CONFORMAL REFINEMENT OF ALL-HEXAHEDRAL ELEMENT MESHES BASED ON MULTIPLE TWIST PLANE INSERTION

Nathan J. Harris[1] Steven E. Benzley[1] Steven J. Owen[2]

[1]Brigham Young University, Provo, UT, U.S.A. {njh33|seb}@et.byu.edu
[2]Sandia National Laboratories*, Albuquerque, NM, U.S.A. sjowen@sandia.gov

ABSTRACT

This paper presents an automated tool for local, conformal refinement of all-hexahedral meshes based on the insertion of multi-directional twist planes into the spatial twist continuum. The refinement process is divided into independent refinement steps. In each step, an inserted twist plane modifies a single sheet or two parallel hex sheets. Six basic templates, chosen and oriented based on the number of nodes selected for refinement, replace original mesh elements. The contributions of this work are (1) the localized refinement of mesh regions defined by individual or groups of nodes, element edges, element faces or whole elements within an all-hexahedral mesh, (2) the simplification of template-based refinement into a general method and (3) the use of hex sheets for the management of template insertion in multi-directional refinement.

Keywords: adaptation, conformal refinement, hexahedral, unstructured mesh

1. INTRODUCTION

Finite element analysis is an essential tool for scientists and engineers. Before analysis can begin, a mesh of the model is created. Much attention has been given to automatic mesh generation. Two-dimensional models frequently use triangle and/or quadrilateral elements while three-dimensional models generally employ tetrahedral and/or hexahedral elements. Mesh adaptation is used to improve the accuracy of the analysis by modifying the mesh to reflect the physics of the problem. Over the years, attention has been given to 2D and all-tetrahedral mesh refinement and coarsening, resulting in techniques that effectively increase or decrease mesh density in localized regions.

In many cases, hexahedral meshes provide advantages over tetrahedral meshes. Several schemes have been developed for the refinement of such meshes. Methods using iterative octrees [1] result in non-conformal elements that cannot be handled by some solvers. Refinement techniques based on pillowing, such as the cleave-and-fill tool [2], allow local conformal refinement, however, the control and scale of refinement is limited. Other techniques insert non-hex elements that result in hybrid meshes or require uniform dicing to maintain a consistent element type [3]. Schneiders' directional refinement method [4] produces a conformal mesh by pillowing layers in alternating *i*, *j* and *k* directions but requires a Cartesian initial octree. The 3D anisotropic refinement scheme presented by Tchon et al. [5] 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.

This paper presents an adaptation to Tchon's method coupled with a technique for refinement of mesh regions around individual nodes, element edges and element faces. The contributions of this work are (1) the localized refinement of mesh regions defined by individual or groups of nodes, element edges, element faces or whole elements within an all-hexahedral mesh, (2) the simplification of template-based refinement into a general method and (3) the use of hex sheets for the management of template insertion in multi-directional refinement.

The refinement algorithm has been implemented in CUBIT, a meshing tool kit written and maintained by Sandia National Laboratories [6].

*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

2. REQUIREMENTS

The refinement technique presented was developed to meet several requirements necessary for inclusion within the CUBIT framework's interactive user interface. It was also developed to be used within a mesh adaptivity module intended to be invoked during a computational analysis. Both the original and refined meshes are all-hexahedral, unstructured and conforming. The technique is capable of refinement based on individual mesh entities (nodes, edges, faces and hexes) and on geometric entities (vertices, curves, surfaces and volumes). For use within an interactive user environment, it is expected that the user would indicate target refinement parameters directly on the geometric entities of the meshed CAD model. Using the geometry-to-mesh associativity provided by the CUBIT framework, the appropriate refinement procedures are performed on the associated mesh. An equally important requirement is the ability to specify refinement directly on mesh entities. Although useful within an interactive setting, the ability to flag certain mesh entities (usually nodes or elements) for refinement is important within an environment where refinement is adaptively driven by analysis results.

To be consistent with existing triangle, quadrilateral and tetrahedral refinement operations within the CUBIT environment, the ability to perform refinement based on one or more of the following parameters was also required.

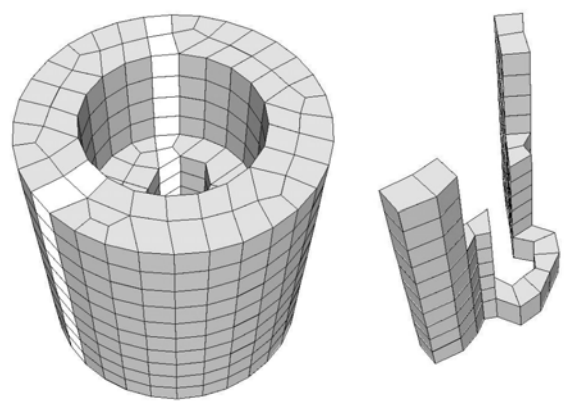
1. Subdivision Count: The number of times to subdivide each element.
2. Size: The target element size.
3. Bias: The maximum change in element size from the specified size to the pre-existing background size.
4. Sizing Function: The specification of a function size $=f(x,y,z)$ that the refinement will attempt to match.
5. Depth: The number of elements distant from the specified refinement region that will be affected by the refinement operation.
6. Radius: The radial distance from the specified refinement region that will be affected by the refinement operation.
7. Smoothing: The ability to either inhibit or promote smoothing to improve element quality after a refinement operation.

To provide the greatest flexibility for both geometric and mesh based refinement control, a node-based refinement algorithm was selected. This was selected because any refinement operation specified in terms of a higher order entity can always be cast in terms of a set of its associated nodes. This also provided consistency with the existing node-based algorithms used for triangle, quadrilateral and tetrahedral refinement within the existing framework.

3. OVERVIEW

To initiate refinement, a target region is specified. The target region is the portion of an existing mesh where the density is to be increased. The region is selected by size criteria based on adaptive analysis. The target is specified by selecting one or more geometry or mesh entities. This may be done either interactively, or through an application programmer's interface (API). The refinement region is then reduced to the nodes that are associated with the selected entities. The algorithm discussed here flags these nodes then assigns templates to replace each element in the target region based on the number of flagged nodes on each element.

The algorithm is divided into two distinguishing procedures; single and parallel hex sheet operations. Hex sheets are features of conforming all-hex meshes and are defined as the elements on a single twist plane of the spatial twist continuum (STC) [7]. A single hex sheet is highlighted in the mesh of Figure 1(a) and displayed in Figure 1(b). Single sheet refinement operations refer to the execution of one direction of refinement within a single hex sheet while parallel sheet operations take place within two neighboring sheets.



(a) (b)

Figure 1: An all-hex mesh (a) and single sheet (b).

Single sheet refinement operations split the edges of target elements into three thereby dividing a single target hex into twenty-seven hexes. Parallel sheet refinement divides each edge into two, splitting a single element into eight hexes.

As summarized in Table 1, both single sheet and parallel sheet techniques are limited as to the mesh entities each can refine. By combining the two techniques into a hybrid method and coupling it with the single sheet technique, all mesh entities within a mesh are then capable of refinement.

Table 1: Comparison of refinement techniques for unstructured meshes. An "X" indicates that the method is capable of refining the mesh entity.

	Hex	Face	Edge	Node
Single Sheet	**X**			
Parallel Sheet			**X**	**X**
Combination		**X**	**X**	**X**

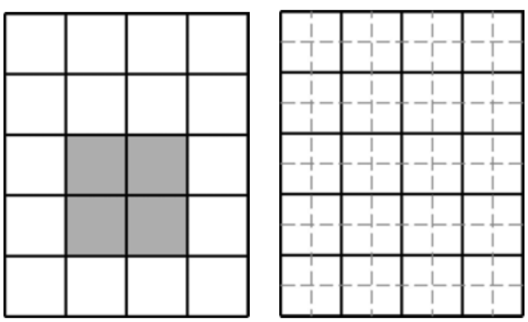
The steps of the complete algorithm are outlined as follows:

1. Select target region.
2. Check for sub-element targets.
   If yes, go to step 3.
   If no, go to step 5.
3. Refine around single nodes by inserting appropriate parallel sheet templates.
4. Insert parallel sheet refinement templates around target edges and faces.
5. Locate sheets that contain elements in target region.
6. Loop through sheets:
   Remove concavities in sheet.
   Insert single sheet refinement templates.

4. SINGLE HEX SHEET OPERATIONS

4.1 Theory

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 Figure 2(b).

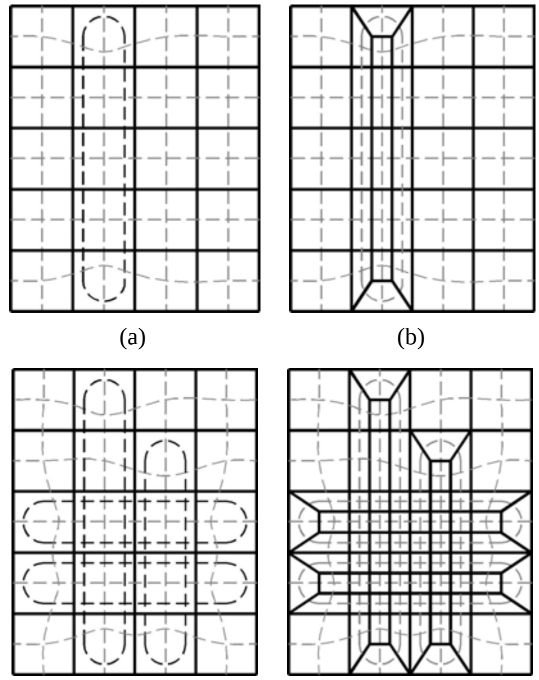


(a) (b)

Figure 2: A 2D mesh with target element selected (a) and the chords that define the mesh (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 Figure 3(a) and (c). Each new intersection between the inserted chord and an original chord defines a new element, Figure 3(b) and (d).

In two dimensions, two directions of refinement divide each target quadrilateral, shown in gray in Figure 2(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.

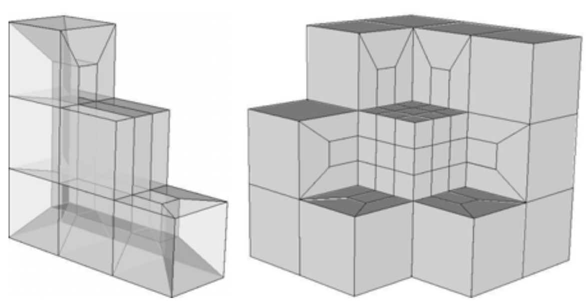


(a) (b)

(c) (d)

Figure 3: A single chord-loop insertion (a) and resulting mesh (b) and multiple chord-loop insertion (c) and resulting mesh (d).

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 4(a) shows a single sheet wherein one direction of refinement has occurred.



(a) (b)

Figure 4: An extracted sheet showing a single direction of refinement on the target center hex (a) and the mesh after three directions of refinement (b).

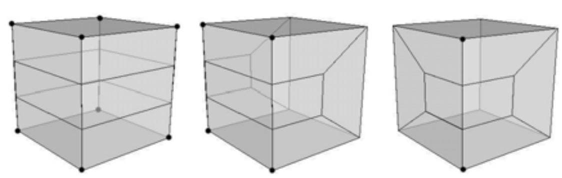
Three directions of refinement divide the central target hex of Figure 4(b) into 27 new elements. As also seen above, transition elements are created in the region where the twist plane is turned back 180 degrees. The transition elements surround the target areas, transitioning between coarse and fine 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 Templates

Only three templates are needed to perform each direction of refinement, a main template shown in Figure 5(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 twenty-seven 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 Figure 5. 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).

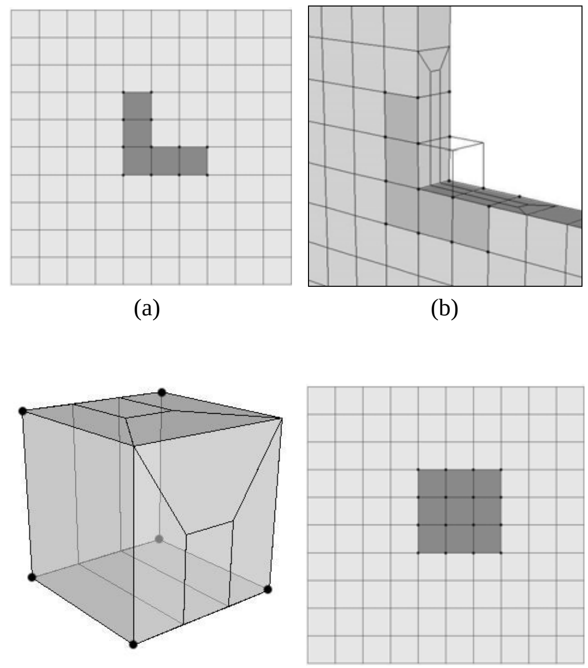


(a) (b) (c)

Figure 5: Single hex sheet refinement templates.

4.3 Restrictions

A single hex sheet refinement target area is restricted to a convex shape. An example of this restriction is shown in Figure 6. Figure 6(a) shows the hexes selected for refinement. In Figure 6(b), one twist plane has been inserted in the extracted sheet. The template necessary to fill the concave region indicated requires the surface characteristics of the mock template in Figure 6(c). Such a configuration forces the inserted twist plane to self-intersect within the template. Such a template cannot be constructed with reasonable quality. To circumvent this problem, the elements in the concave region are added to the target region, Figure 6(d).



(a) (b) (c) (d)

Figure 6: Concavity restrictions of selection region (a).

This refinement technique is further restricted by the inability to effectively refine the mesh around single nodes, element edges or element faces. For very localized refinement around a single node, the target area would be extended to all hexes attached to the node to maintain at a least one layer of good quality hexes, i.e., no transition hexes, immediately surrounding the node. However, extending the region may not be desirable. Template (c) could be used to refine elements around a selected edge, but would result in lower quality elements in the target refinement region.

4.4 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, removing concavities and inserting the templates. With the concavities removed, each hex in the sheet will only have eight, four, two or zero marked nodes.

The mesh in figure 7(a) is composed of all hex elements and is conforming. Figure 7(b) shows the refinement of a group of selected hexes. Figure 7(c) illustrates refinement of a selected curve (lower left). Refinement of a node, element edge or element face using this method would all appear as shown in the upper left refinement of 7(c). Figure 7(d) shows the refinement of a single surface.

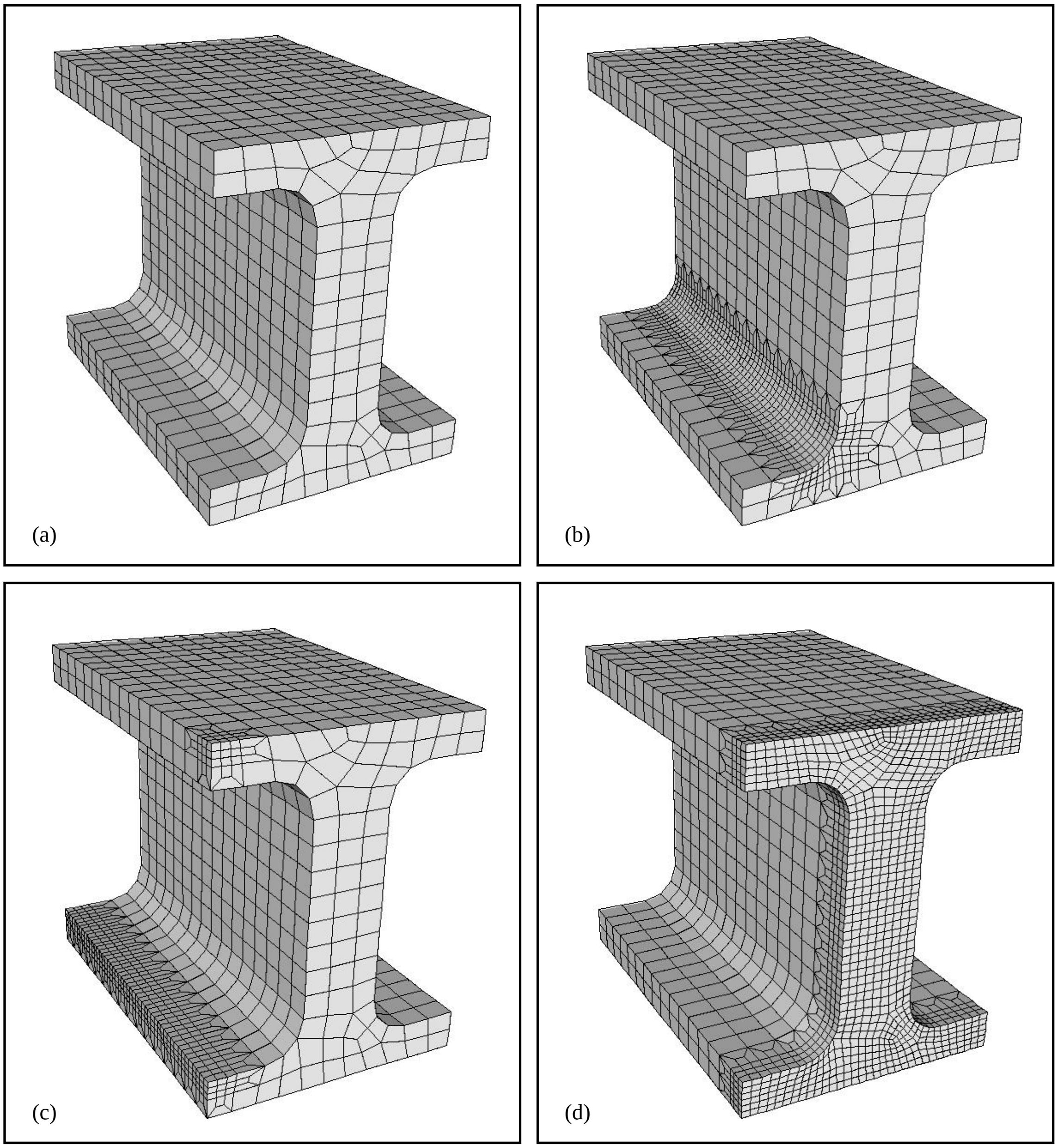


Figure 7: Single sheet operations: Original mesh (a), selected elements (b), selected curve and vertex (c), selected surface (d).

5. PARALLEL HEX SHEET OPERATIONS

5.1 Theory

Like single hex sheet operations, parallel hex sheet refinement operations are also based on the insertion of twist planes to increase local mesh density. Figure 8 shows examples of parallel sheet refinement reduced to two dimensions. In Figure 8(a), a single chord loop (black dashed line) is inserted that circumscribes the nodes of the target elements shared between two parallel columns of elements representing sheets. In the corresponding mesh, Figure 8(b), two sheets of elements are modified in this single refinement direction.

Figures 8(c) and (d) show that inserting new chord loops in two directions divide the target elements into four. Because the inserted chord is expanded into two neighboring sheets, the refinement is less dense than that seen in single sheet operations.

In three dimensions, a twist plane, the dashed line in Figure 9(b), is inserted that encloses the plane of nodes (marked in black) shared between two parallel sheets. Three directions of refinement divide original hexes into eight new hexes.

The scope of the refinement is adjusted through inserting multiple intersecting sheets.

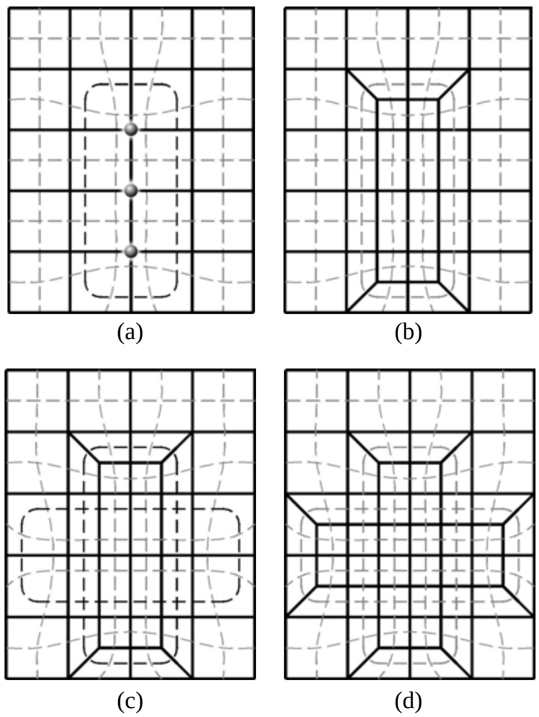


Figure 8: Single chord insertion (a) and resulting mesh (b) and multi-chord insertion (c) and resulting mesh (d).

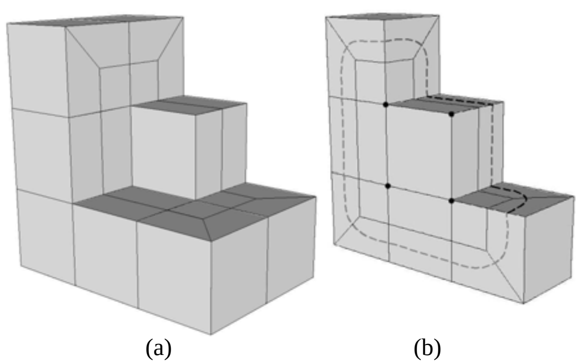


Figure 9: A single direction of lower order refinement (a) twist plane and selected nodes (b).

5.2 Templates

The refinement process is divided into multiple refinement directions. In each direction of refinement, a single twist plane loop is inserted into the mesh. At each step, the refinement zone can be described by a set of selected nodes shared by two neighboring hex sheets. Like single hex sheet refinement, parallel hex sheet refinement can be accomplished with three templates, a main template and two transition templates. The main template, Figure 10(a), is defined by four selected nodes on a face. Dividing a target hex with the main template in three directions carves the element into eight new elements. Two selected nodes on an element edge define template (b). Two (b) templates mirrored on two parallel sheets receive the new twist plane from one sheet and turn it completely into the neighboring sheet. Template (c) is defined by a single selected node and is used to reverse the twist plane direction at corners of the refinement region seen in Figure 9(b).

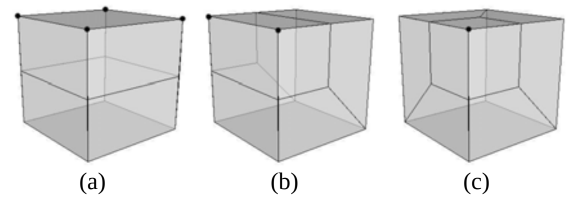


Figure 10: Parallel hex sheet refinement templates.

5.3 Restrictions

As with single hex sheet refinement, no simple template can be constructed to effectively accommodate concavity, Figure 11(a), into the inserted twist plane. Such a template would have the surface characteristics shown in figure 11(b). This template attempts to merge the two twist planes that enter the bottom face of the original element as illustrated in figure 11(c). Modifying the twist planes in such a manner results in unacceptable element quality within the template.

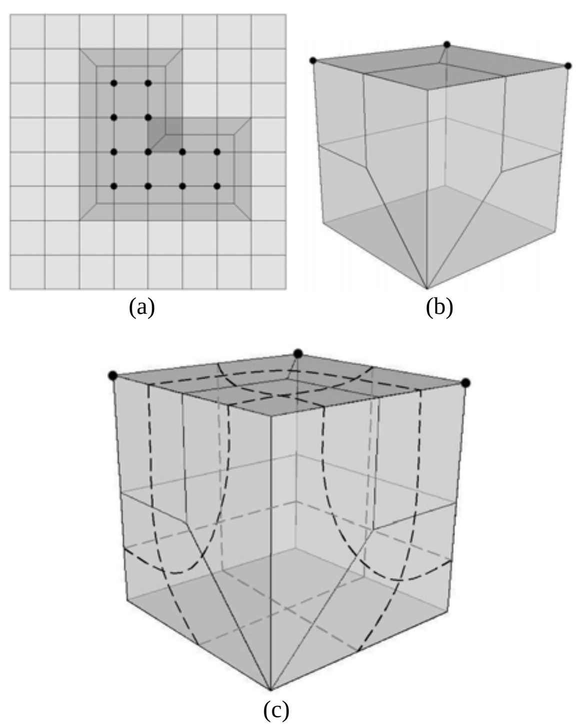
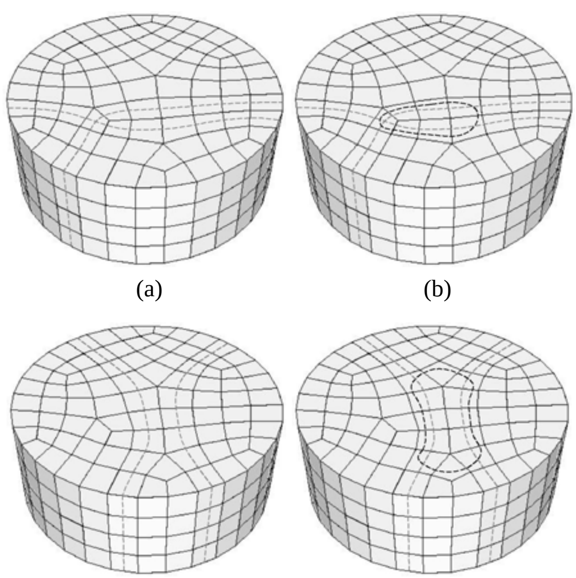


Figure 11: Concavity restriction for lower order refinement.

Parallel hex sheet operations can be performed in an area as localized as a single node and up to 1-into-8 refinement of an entire mesh provided the hex sheets within the refinement region remain parallel. This restriction makes parallel refinement of large portions of meshes impractical in highly unstructured meshes.

In unstructured meshes, neighboring hex sheets, defined by the dashed lines in Figure 12, can intersect (a) or diverge (c) within the mesh. In the case of intersecting sheets, the inserted twist plane must be reversed within the intersection elements. Continuing the inserted twist plane into another sheet near the intersection location would require the use of the template shown in Figure 11(c). For diverging sheets, the three templates given are capable of closing the twist plane at any location between the separated sheets. However, reversing the twist plane at the point of diversion confines the transition region to the fewest number of elements.

Sheet intersections and diversions within the mesh may cause the inserted twist planes to be reversed prematurely thereby introducing transition templates into the target refinement regions. Three directions of refinement would then no longer be possible for all target elements. Furthermore, multiple refinement directions would further increase transition template overlapping, significantly reducing mesh quality.



(a) (b)

(c) (d)

Figure 12: Intersecting and diverging hex sheets (a, c) and twist plane insertion restrictions (b, d).

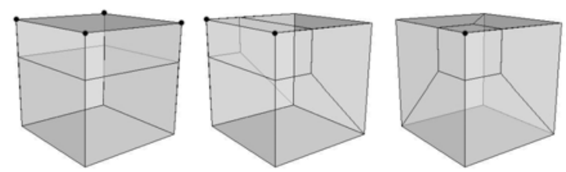
Because of the above restrictions, parallel sheet refinement is only suited for mesh regions where parallel hex sheets occur. Template (c) of Figure 10 is used to refine the elements surrounding a single node. In such situations, hex sheet orientation has no influence on refinement pattern.

Refinement of element edges using this technique is possible where the modified elements will replace only one or two layers of hexes. In such confined areas, intersecting or diverging sheets do not pose a problem. An example of refinement of element edges using parallel sheet refinement is shown in Figure 17(c). Refinement of element faces is only done by refining the entire element, which this technique is not well suited to do.

5.4 Combination

A more robust alternative to refinement around element edges, that also provides refinement of element faces, is to modify the parallel sheet refinement templates and use them in conjunction with the single sheet templates.

The templates in Figure 10 are modified by splitting the original element edges at one-third the edge length from the marked node to form those shown in Figure 13.

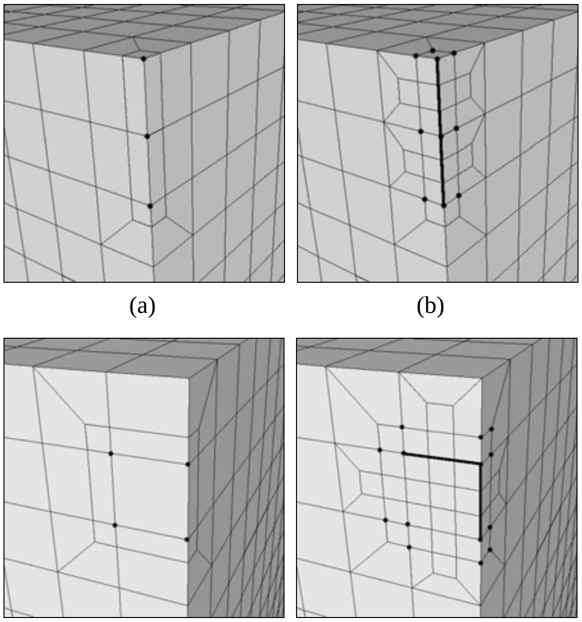


(a) (b) (c)

Figure 13: Modified parallel hex sheet templates.

The two techniques are combined by first refining the selected nodes, element edges and element faces with the templates of Figure 13. For individual nodes, only this first step is needed. Figure 14(a) shows parallel sheet refinement of two edges and the selected nodes (in black) that defined the refinement region. Only templates (b) and (c) were used to perform the refinement. Figure 14(c) shows parallel sheet refinement of a single selected face with corresponding selected nodes. All three parallel sheet templates were used in this step.

Following a single direction of parallel sheet refinement, all eight nodes on the elements adjacent to the target edges or faces are flagged as depicted by the black nodes in Figures 14(b) and (d). Single sheet refinement then follows as described previously with hex sheets defined by the target edges, shown with the heavy black line in Figure 14(b), or two adjacent edges of a target face, shown in Figure 14(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 exits between the selected entity and the transition elements.



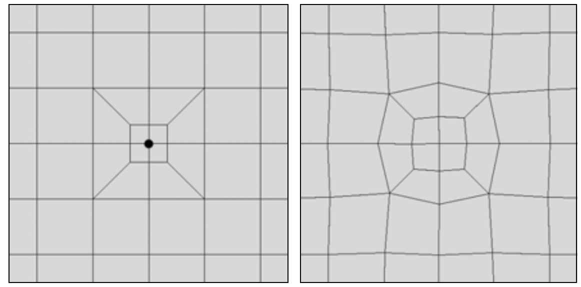
(a) (b)

(c) (d)

Figure 14: Combination refinement: parallel sheet refinement (a and c) followed by multiple directions of single sheet refinement (b and d).

5.5 Examples

Figure 15(a) shows the refinement of a single selected node. Each element surrounding the target node (marked with black) is replaced with template (c) from Figure 13. Figure 15(b) shows the mesh after smoothing.

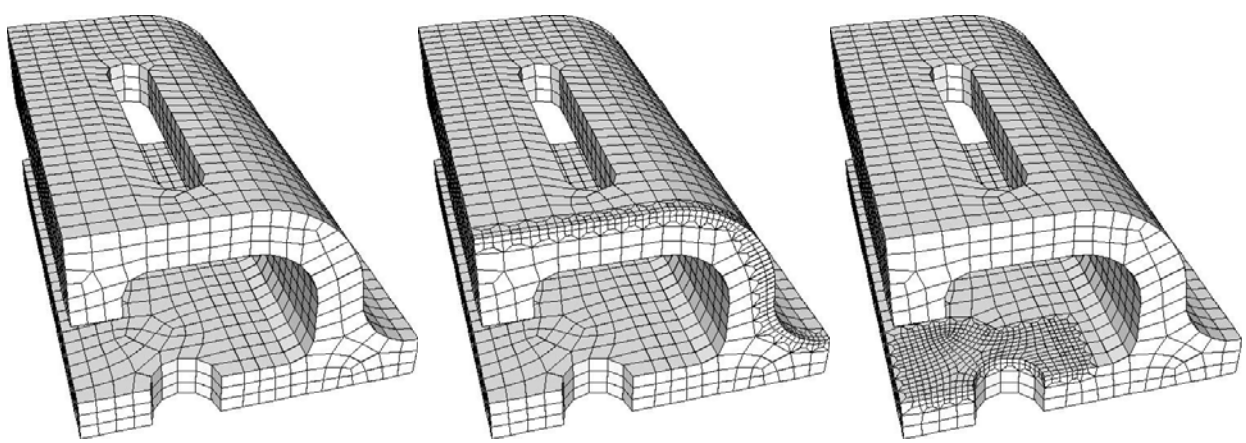


(a) (b)

Figure 15: Single node refinement: refined mesh (a), smoothed mesh (b).

Figure 16(a) is an all-hex conforming mesh. Figure 16(b) shows combination refinement of several edges after smoothing. Figure 16(c) shows combination refinement of a group of faces after smoothing.

Figure 17 compares the three refinement methods along a selected geometric boundary. Figure 17(a) is the original mesh. Figure 17(b) uses combination refinement, Figure 17(c) is refined using parallel sheet refinement, and Figure 17(d) is refined using single sheet refinement. Note that with combination refinement, the depth of the refined region including transition zones is kept within a single original element, while parallel and single sheet refinement spreads over the depth of two original elements. Furthermore, refinement using the combination method and parallel sheet method results in a single layer of good quality elements between the target entity and transition regions. Single sheet refinement completely refines all the elements on the boundary and then places the transitions elements in the next layer.



(a) (b) (c)

Figure 16: Combination refinement: Original mesh (a), selected edges (b) and selected faces (c).

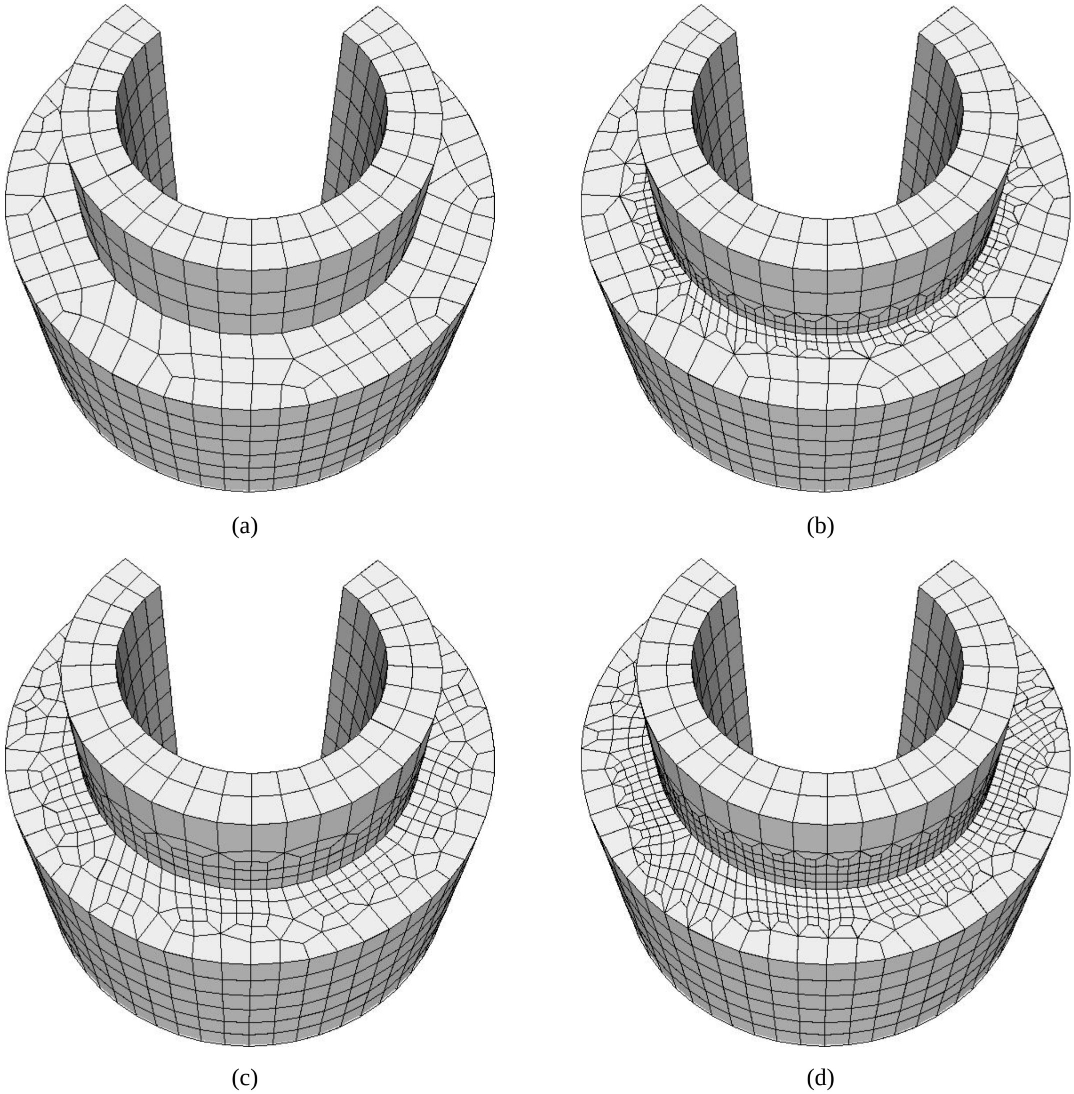


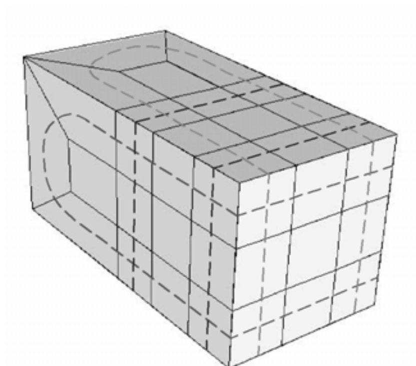
Figure 17: Refinement Comparison: original mesh (a), single-parallel sheet combination refinement (b), parallel sheet refinement (c), single sheet refinement on curve (d).

6. RESULTS AND DISCUSSION

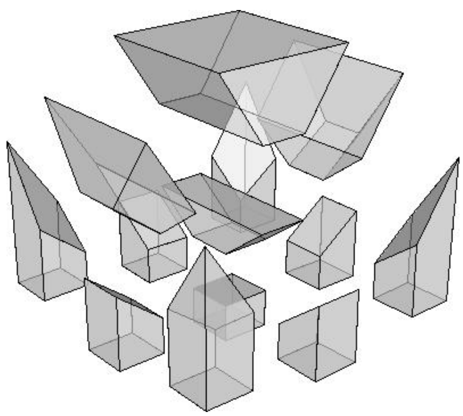
The refinement process always introduces elements with quality lower than the original elements in the transition regions between coarse and fine mesh. The quality is degraded most in areas of overlapping transition templates. Figure 18 displays a common situation in single sheet refinement where two (b) templates from Figure 5 intersect the same transition element from two directions. This occurs when two closing twist planes intersect perpendicularly as shown by the dashed lines in Figure 18(a). The exploded view of the transition region in Figure 18(b) shows several lower quality elements.

For parallel sheet refinement, the intersection of transition regions is simpler to avoid making the quality in the transition regions somewhat better than in single sheet refinement, see Figure 17(c).

Smoothing algorithms can be used to improve the quality of the mesh after refinement through node relocation. Smoothing tools that operate on quality metric objective functions such as the mean ratio metric or condition number metric [8] work best for improvement of mesh quality. As displayed in tables 2 to 4, smoothing after single sheet refinement resulted in minor increases in quality. Smoothing combination and parallel sheet refined meshes significantly increased mesh quality. The examples contained in this paper were smoothed with Mesquite's mean ratio smoother within CUBIT [9].



(a)



(b)

Figure 18: Overlapping transition zones in single sheet refinement (a) and exploded view of transition region (b).

Tables 2 to 4 give the shape quality metric of the examples in this paper for comparison before and after refinement. The shape quality metric equals one if the element is ideal and zero if the element is degenerate [8].

Table 2: Comparison of refined meshes in Figure 7 using the shape quality metric.

Figure	Num. Hexes	Ave.	Std. Dev.	Min.	Max.
7(a)	1248	0.932	0.070	0.726	0.999
7(b)	4448	0.789	0.128	0.294	0.999
7(b) smoothed	4448	0.818	0.184	0.312	0.999
7(c)	2112	0.851	0.190	0.278	0.999
7(c) smoothed	2112	0.867	0.161	0.319	0.999
7(d)	4212	0.851	0.205	0.207	0.999
7(d) smoothed	4212	0.863	0.180	0.266	0.999

Table 3: Comparison of refined meshes in Figure 16 using the shape quality metric.

Figure	Num. Hexes	Ave.	Std. Dev.	Min.	Max.
16 (a)	3093	0.942	0.061	0.572	0.997
16 (b)	3333	0.919	0.113	0.354	0.997
16 (b) smoothed	3333	0.925	0.088	0.535	0.998
16 (c)	3940	0.883	0.163	0.128	0.997
16 (c) smoothed	3940	0.895	0.134	0.163	0.998

Table 4: Comparison of refined meshes in Figure 17 using the shape quality metric.

Figure	Num. Hexes	Ave.	Std. Dev.	Min.	Max.
17 (a)	1116	0.844	0.070	0.612	0.934
17 (b)	1962	0.721	0.208	0.306	0.952
17 (b) smoothed	1962	0.775	0.107	0.447	0.947
17 (c)	2135	0.711	0.174	0.417	0.954
17 (c) smoothed	2135	0.742	0.128	0.416	0.955
17 (d)	5828	0.715	0.218	0.166	0.995
17 (d) smoothed	5828	0.754	0.198	0.257	0.998

7. CONCLUSION

A refinement technique based on the insertion of twist planes into the STC has been presented. This technique is divided into two portions, single sheet operations and parallel sheet operations. Single sheet refinement is effective for dividing single and groups of elements. Parallel sheet refinement is limited by the intersecting and diverging nature of hex sheets in unstructured meshes but when used in conjunction with single sheet operations becomes well suited for refinement of single nodes, element edges and element faces. Both methods use basic templates defined by the number of selected nodes on the original element to replace the element.

The contributions of this work are (1) the localized refinement of mesh regions defined by individual or groups of nodes, element edges, element faces or whole elements within an all-hexahedral mesh, (2) the simplification of template-based refinement into a general method and (3) the use of hex sheets for the management of template insertion in multi-directional refinement.

In the future, parallel sheet refinement could be expanded to the refinement of single and groups of hexahedral elements in structured or nearly structured mesh regions. An algorithm could be developed that determines the relative structuredness of a selected portion of a mesh and, if suitable, refines the region with one-into-eight refinement.

REFERENCES

[1] K.-F. Tchon, C. Hirsch and R. Schneiders, “Octree-based hexahedral mesh generator for viscous flow simulations,” in *13th AIAA Computaional Fluid Dynamics Conference*, no. AIAA-97-1980, (Snowmass, CO), AIAA, June 1997.

[2] M. Borden, S. Benzley, S. Mitchell, D. White, R. Meyers, “The cleave and fill tool: An all-hexahedral refinement algorithm for swept meshes,” in *9th International Meshing Roundtable*, (New Orleans, Louisiana), pp. 69-76, Sandia National Laboratories, Oct. 2000.

[3] L. Marechal, “A new approach to octree-based hexahedral meshing,” in *10th International Meshing Roundtable*, (Newport Beach, CA), pp. 209-221, Sandia National Laboratories, Oct. 2001.

[4] R. Schneiders, “Octree-based hexahedral mesh generation,” *Int. Journal of Comp. Geom. & Applications*, vol. 10, no. 4, pp. 383-398, 2000.

[5] K.-F. Tchon, J. Dompierre, R. Camarero, “Conformal refinement of all-quadrilateral and all-hexahedral meshes according to an anisotropic metric,” in *11th International Meshing Roundtable*, (Ithaca, New York), pp. 231-242, Sandia National Laboratories, Oct. 2002.

[6] “CUBIT Mesh Generation Toolsuite.” URL: http://cubit.sandia.gov, 2004. Sandia National Laboratories.

[7] P. Murdoch and S. Benzley, “The spatial twist continuum,” in *4th International Meshing Roundtable*, pp. 243-252, 1995.

[8] P. Knupp, “Algebraic mesh quality metrics,” *SIAM J. Sci. Comput*., vol. 23, no. 1, 193-218, 2001.

[9] M. Brewer, L.F. Diachin, P. Knupp, T. Leurent, D. Melander, “The mesquite mesh quality improvement toolkit,” in *12th International Meshing Roundtable*, (Santa Fe, New Mexico), pp. 239-250, Sandia National Laboratories, Sep. 2003.