Steven E. Benzley
e-mail: seb@byu.edu

Nathan J. Harris
e-mail: njh33@byu.edu

Michael Scott
e-mail: mas88@byu.edu

Brigham Young University, Provo, UT

Michael Borden
e-mail: mborden@sandia.gov

Steven J. Owen
e-mail: sjowen@sandia.gov

Sandia National Laboratories,
Albuquerque, NM

Conformal Refinement and Coarsening of Unstructured Hexahedral Meshes

This paper describes recently developed procedures for local conformal refinement and coarsening of all-hexahedral unstructured meshes. Both refinement and coarsening procedures take advantage of properties found in the dual or "twist planes" of the mesh. A twist plane manifests itself as a conformal layer or sheet of hex elements within the global mesh. We suggest coarsening techniques that will identify and remove sheets to satisfy local mesh density criteria while not seriously degrading element quality after deletion. A two-dimensional local coarsening algorithm is introduced. We also explain local hexahedral refinement procedures that involve both the placement of new sheets, either between existing hex layers or within an individual layer. Hex elements earmarked for refinement may be defined to be as small as a single node or as large as a major group of existing elements. Combining both refinement and coarsening techniques allows for significant control over the density and quality of the resulting modified mesh. [DOI: 10.1115/1.2052848]

1 Introduction

Both tetrahedral (Tets) and hexahedral (Hexes) elements are effectively used to model three-dimensional objects for finite element analysis. Tets can be the element of choice because of the robustness of their modeling capabilities for any general shape. Hexes can be the element of choice for their abilities to provide more efficiency and accuracy in the computational process. Techniques for mesh refinement and coarsening are generally more robust for Tets as opposed to Hexes. Straightforward Tet refinement schemes, based on longest-edge division, as well as the extension to include adaptive derefinement to coarsen a refined mesh, are described by Carey and Plaza [1,2]. In addition, Plaza and Rivara have recently presented an iterative longest-edge refinement scheme that reduces the average propagation path of a refinement [3]. De Cougny and Shephard provide a detailed description of both refinement and coarsening of Tets as applied to parallel computations [4]. Further, several authors have found success with an efficient "red green" mesh subdivision algorithm for tetrahedra [5–7].

Less attention has been given to the modification of all-hex meshes. Methods using iterative octrees [8] have been proposed, but these methods result in nonconformal elements that cannot be accommodated by some solvers. Other techniques insert non-hex elements that result in hybrid meshes or require uniform dicing to maintain a consistent element type [9]. Schneiders' directional refinement method [10] produces a conformal mesh by pillowing layers in alternating i, j, and k directions but relies upon a Cartesian initial octree hex mesh. A slight modification to Schneiders' approach was given by Kwak and Im [11]. Yet another method, as proposed by Li and Cheng [12], begins with a mapped definition of a six-sided cube and selectively inserts 3D transition templates at the corners or along the edges. This method is restricted to solids that initially conform to traditional mapping schemes. The 3D anisotropic refinement scheme presented by Tchon et al. [13] expands Schneiders' multi-directional refinement to initially unstructured meshes by pillowing layers of elements without the use of octrees. This method is quite robust but does not offer the capability to refine mesh regions around individual nodes, element edges or element faces. The refinement process developed in this work is not dependent upon initial mapped or octree-based grids and also allows more flexibility in regions of the mesh that are smaller than entire hex elements or are more easily described specifically around individual nodes, element edges and element faces [14].

This paper describes general methods for local conformal refinement and coarsening all-hexahedral unstructured meshes. Both refinement and coarsening procedures presented take advantage of properties found in the dual of an all-hexagonal mesh. The properties of the dual are explained in the next section.

2 Background

A hexahedral element can be viewed as a volume with three sets of opposing faces. In valid conformal meshes, two neighboring hexahedrons share exactly one face. Lining hexahedral elements up so that each element has two neighboring elements that are attached to opposing faces will create a stack of hexahedral elements. A cord in the dual of the mesh is used to represent these stacks (see Fig. 1) [15].

Multiple stacks can then be grouped together to create a layer of elements. Each element in a layer has four neighboring elements that are attached to two orthogonal pairs of opposing faces. The shaded area of Fig. 2 shows how a sheet in the dual of the mesh can represent these layers. The two-dimensional sheet in the dual is called a twist plane [16], and the elements that define this twist plane are called a hex sheet.

Conformal all-hex meshes are assemblages of multiple hex sheets. Figure 3 shows a hex meshed cylinder on the left and its corresponding dual on the right. The complete dual is referred to as the spatial twist continuum. The figure clearly shows how hex sheets interact together to define the completed mesh. These hex sheets provide a basis for both conformal refining and coarsening of all-hexahedral meshes.

3 Coarsening

3.1 Global 3D Coarsening. Figure 4 shows a meshed object with a specific hex sheet highlighted in (a) and shown as a complete sheet in (b). To coarsen a mesh, a hex sheet can be removed and the gap remaining is then closed conformally by collapsing the opposing hex faces across the gap. Note that upon collapsing

Contributed by the Engineering Simulation and Visualization Committee of ASME for publication in the Journal of Computing and Information Science in Engineering. Manuscript received October 7, 2004; final manuscript received June 28, 2005. Guest Editor: K. Shimada.

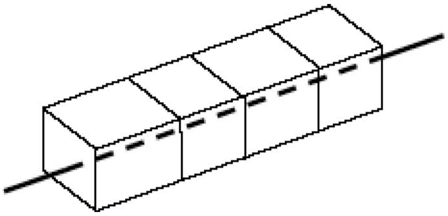

Fig. 1 A stack of hexahedrons represented by a dual chord (black line)

the elements through the void created by removing the hex sheet, the mesh has been coarsened. The original mesh had 32 increments around the top outer surface, the coarsened mesh has 30 increments.

Uniform removal of several hex sheets is demonstrated in Fig. 5. Here the upper edge has been specified for the coarsening procedure. Figure 5(*a*) has 26 increments along the upper edge, and Fig. 5(*b*) has been coarsened to 8 increments along this upper edge. Note however that removing sheets in this manner can affect regions remote from the area defined for coarsening [17].

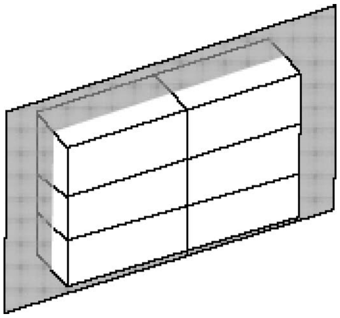

Fig. 2 A sheet of hexahedrons represented by a dual "twist plane"

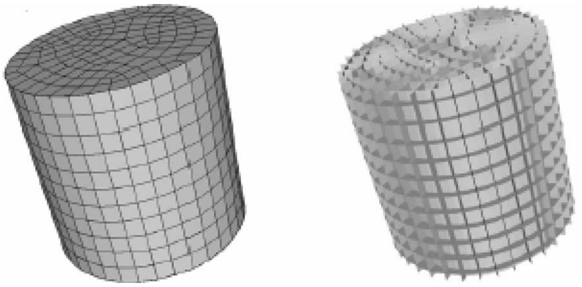

Fig. 3 A meshed cylinder and the corresponding dual

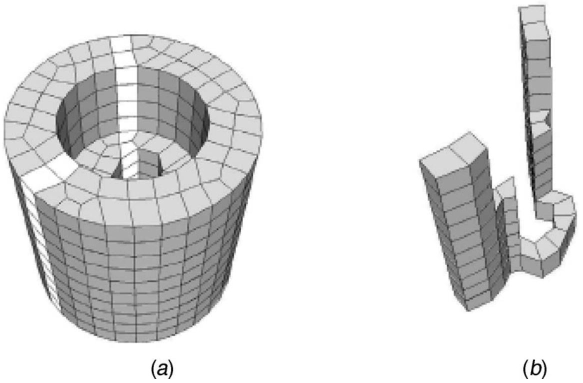

Fig. 4 An all-hex mesh (*a*) and single sheet (*b*)

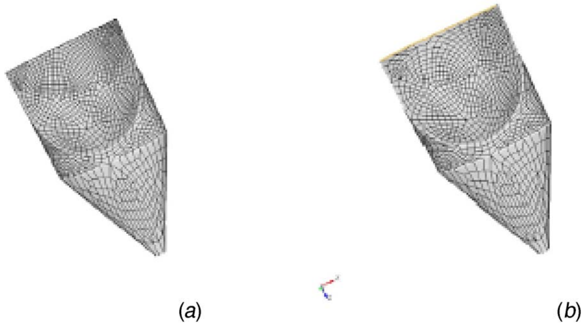

Fig. 5 Uniform removal of numerous sheets along a specified curve

3.2 Local Coarsening. The coarsening techniques described earlier coarsen a mesh through the removal of complete sheets. Removing entire sheets may introduce undesired characteristics in remote locations of the mesh and change mesh boundary conditions. For these reasons it is often desirable to locally coarsen the mesh. Local coarsening is achieved through the application of what we define as an initial grid structure followed by the insertion of transition templates to preserve the conformal characteristics of the mesh. Smoothing routines may then be applied as needed to optimize the quality of the mesh.

3.2.1 Application of the Grid Structure. The basic steps for the coarsening algorithm presented here is to first identify and then to modify hex triad configurations. A hex triad and its resulting coarsened state are shown in Fig. 6. Multiple hex triads are grouped into an appropriate local grid structure to effectively coarsen a specific region of the mesh. These grid structures represent the successful combination of three neighboring elements that all lie on the same chord into a coarsened element.

Using hex triads reduces the computational complexity of grid placement and also respects the rules of the dual. By grouping hex triads into groups of three, a triple triad can be created. The manner and order of placement of these triads to create a temporary grid structure is determined automatically by evaluation of the defined coarsening region. Algorithmically, this is accomplished through a number of weighting techniques that respect the density and geometric boundary conditions of the uncoarsened mesh. Also, only those configurations of grids that maintain the connectivity of the dual can be used. Section 3.2.3 explains how this grid structure has been placed to avoid the creation of any nonconformal areas in the mesh when transition templates are inserted.

3.2.2 Creation of Valid Coarsened Elements. The placement of the initial grid structure does not delete or create new mesh elements. A number of things must occur before the grid structure can be transformed into elements. The first step is to create two continuous edges (2D) out of the six edges that currently exist or four continuous edges (3D) out of the twelve existing edges on the underlying uncoarsened elements. The edges being discussed never encompass the edges that constitute the caps or ends of an element. Figure 6(*b*) shows a coarsened hexahedral (3D) element

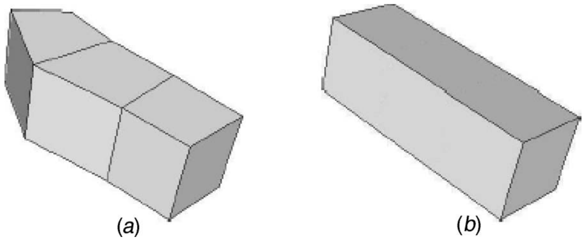

Fig. 6 A hex triad (*a*), coarsened element (*b*)

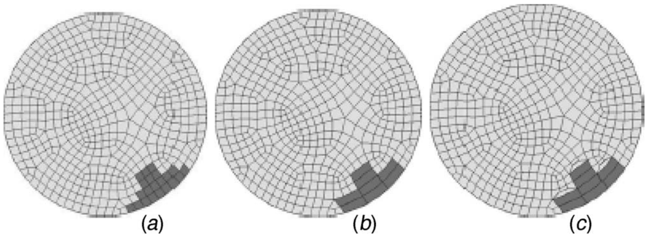

Fig. 7 User selected elements (*a*), applied grid structure (*b*), transition element insertion (*c*)

with the twelve original edges merged into four edges on the coarsened element. It is important to note that these edges in an unstructured mesh will often not create a straight line [see Fig. 6(*a*)]. Optimization techniques are applied to select the best placement of this new merged edge while preserving the quality of the mesh in that region. Next, a single new element must be created from the three current mesh elements. The last step is to delete the three former elements leaving just the new coarsened element.

3.2.3 Insertion of Transition Elements to Preserve Mesh Conformity. Figures 7(*a*) and 7(*b*) show how a two-dimensional grid structure is developed from a defined region to be coarsened. As is shown in Fig. 7(*b*), once new coarsened elements are placed according to the requirements of the grid structure, there will be nonconformities in the mesh. These nonconformities are resolved through the use of transition templates. Transition templates redirect a dual's chord direction in a mesh, thus preserving conformity. Figure 7(*c*) shows the mesh from Figs. 7(*a*) and 7(*b*) after transition elements have been inserted. Comparing Figs. 7(*a*)–7(*c*) show the progression from a defined coarsened region to a newly configured and conformal locally coarsened mesh.

3.2.4 Execution of Postprocessing, Mesh Clean-Up and Smoothing Routines. Because the mesh has been modified, it is important to verify that the final mesh quality is adequate for analysis. If the mesh quality is poor, it may be desirable to apply a smoothing algorithm to improve areas in the mesh that have low quality elements.

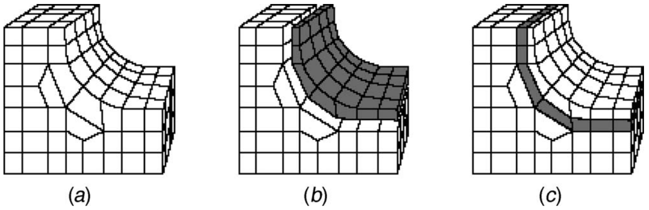

Fig. 8 Using a surface to define the shrink region

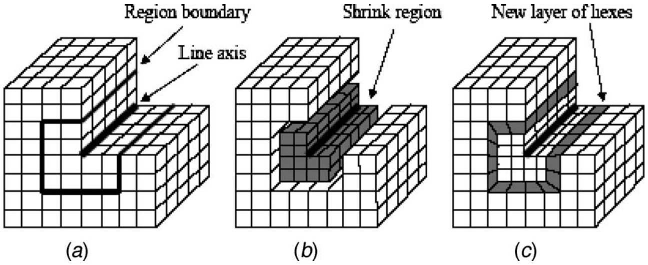

Fig. 10 Using a line to define a shrink region

4 Refinement

In this section we explain how hex meshes are refined by either adding sheets in gaps that are created in the original mesh, or by adding additional sheets in existing sheets [18,19].

4.1 Refinement by Placing New Sheets in Created Gaps.

4.1.1 Defining the Shrink Region With a Surface. Surfaces can be used as a reference for defining a shrink region. When using surfaces to define the shrink region the layer of new elements will be inserted into the mesh so that it runs parallel to the reference surfaces. This is illustrated for a single reference surface in Fig. 8. The original mesh is shown in Fig. 8(*a*) and the concave surface is used as the reference surface. In Fig. 8(*b*) the hexahedral elements that are adjacent to the concave surface have been shrunk away from the original mesh. After shrinking, a void is created that is ready to be filled with a layer of new elements. The highlighted elements in Fig. 8(*c*) show the final mesh after the new elements have been inserted.

Once a good boundary layer has been created, dicing [20] can then be used to create biased elements. For problems with boundary sensitive analyses, this could provide improved accuracy.

Figure 9 shows an example of inserting a sheet adjacent to a surface and then dicing the new elements to create boundary layer elements. An original mesh of an airfoil is shown in Fig. 9(*a*). Figure 9(*b*) shows the mesh after inserting a layer of elements around the airfoil and dicing the layer with a biased scheme and six intervals. A close-up of the leading edge of the airfoil is shown in Fig. 9(*c*).

4.1.2 Defining the Shrink Region With a Line. Another useful method for specifying the elements in the shrink region is to use a characteristic line of the model [21]. Here, with this line as the axis and an associated distance as the radius, a topological cylinder (or torus) defines the boundary of the shrink region. A convenient way to specify the line and radius is to use a geometric curve as the line and mesh element intervals as the radius.

Figure 10 illustrates the process of defining a shrink region with a line. Figure 10(*a*) shows a sketch of an unrefined mesh. The geometric curve that defines the axis and the mesh elements that define the shrink region boundary are highlighted. In this example, the number of hexahedral elements from the curve defines the radius to the boundary of the shrink region. Here, a distance of

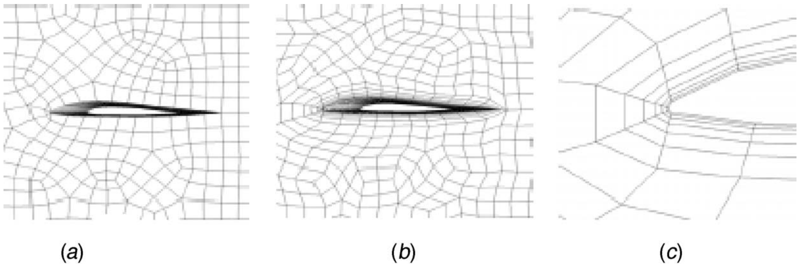

Fig. 9 Mesh of airfoil before and after refinement

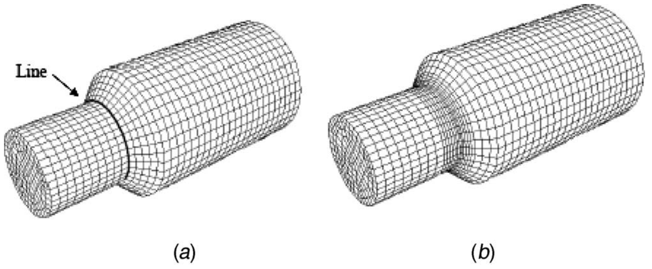

Fig. 11 Meshed shaft before and after refinement, the line shown defines the shrink region

two elements is used. Figure 10(*b*) shows the shrink region after it has been separated from the original mesh and Fig. 10(*c*) shows the new layer of elements inserted into the void.

An example of a mesh that has been refined using a line to define the shrink region is shown in Fig. 11. Figure 11(*a*) shows the original mesh of a shaft and the line that is used to define the shrink region. In this example, four layers of hexahedral elements are inserted into the void created by shrinking the elements in the original mesh. The final mesh is shown in Fig. 11(*b*). Figure 12 shows a cross section from before and after the refinement operation.

Table 1 gives a comparison of the quality of the meshes in Fig. 11. The metric used here is the "shape" quality, f_{shape}, as proposed by Knupp [22]. This metric has a value of 1.0 if the element is a perfect cube and has a value of 0.0 if the element is degenerate. The metric is mathematically defined as

$$f_{\text{shape}} = \frac{24}{\sum_{k=0}^{7} (\lambda_{11}^{k} + \lambda_{22}^{k} + \lambda_{33}^{k})/(\alpha_k^{2/3})}$$

where λ_{ij}^{k}=the ijth component of the kth metric tensor; α_k=the determinant of the kth Jacobian matrix.

The refinement process has increased the number of elements from 8925 to 9883 without causing a drastic change in the overall quality of the mesh. Although the quality of this example did not decrease much, it is important to note that in some cases adding elements to a mesh can cause a significant decrease in the mesh quality.

4.1.3 Defining the Shrink Region With a Point. A final method that will be discussed in this paper for specifying the elements in a shrink region when doing feature refinement is to use a point. In this case, the point and an associated radius are used to create a topological sphere. A point can be conveniently defined using a vertex or a node.

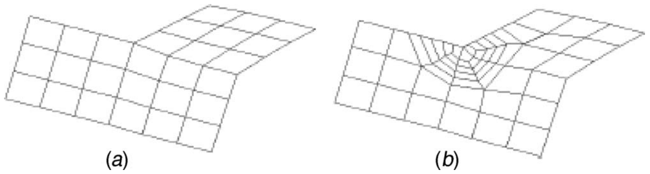

Fig. 12 Cross-section views of refinement to shaft mesh

Table 1 Shape metric quality of shaft mesh before and after refinement

Shape metric, f_{shape}	No. hexes	Avg.	Std. Dev.	Min.	Max.
Before	8925	0.7943	0.1059	0.4178	0.9675
After	9883	0.7611	0.1274	0.4266	0.9675

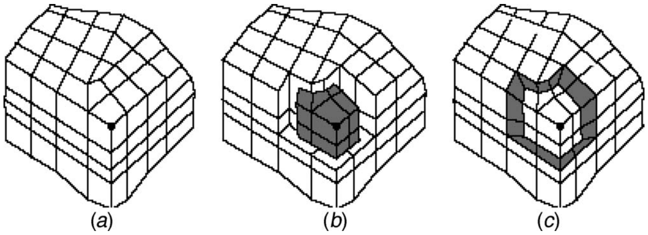

Fig. 13 Using a point to define the shrink region

Figure 13 illustrates the process of defining a shrink region with a point. Figure 13(*a*) shows a sketch of an unrefined mesh. The vertex that will be used to define the shrink region is highlighted. In this example, the number of hexahedral elements from the point defines the radius to the boundary of the shrink region. Here, a distance of two elements is used. Figure 13(*b*) shows the shrink region separated from the original mesh and Fig. 13(*c*) shows the new layer of elements inserted into the void.

4.2 Refinement by Placing Multiple New Sheets Within Existing Sheets. The following scheme was originally presented by Harris et al. [19] and improved here by providing an accommodation for concave corners (Sec. 4.2.2) and controlling the density of the refinement area (Sec. 4.5). The theory supporting single sheet operation refinement is based on modification of the spatial twist continuum. Each hex element is defined by the intersection of three twist planes. In two dimensions, these planes are reduced to chords shown by the dashed lines in Fig. 14(*b*).

To increase local mesh density, additional chords are inserted that intersect the original chords and either exit the mesh at a boundary or close back to create loops as shown by the dark dashed lines in Figs. 15(*a*) and 15(*c*). Each new intersection between the inserted chord and an original chord defines a new element, Figs. 15(*b*) and 15(*d*).

In two dimensions, two directions of refinement divide each target quadrilateral, shown in gray in Fig. 14(*a*), into nine new quadrilaterals. Each inserted chord loop represents a single direction of refinement. Note that each direction of refinement takes place within a single column or row of elements. Each column or row is defined by the chord that runs through the centers of all its elements.

The above refinement concept is directly expanded to three dimensions. Instead of intersecting chords, the elements within an all-hexahedral mesh are defined by the intersections of three twist planes. All elements intersected by a single twist plane compose a hex sheet. Each direction of refinement occurs within a single hex sheet where a completely enclosing twist plane spheroid, the 3D equivalent of a chord loop, is inserted. Figure 16(*a*) shows a single sheet wherein one direction of refinement has occurred.

Three directions of refinement divide the central target hex of Fig. 16(*b*) into 27 new elements. As also seen above, transition elements are created in the region where the twist plane is turned

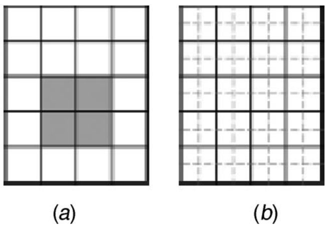

Fig. 14 2D mesh with target element selected (*a*) and the chords that define the mesh (*b*)

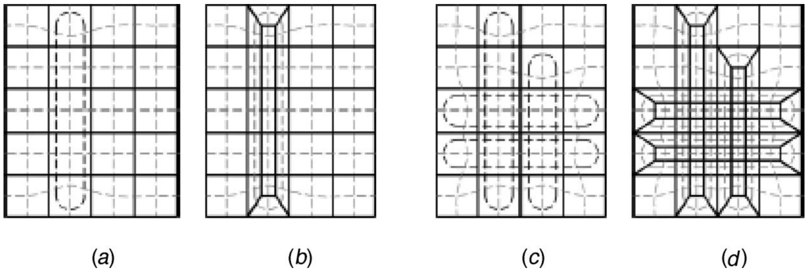

Fig. 15 A single chord-loop insertion (*a*) and resulting mesh (*b*); multiple chord-loop insertion (*c*) and resulting mesh (*d*)

back 180 deg. The transition elements surround the target areas, transitioning between coarse and fine mesh regions. Crossing multiple twist planes extends the refinement region.

Complete hex sheets are guaranteed features of conforming all-hexahedral meshes. Because each direction of refinement occurs completely within a single hex sheet, a conforming mesh after refinement is also guaranteed.

4.2.1 Templates. Only three templates are needed to perform each direction of refinement, a main template shown in Fig. 17(*a*) and two transition templates (*b* and *c*). Template (*a*) is used to divide the target hex first into three hexes in one direction, then the three into nine in the second direction and finally the nine into 27 in the third direction. Template (*b*) borders a face of the target element and serves to reverse the path of the inserted twist plane back through the target hex. Template (*c*) reverses the twist plane through an edge of the target hex.

The templates are chosen based on the number of selected nodes on an element in a sheet that is to be refined. These selected nodes are, for each template, marked with black dots in Fig. 17. All nodes of a target element are always selected, thus template (*a*) requires eight marked nodes. Four selected nodes on a single element face define template (*b*). The orientation of templates (*a*) and (*b*) will be correct if the divided edges between the selected nodes lie entirely within the hex sheet. Two selected nodes define the corner template (*c*).

4.2.2 Concave Corner Accommodation. When concave corners arise, another template is needed. An example of this situation is shown in Fig. 18. Figure 18(*a*) shows hexes selected for refinement. In Fig. 18(*b*), one twist plane has been inserted in the extracted sheet. A template that could fill the concave region requires the surface characteristics of the mock template in Fig. 18(*c*). However, such a configuration forces the inserted twist plane to self-intersect within the template. Such a template cannot be constructed with reasonable quality. However, to accommodate this situation, an adjustment to the mock template can be made. This adjustment is shown in Fig. 19. Figure 19(*a*) shows two shaded surfaces on the mock template and two surfaces on adjoining elements that will be merged to form a common surface. Figure 19(*b*) shows the element configuration after the shaded surfaces have been merged. This element configuration in the corner of the concavity now accommodates the refinement of the region.

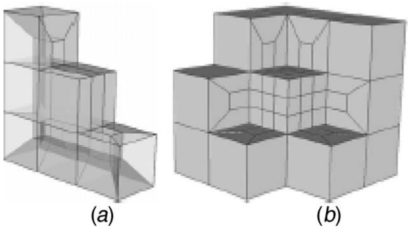

Fig. 16 An extracted sheet showing a single direction of refinement on the target center hex (*a*) and the mesh after three directions of refinement (*b*)

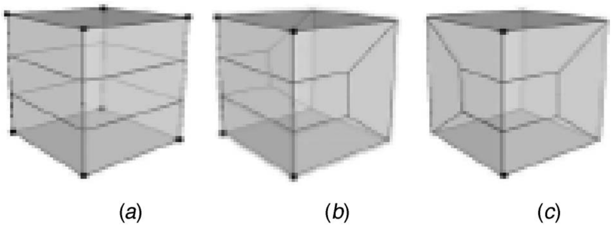

Fig. 17 Single hex sheet refinement templates

4.2.3 Algorithm Description and Examples. Due to the above restrictions, refinement using single sheet operations is most suited for target regions requiring the division of single elements and groups of elements. The actual algorithm marks the nodes of all target hexes then loops through the hex sheets, accommodating concavities and inserting the templates. With the concavities accommodated, each hex in the sheet will only have eight, four, two or zero marked nodes.

The mesh in Fig. 20(*a*) is composed of all-hex elements and is conforming. Figure 20(*b*) shows the refinement of a group of selected hexes. Figure 20(*c*) illustrates refinement of a selected curve (lower left). Refinement of a node, element edge or element

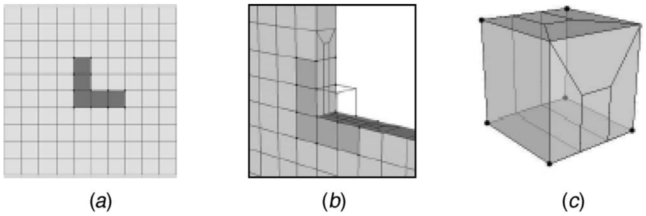

Fig. 18 Concavity issues

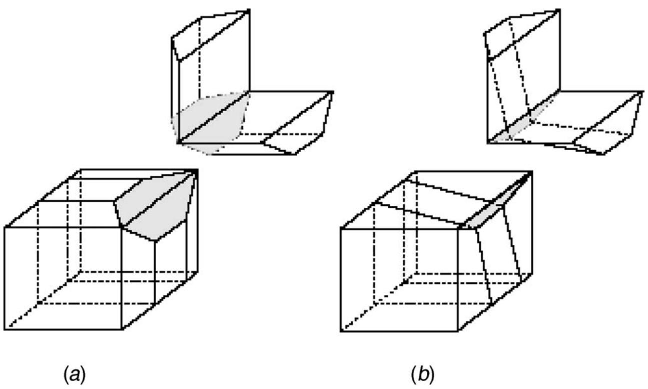

Fig. 19 Concavity adjustment

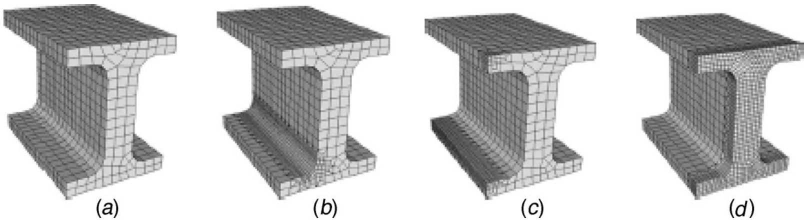

Fig. 20 Single sheet operations: Original mesh (*a*), selected elements (*b*), selected curve and vertex (*c*), selected surface (*d*)

face using this method would all appear as shown in the upper left refinement of Fig. 20(*c*). Figure 20(*d*) shows the refinement of a single surface.

4.3 Near Node and Edge Refinement. A modification of the above procedure to allow refinement directly around element nodes, edges, and faces, is to introduce the templates shown in Fig. 21.

This technique is applied by refining the selected nodes, element edges, and element faces with the templates of Fig. 21. For individual nodes, only this first step is needed. Figure 22(*a*) shows the additional sheet refinement of two edges and the selected nodes (in black) that defined the refinement region. Only additional templates (*b*) and (*c*) were used to perform the refinement. Figure 22(*c*) shows additional sheet refinement of a single selected face with corresponding selected nodes. All three additional sheet templates were used in this step.

Following a single direction of additional sheet refinement, all eight nodes on the elements adjacent to the target edges or faces are flagged as depicted by the black nodes in Figs. 22(*b*) and 22(*d*). Single sheet refinement then follows as described previously with hex sheets defined by the target edges, shown with the heavy black line in Fig. 22(*b*), or two adjacent edges of a target face, shown in Fig. 22(*d*). Only one direction of single sheet refinement is necessary for every target edge while two directions are necessary for each target face. After refinement is completed, a single layer of quality elements exists between the selected entity and the transition elements.

4.3.1 Examples. Figure 23(*a*) shows the refinement of a single selected node. Each element surrounding the target node (marked with black) is replaced with template (*c*) from Fig. 21. Figure 23(*b*) shows the mesh after applying a "mean ratio" smoothing operator [23]. Care should always be given before smoothing after any refinement in that the quality of some elements may decrease as the quality of others increase. In addition, smoothing may expand the size of some elements in a region where the size should remain small. However, smoothing can be used to minimize the size transition between elements as shown in Fig. 23.

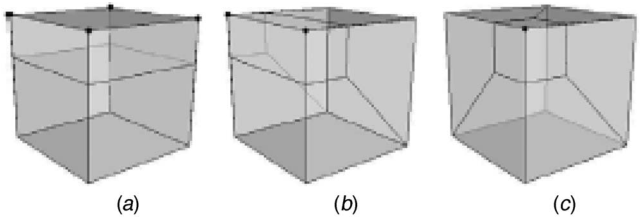

Fig. 21 Additional hex sheet templates

Figure 24(*a*) is an all-hex conforming mesh. Figure 24(*b*) shows combination refinement of several edges after smoothing. Figure 24(*c*) shows combination refinement of a group of faces after smoothing.

4.4 Quality Issues. The quality of a hexagonal mesh is usually degraded subsequent to refinement. Again, Knupp's shape metric, f_{shape}, can be used to evaluate quality of hexahedral meshes prior to and after refinement. Recall that this metric has a value of 1.0 if the element is a cube and has a value of 0.0 if the element is degenerate. The meshes of Fig. 25, which are summarized in Table 2, show the typical quality degradation associated with refinement. This example demonstrates that the new combination technique, as presented above, restricts the refinement to a smaller region and causes less degradation in both overall and minimum quality. These features of the proposed combination technique thus provide some advantages over previous methods.

4.5 Refinement by Placing New Sheets Selectively Within Existing Sheets. Some of the significant limitations of the schemes explained in Secs. 4.2 and 4.3 are that they often overrefine a target area or do not allow a smooth transition from the refined to the unrefined area. A method to accommodate such needs can be accomplished by slight modification of the method presented in Sec. 4.2. The modification we suggest uses only a one directional refinement we call a "stint" as shown in Fig. 26. Note that a stint could be considered a planar template defined to reside in an existing sheet. The stint is completely defined by giving the number of elements it encompasses in the two planar directions of the sheet in which it resides. The stint in Fig. 26 is of dimension 3×3.

Figure 27 shows an example of controlled stint placement. Figure 27(*a*) shows the original unrefined mesh. We desire to refine the front face of the object by increasing from a 10×10 increment to a 22×22 increment. To accomplish this, we place stints in the object that penetrate 3, 4, and 7 elements deep. The locations of

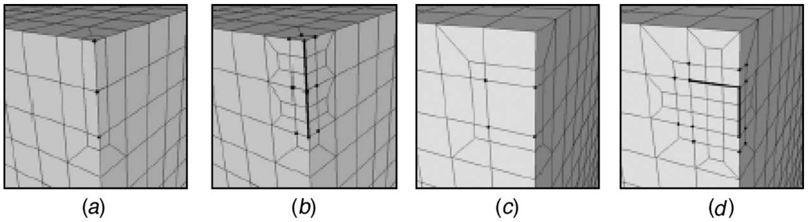

Fig. 22 Combination refinement: parallel sheet refinement (*a* and *c*) followed by multiple directions of single sheet refinement (*b* and *d*)

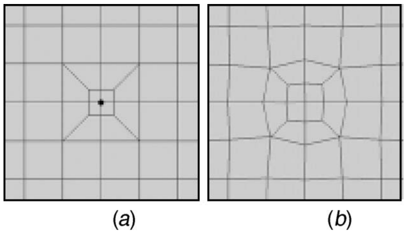

Fig. 23 Single node refinement: refined mesh (*a*), smoothed mesh (*b*)

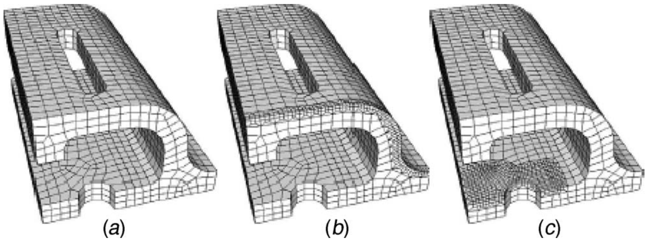

Fig. 24 Combination refinement: Original mesh (*a*), selected edges (*b*), and selected faces (*c*)

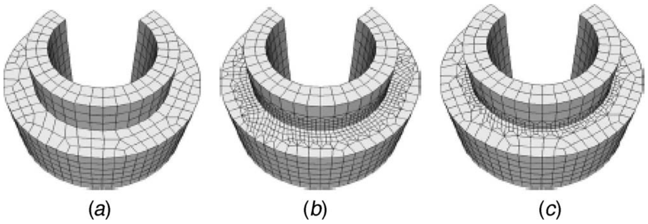

Fig. 25 Quality comparison (*a*) original mesh, (*b*) single sheet refinement, (*c*) combination refinement

Table 2 Comparison of f_{shape} shape quality, of the meshes shown in Fig. 25

Refinement scheme	Figure	No. hexes	Avg.	Std. Dev.	Min.	Max.
Original mesh	25(*a*)	1116	0.84	0.07	0.61	0.93
Single sheet	25(*b*)	5828	0.75	0.2	0.26	1
Combination	25(*c*)	1962	0.78	0.11	0.45	0.95

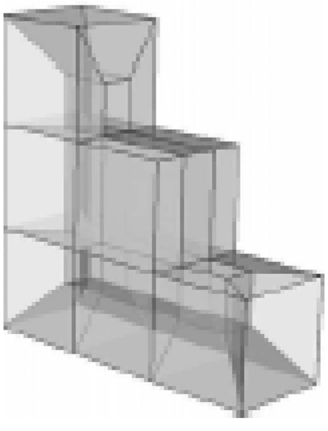

Fig. 26 A single direction of refinement, i.e., a "stint"

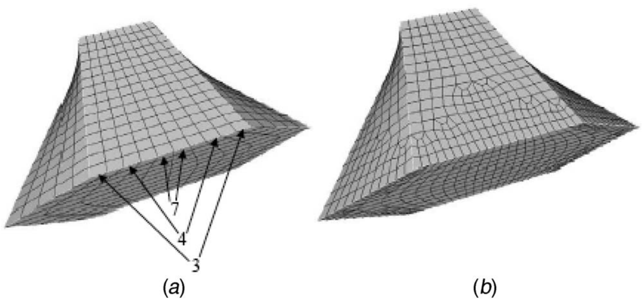

Fig. 27 Variable single sheet refinement example

some of these stints are shown in the figure. Note that the achieved refinement is neither too fine and that it staggered as the refinement goes from the fine to coarser regions.

5 Conclusions

In this paper we have presented both refinement and coarsening schemes for conformal all-hexahedral meshes. All the schemes rely upon the dual of the mesh for directing the final mesh modification. Coarsening algorithms are currently in their infancy and much work is still needed to develop robust schemes. Current conformal refinement schemes provide many options for adding elements to a specified region, but often these schemes lead to over-refinement in a particular region. The variable single sheet refinement presented here shows promise for allowing both smoother and coarser refinement of conformal unstructured all-hexagonal meshes. Sophisticated all-hexahedral meshing schemes should combine the best features of both coarsening and refining to create quality adapted conformal hexahedral meshes.

Acknowledgments

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the U.S. Department of Energy's National Nuclear Security Administration under Contract No. DE-AC04-94AL85000.

References

[1] Carey, G., 1997, *Computational Grids: Generation, Adaptation, and Solution Strategies*, Taylor & Francis, New York.
[2] Plaza, A., and Cary, G., 1996, "About Local Refinement of Tetrahedral Grids," in *5th International Meshing Roundtables*, Pittsburgh, PA, Sandia National Laboratories, pp. 123–136..
[3] Plaza, A., and Rivara, M.-C., 2003, "Mesh Refinement Based on the 8-Tetrahedra Longest-Edge Partition," in *12th International Meshing Roundtable*, Santa Fe, NM, Sandia National Laboratories, pp. 67–78.
[4] De Cougny, H. L., and Shephard, M. S., 1999, "Parallel Refinement and Coarsening of Tetrahedral Meshes," Int. J. Numer. Methods Eng., **46**, pp. 1101–1125.
[5] Bey, J., 1995, "Tetrahedral Grid Refinement," Computing, **55**, pp. 355–378.
[6] Grosso, R., Lurig, C., and Ertl, T., 1997, "The Multilevel Finite Element Method for Adaptive Mesh Optimization and Visualization of Volume Data," in *Eighth IEEE Visualization* 1997 (VIS '97), Phoenix, AZ, pp. 387–394.
[7] Molino, N., et al., 2003, "A Crystalline, Red Green Strategy for Meshing Highly Deformable Objects With Tetrahedra," in *12th International Meshing Roundtable*, Santa Fe, NM, Sandia National Laboratories, pp. 103–114.
[8] Tchon, K.-F., Hirsch, C., and Schneiders, R., 1997, "Octree-Based Hexahedral Mesh Generator for Viscous Flow Simulations," in *13th AIAA Computational Fluid Dynamics Conference*, No. AIAA-97-1980, Snowmass, CO.
[9] Marechal, L., 2001, "A New Approach to Octree-Based Hexahedral Meshing," in *10th International Meshing Roundtable*, Newport Beach, CA, Sandia National Laboratories, pp. 209–221.
[10] Schneiders, R., 2000, "Octree-Based Hexahedral Mesh Generation," Int. J. Comput. Geom. Appl., **10**, No. 4, pp. 383–398.
[11] Kwak, D-Y., and Im, Y-T., 2002, "Remeshing for Metal Forming Simulations—Part II: Three Dimensional Hexahedral Mesh Generation," Int. J. Numer. Methods Eng., **53**, pp. 2501–2528.
[12] Li, Hua, and Cheng, Gengdong, 2000, "New Method for Graded Mesh Generation of All Hexahedral Finite Elements," Comput. Struct., **76**, pp. 729–740.
[13] Tchon, K.-F., Dompierre, J., and Camarero, R., 2002, "Conformal Refinement of All-Quadrilateral and All-Hexahedral Meshes According to an Anisotropic Metric," in *11th International Meshing Roundtable*, Ithaca, New York, Sandia

National Laboratories, pp. 231–242.

[14] Harris, N. J., “Conformal Refinement of All-Hexahedral Finite Element Meshes,” M.S. thesis, Brigham Young University, Provo, UT, 2004.

[15] Tautges, T., Blacker, T., and Mitchell, S., 1996, “The Whisker Weaving Algorithm: A Connectivity-Based Method for All-Hexahedral Finite Element Meshes,” Int. J. Numer. Methods Eng., **39**, pp. 3327–3349.

[16] Murdoch, P., Benzley, S., Blacker, T., and Mitchell, S. A., 1997, “The Spatial Twist Continuum: A Connectivity Based Method for Representing All-Hexahedral Finite Element Meshes,” Finite Elem. Anal. Design, **28**, pp. 137–149.

[17] CUBIT, Version 9.1 Users Guide, Sandia National Laboratories, (2004) URL: http://cubit.sandia.gov/help-version9.1/cubithelp.html.

[18] Borden, M., Benzley, S. E., Mitchell, S. A., White, D. R., and Meyers, R., 2000, “The Cleave and Fill Tool: An All-Hexahedral Refinement Algorithm for Swept Meshes,” *Proceedings 9th International Meshing Roundtable*, SNL, New Orleans, LA, pp. 69–76.

[19] Harrris, N. J., Benzley, S. E., and Owen, S. J., 2004, “Conformal Refinement of All-Hexahedral Element Meshes Based on Multiple Twist Plane Insertion,” in *Proceedings, 13th International Meshing Roundtable*, SNL, Williamsburg, VA, pp. 157–167.

[20] Melander, D. J., Tautges, T. J., and Benzley, S. E., 1997, “Generation of Multi-Million Element Meshes for Solid Model-Based Geometries: The Dicer Algorithm,” *AMD-Vol. 220 Trends in Unstructured Mesh Generation*, ASME, pp. 131–135.

[21] Benzley, S. E., Borden, M., and Mitchell, S. A., 2001, “A Refinement Technique for All-Hexahedral Adaptive Meshing,” *Computational Fluid and Solid Mechanics, Proceedings, First MIT Conference on Computational Fluid and Solid Mechanics*, Elsevier Science, pp. 1519–1523.

[22] Knupp, P., 2003, “Algebraic Mesh Quality Metrics for Unstructured Initial Meshes,” Finite Elem. Anal. Design, **39**, pp. 217–241.

[23] Brewer, M., Diachin, L., Knupp, P., Leurent, T., and Melander, D., “The Mesquite Mesh Quality Improvement Toolkit,” in *Proceedings of the 12th International Meshing Roundtable*, SNL, Albuquerque, NM, pp. 239–250.