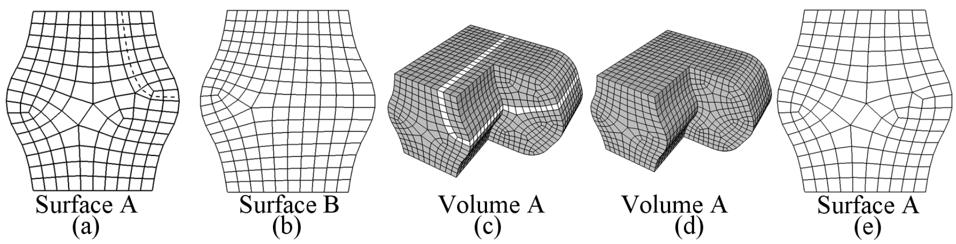

Fig. 16. Sheet extraction during mesh matching: (a) The mesh topology on Surface A with one unpaired chords indicated. (b) The mesh topology on Surface B. (c) Volume A with the sheet extending from the chord indicated in (a) highlighted. (d) Volume A after indicated sheet is extracted. (e) Surface A the sheet is extracted.

mesh. Figure 16c clearly shows that the sheet to be extracted extends far away from the interface surfaces, resulting in a global change. However, the changes can be kept local to the region around Surface A if we first perform a column collapse operation (see Section 2.1). For example, in Figure 17a one additional sheet, which remains local to the interface surfaces, is identified. If such a local sheet does not exist in the mesh, one can be inserted by pillowing a few layers of hexahedra away from the interface surfaces. As described in Section 2, the two sheets indicated in Figure 17a intersect in a column of hexahedra. By collapsing this column, we redirect the sheet to extract in such a way that it now remains local to the interface surfaces as illustrated in Figure 17b. The extraction sheet can then be extracted as illustrated in Figure 17c keeping all changes local to the interface surfaces.

The pillow, sheet inflation, sheet extraction, dicing and column collapse operations can be applied repeatedly until the topology on the interface surfaces matches allowing the mesh to be merge into a single conforming mesh. Figure 18a shows the final mesh with the final interface quadrilateral mesh shown in Figure 18b. Table 1 shows the element counts and element quality before and after mesh matching. As with any hexahedral mesh modification, mesh matching introduces irregular nodes into the mesh topology which will

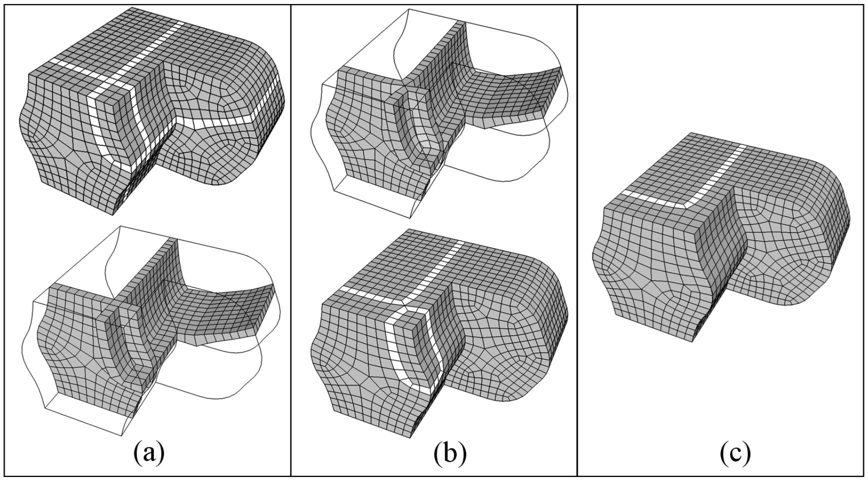

Fig. 17. Localized sheet extraction during mesh matching: (a) The mesh in Volume A with two sheets identified, the one to extract, and one that remains local to the interface surface. (b) The column where the two indicated sheets intersect has been collapsed, redirecting the sheet to extract to remain local to the interface region. (c) After sheet extraction; all changes are local to interface surfaces.

tend to decrease element quality. In this case, the resulting mesh has a minimum scaled Jacobian [13] of 0.5335, which is still well suited for analysis.

In the case that one side of the interface cannot be changed, the dicing, pillowing, and sheet extraction operations are restricted to be performed only where changes are allowed. For example if Surface A cannot be modified, all unpaired chords in Surface B are removed through sheet extraction in Volume B. Likewise, any unpaired chords in Surface A are inserted into Surface B through sheet insertion in Volume B. Thus, Surface and Volume A remain unchanged. If extraction of a sheet is required that would result in invalid geometric associativity, this sheet can be redirected before the extraction using a column collapse thus allowing it to be extracted.

Table 1. Element Quality Results for Simple Example Model

	Number of Elements	Minimum Scaled Jacobian
Before Matching	5,925	0.7334
After Matching	6,024	0.5335

5.2 Industry Example

Figure 19a shows an I-beam structure used in a civil engineering application. The critical component to be analyzed is the diagonal stiffener. In fact, as part of the analysis, several different designs of the stiffener as well as adaptive studies using different size elements will be used. The mesh on the rectangular

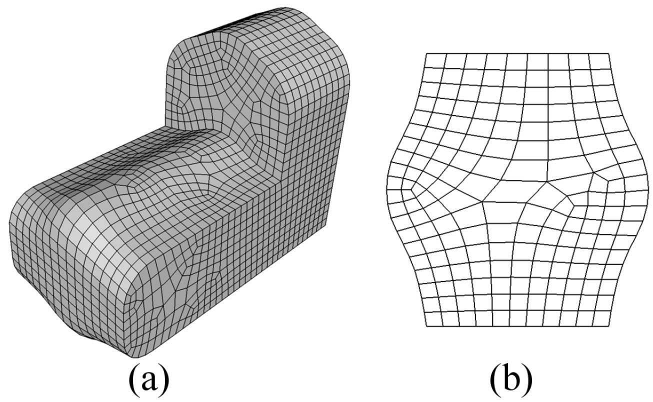

Fig. 18. Final mesh after mesh matching: (a) Final conforming mesh on example model after all topology on interface surfaces was matched. (b) The final mesh topology on the interface surface.

I-beams has approximately one million hexahedral elements, and required significant effort to generate. Ideally each time a new stiffener is introduced, the existing mesh on the rectangular I-beam structure can be re-used rather than requiring it to be re-meshed. The concepts of mesh matching presented in this paper apply to this application since every time a new mesh is generated for the stiffener, the mesh matching algorithm can be run on the connection with the rectangular I-beam structure to create a conforming mesh.

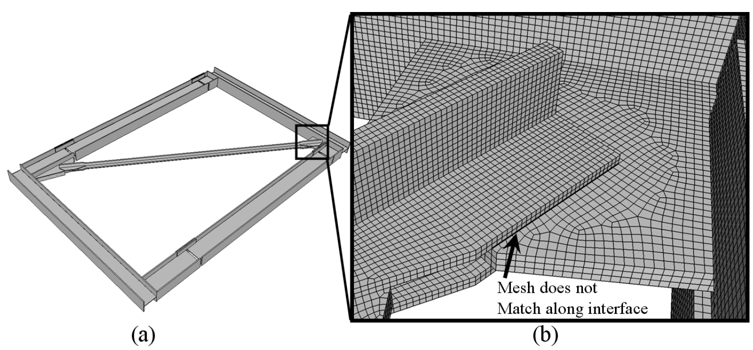

Fig. 19. Industrial example: (a) An I-beam structure for a civil engineering application. (b) Close look at interface between diagonal stiffener and corner I-Beams. The meshes do not matches at the interface. This model provided courtesy of Tyler Josephson and Professor Paul Richards from the Civil and Environmental Engineering Department at Brigham Young University.

Figure 19b shows a close up of the corner of the structure where the stiffener connects to the corner plate. The stiffener is meshed with hexahedral elements that are slightly smaller than that of the corner plate. As a result, we have a nonconforming mesh. Figure 20a shows the interface surface on

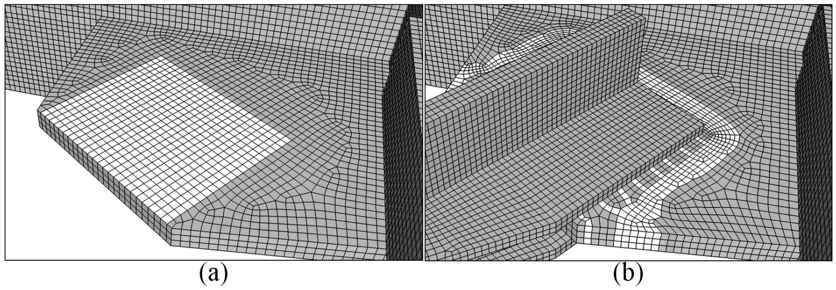

Fig. 20. Industrial example: (a) The mesh on the interface surface on the corner plate. (b) The mesh after mesh matching. The mesh topology was matched and merged into a single all-hexahedral mesh. The highlighted elements indicate the sheets that were inserted during mesh matching.

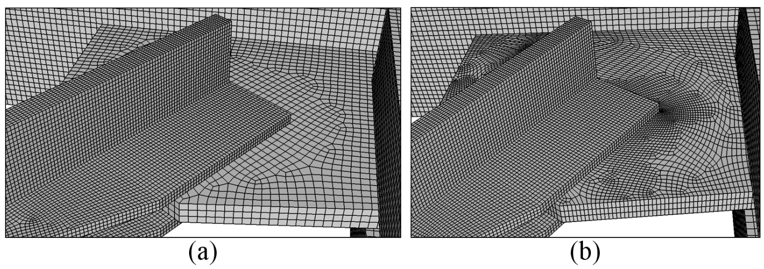

Fig. 21. Industrial Example: (a) The same I-beam model before mesh matching, this time stiffener is meshed at a much higher density. (b) After mesh matching.

Table 2. Element Quality Results for I-beam Model before and after mesh matching.

	Number of Element in Corner Plate	Minimum Scaled Jacobian
Before Mesh Matching	4,611	0.6624
After Mesh Matching - course Stiffener	7,221	0.5872
After Mesh Matching - fine Stiffener	17,883	0.4924

the corner plate. It is meshed with 21×24 mapped quadrilateral mesh. The interface surface on the stiffener is meshed with a 29×30 mapped quadrilateral mesh. The mesh matching algorithm will need to increase the density of elements in the corner plate so that it also has a 29×30 mapped quadrilateral mesh so that the mesh on the interface can be merged. Figure 20b illustrates the mesh after successfully creating a conforming mesh using mesh matching.

Figure 21a shows the same I-beam model, however, the stiffener has been meshed at a much higher density of elements. The element size difference between the stiffener and the corner plate is 2.3 to 1. Rather than remeshing the rectangular I-beam structure mesh matching is used to enforce a conforming mesh across the interface. Figure 21b shows the mesh after mesh matching has successfully matched the mesh topology on the interface creating a conforming all-hexahedral mesh through the interface. Table 2 shows the element count and element qualities before and after mesh matching. As with the first example, although element quality is reduced some by mesh matching, the resulting element qualities are still suitable for analysis.

6 Conclusion

A new computational method, called mesh matching, for making nonconforming hexahedral-to-hexahedral mesh interfaces conforming has been presented. Mesh matching modifies the mesh topology local to nonconforming hexahedral-to-hexahedral interfaces to create a conforming mesh interface. Mesh matching eliminates the need for artificial constraint conditions, such as tied contacts, gap elements, and multi-point constraints. Mesh topology is modified using the dual operators of column collapse, dicing, pillowing, and sheet extraction, along with a new operator, sheet inflation.

Many meshing algorithms require entire assemblies to be meshed at once in order to have conforming meshes between components [1, 2]. Mesh matching relaxes this requirement by creating conforming meshes between assembly components after each component is meshed individually. Further, Tautges [22] indicates that hex meshing would be greatly simplified if global coupling between assemblies could be reduced or eliminated. Mesh matching reduces global coupling by enforcing a conforming mesh after the initial meshes have been created. Thus, mesh matching has the potential to greatly simplify the generation of conforming assembly hexahedral meshes.

Research on mesh matching continues with focus on the following areas. First, mesh matching requires repeated dicing, pillowing, sheet extraction, and sheet inflation. The example in Figure 21 required more than 50 pillowing operations. Manually specifying 50 shrink sets is a tedious task that is best automated. Second, mesh matching requires the interface be limited to two topologically identical and spatially similar surfaces. In practice, interface conditions often include multiple adjacent surfaces on each component. Compared to a serial approach, simultaneous matching of all interface surfaces will likely result in a more optimal mesh. Third, mesh matching theory depends upon *Assertion 1* in Section 4.2. Future research to prove this assertion will guarantee the usefulness of mesh matching. The unproven elements of *Assertion 1* involve guaranteeing that sheet inflation is always possible for any possible hexahedral topology.

References

1. Yongjie Zhang, C. L. Bajaj, B.S. Sohn, 3D Finite Element Meshing from Imaging Data. *The special issue of Computer Methods in Applied Mechanics and Engineering (CMAME) on Unstructured Mesh Generation* 2005; **194**: no. 48-49 : 5083-5106.
2. Matthew L. Staten, R. A. Kerr, S. J. Owen, T. D. Blacker, Unconstrained Paving and Plastering: Progress Update. *Proceedings 15th International Meshing Roundtable* 2006; 469-486.

3. T. D. Blacker, The Cooper Tool. *Proc. 5th International Meshing Roundtable* 1996; 13-29.
4. Matthew L. Staten, S. A. Canann, S. J. Owen, BMSweep: Locating Interior Nodes During Sweeping. *Engineering with Computers* 1999; **15**: 212-218.
5. C. R. Dohrmann, S. W. Key, A Transition Element for Uniform Strain Hexahedral and Tetrahedral Finite Elements. *International Journal for Numerical Methods in Engineering* 1999; **44**: 1933-1950.
6. David M. Trujillo, C. G. Pappoff, A General Thermal Contact Resistance Finite Element. *Finite Elements in Analysis and Design* 2002; **38**:3, 263-276.
7. C. R. Dohrmann, S. W. Key, M. W. Heinstein, Methods for Connecting Dissimilar Three-Dimensional Finite Element Meshes. *International Journal for Numerical Methods in Engineering* 2000; **47**: 1057-1080.
8. Michael A. Puso, T. A. Laursen, Mesh Tying on Curved Interfaces in 3D. *Engineering Computations* 2003; **20**:3, 305-319.
9. Peter Murdoch, S. E. Benzley, The Spatial Twist Continuum. *Proc. 4th International Meshing Roundtable* 1995; 243-251.
10. Karl Merkley, C. Ernst, J. F. Shepherd, M. J. Borden, Methods and Applications of Generalized Sheet Insertion for Hexahedral Meshing. *Proc. 16th International Meshing Roundtable* 2007; 233-250.
11. Michael J. Borden, S. E. Benzley, J. F. Shepherd, Hexahedral Sheet Extraction. *Proc. 11th International Meshing Roundtable* 2002; 147-152.
12. Scott A. Mitchell, T. J. Tautges, Pillowing Doublets: Refining a Mesh to Ensure That Faces Share At Most One Edges. *Proc. 4th International Meshing Roundtable* 1995; 231-240.
13. Patrick M. Knupp, Algebraic Mesh Quality Measures. *SIAM Journal on Scientific Computing* 2001; **23**:193-218.
14. Patrick M. Knupp, S. A. Mitchell, Integration of Mesh Optimization with 3D All-Hex Mesh Generation, SAND99-2852, Sandia National Laboratories, 1999.
15. Michael Borden, S. E. Benzley, S. A. Mitchell, D. R. White, R. Meyers, The Cleave and Fill Tool: An all-Hexahedral Refinement Algorithm for Swept Meshes. *Proc. 9th International Meshing Roundtable* 2000; 69-76.
16. Nathan J. Harris, S. E. Benzley, S. J. Owen, Conformal Refinement of All-Hexahedral Element Meshes Based on Multiple Twist Plane Insertion. *Proc. 13th International Meshing Roundtable* 2004; 157-167.
17. Ko-Foa Tchon, J. Dompierre, R. Camarero, Conformal Refinement of All-Quadrilateral and All-Hexahedral Meshes According to an Anisotropic Metric. *Proc. 11th International Meshing Roundtable* 2002; 231-242.
18. Steve R. Jankovich , S. E. Benzley, J. F. Shepherd, S. Mitchell, The Graft Tool: an All-Hexahedral Transition Algorithm for Creating a Multi-Directional Swept Volume Mesh. *Proc. 8th International Meshing Roundtable* 1999; 387-392.
19. Darryl J. Melander, T. J. Taugtes, S. E. Benzley, Generation of Multi-Million Element Meshes for Solid Model-Based Geometries: the Dicer Algorithm. *AMD - Trends in Unstructured Mesh Generation* 1997; **220**: 131-135.
20. Jason Shepherd, Topologic and Geometric Constraint-Based Hexahedral Mesh Generation. *Dissertation - University of Utah*, 2007.
21. Patrick M. Knupp, A Method for Hexahedral Mesh Shape Optimization. *International Journal for Numerical Methods in Engineering* 2003; **58**:319-332.
22. Timothy J. Tautges, The Generation of Hexahedral Meshes for Assembly Geometry: Survey and Progress. *International Journal for Numerical Methods in Engineering* 2001; **50**:2617-2642;