Conformal Refinement of All-Hexahedral Finite Element Meshes

Nathan Harris

Brigham Young University - Provo

Follow this and additional works at: https://scholarsarchive.byu.edu/etd

Part of the Civil and Environmental Engineering Commons

BYU ScholarsArchive Citation
Harris, Nathan, "Conformal Refinement of All-Hexahedral Finite Element Meshes" (2004). All Theses and Dissertations. 3461.
https://scholarsarchive.byu.edu/etd/3461

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.
CONFORMAL REFINEMENT OF ALL-HEXAHEDRAL FINITE ELEMENT MESHES

NATHAN J. HARRIS
CONFORMAL REFINEMENT OF ALL-HEXAHEDRAL FINITE ELEMENT MESHES

by

Nathan J. Harris

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Civil and Environmental Engineering
Brigham Young University
August 2004
This dissertation has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date
7/12/2004

Steven E. Benzley, Chair

Date
6/24/2004

Steven J. Owen

Date
7/12/2004

Alan K. Zundel
As chair of the candidate's graduation committee, I have read the thesis of Nathan J. Harris in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

July 13, 2004

Date

Steven E. Benzley
Chair, Graduate Committee

Accepted for the Department

E. James Nelson
Graduate Coordinator

Accepted for the College

Richard W. Chabries
Dean, College of Engineering and Technology
ABSTRACT

CONFORMAL REFINEMENT OF ALL-HEXAHEDRAL FINITE ELEMENT MESHES

Nathan J. Harris
Department of Civil and Environmental Engineering
Master of Science

Mesh adaptation techniques are used to modify complex finite element meshes to reduce analysis time and improve accuracy. Modification of all-hexahedral meshes has proven difficult due to the unique connectivity constraints they exhibit.

This thesis 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 contributions of this thesis are (1) the ability to conformally refine all entities of an all-hexahedral element mesh, (2) the simplification of template-based refinement into a general method and (3) the application of template insertion to multi-directional refinement.

The refinement algorithm is divided into single hex sheet operations, where individual refinement steps are performed completely within a single hex sheet, and parallel sheet operations, where each refinement step occurs within two parallel hex
sheets. Combining these two procedures facilitates the refinement of any mesh feature. Refinement is accomplished by replacing original mesh elements with one or more of six base templates selected by the number of nodes, flagged for refinement, on the element.

The refinement procedures are covered in detail with representative graphics and examples that illustrate the application of the techniques and the results of the refinement.
# Table of Contents

1 INTRODUCTION .................................................................................. 1

2 HEX MESHING.................................................................................... 5
  2.1 Hex Meshing Challenges ................................................................. 5
     2.1.1 Connectivity Constraints ......................................................... 5
     2.1.2 Quality Constraints .................................................................. 8
     2.1.3 Geometric Constraints .............................................................. 8
  2.2 All-Hex Meshing Techniques ........................................................... 9
     2.2.1 Mapping .................................................................................. 9
     2.2.2 Sub-mapping ........................................................................... 10
     2.2.3 Sweeping ................................................................................ 11
     2.2.4 Whisker Weaving .................................................................... 11
     2.2.5 THexing and HMorphing ........................................................... 12

3 HEX REFINEMENT ............................................................................... 15
  3.1 Classification .................................................................................. 15
  3.2 Current Refinement Techniques ....................................................... 16
     3.2.1 Octrees ................................................................................... 16
     3.2.2 Dicing .................................................................................... 17
     3.2.3 Sheet Insertion ......................................................................... 18
     3.2.4 Template-based Refinement ...................................................... 19
     3.2.5 Multi-directional Refinement with Pillowing .......................... 20
     3.2.6 Contributions of this Thesis ....................................................... 20

4 REQUIREMENTS ................................................................................ 23
List of Tables

5.1 Comparison of refinement techniques for unstructured meshes. An “X” indicates that the method is capable of refining the mesh entity ........................................ 35

9.1 Comparison of refined meshes in Figures 9.3 to 9.6 using the “shape quality” metric ........................................................................................................... 72

9.2 Comparison of single sheet refined meshes in Figures 6.8 to 6.11 using the “shape quality” metric ................................................................. 73

9.3 Comparison of combination refined meshes in Figures 8.5 to 8.7 using the “shape quality” metric ........................................................................... 73
List of Figures

1.1 A coarse mesh of a cracked surface (a) and localized refinement in region of high stress gradient (b) ................................................................. 2
2.1 A stack of hexahedral elements (a), a sheet of hexahedral elements (b) ........... 6
2.2 An all-hexahedral mesh with highlighted sheet (a), a single hex sheet within the mesh (b), the twist plane that defines the sheet (c) ........................................... 7
2.3 An ideal hex element, interior angles equal to \( \pi/2 \) (a), inverted element with interior angle greater than \( \pi \) ................................................................. 8
2.4 Poor quality elements in the region of a small angle ................................... 8
2.5 A mapped volume mesh. (Model courtesy of Matthew Earp) .................... 10
2.6 A sub-mapped volume mesh. (Model courtesy of Matthew Earp) .............. 10
2.7 A swept volume mesh ............................................................................. 11
2.8 A tetrahedral mesh element (a) and THex template (b) ............................ 12
2.9 A complex geometry (a), a tetrahedral mesh (b), a THexed mesh (c) ........ 13
3.1 Non-conforming refinement ........................................................................ 16
3.2 Mesh refined with octrees ........................................................................ 17
3.3 Coarse original mesh (a) and diced mesh (b) ........................................... 17
3.4 A mesh with region selected for pillowing in dark gray (a) shrunken region with new pillow elements in white (b) ...................................................... 18
3.5 A swept mesh (a) and mesh with inserted sheet (b) .................................. 19
5.1 The dual features of a two-dimensional mesh ........................................ 26
5.2 Consecutive dual chord crossing .............................................................. 27
5.3 Self-crossing of a dual chord .................................................................... 27
5.4 Four hex elements with chords and centroids ......................................... 28
5.5 A simple structured mesh (a) and its dual (b and c) ............................... 29
5.6 A simple unstructured mesh (a) and its dual (b) ..................................... 30
5.7 Quadrilateral refinement templates. Main template (a) and transition template (b)...
5.8 Quadrilateral mesh with target region shown in gray (a) and refined mesh (b) .......
5.9 Insertion of templates into transition zones .........................................................
5.10 An all-hex mesh (a) and single sheet (b) ..............................................................
5.11 A hexahedral element (a) one-into-twenty-seven split (b) and one-into-eight split  
   (c) ................................................................................................................................
6.1 A 2D mesh with target elements selected (a) and the chords that define the mesh  
   (b) ................................................................................................................................
6.2 A single chord-loop insertion (a) and resulting mesh (b) and multiple chord-loop  
   insertion (c) and resulting mesh (d) ............................................................................
6.3 An extracted sheet showing a single level of refinement on the target center hex  
   (a) and the mesh after three levels of refinement (b) .................................................
6.4 Single hex sheet refinement templates ........................................................................
6.5 Concavity restrictions of selection region ............................................................... 
6.6 Highlighted target region in area of geometric concavity ........................................ 
6.7 Concavity restrictions of selection region (a) ............................................................
6.8 Single sheet operations: Original mesh ......................................................................
6.9 Single sheet operations: Selected elements ..............................................................
6.10 Single sheet operations: Selected curve and vertex ................................................
6.11 Single sheet operations: Selected surface ..............................................................
7.1 Single chord insertion (a) and resulting mesh (b) and multi-chord insertion (c) and  
   resulting mesh (d) .........................................................................................................
7.2 A single level of lower order refinement (a) twist plane and selected nodes (b) ......
7.3 Parallel hex sheet refinement templates ......................................................................
7.4 Concavity restriction for lower order refinement .....................................................
7.5 Path of the twist plane for the mock template in Figure 7.4 ......................................
7.6 A structured hex mesh with highlighted refinement target region (a) and refined   
   mesh with darkened refined region and highlighted transition zone (b) ......................
7.7 Intersecting and diverging hex sheets (a, c) and corresponding twist plane     
   insertion restrictions (b, d) ...........................................................................................
7.8 An unstructured hex mesh with target node selected for refinement (a), inserted    
   templates in elements adjacent to node (b) and the final smoothed mesh (c) ..........
1 INTRODUCTION

Finite Element Analysis is a tool used to compute approximate solutions to multivariable systems. The finite element method converts the complex differential equations that govern the properties of the system into piecewise approximations that can be effectively solved with the aid of computers. Finite element analysis has been applied to a myriad of physics and engineering applications. As computing power increases, the scale and complexity of problems solved with this method will also increase.

A primary step in the finite element method is to discretize the domain of the system, or in other words, to mesh the geometric model. Much attention has been given to automatic mesh generation. Two-dimensional mesh generators typically produce triangle and/or quadrilateral elements while three-dimensional mesh generators generally employ tetrahedral and/or hexahedral elements. Techniques for producing all-hex meshes will be discussed in detail in Chapter 2.

The accuracy of the analysis is highly dependent on the quality of the mesh. In theory, as the number of mesh elements goes to infinity, the analysis converges to the exact solution. Ideally, a very fine mesh would be suitable for accurate analyses. However, because computation time is directly related to the number of elements in the mesh, accuracy must be sacrificed for computational feasibility in large meshes.

To optimize the accuracy of the analysis, techniques have been developed to adapt the mesh to best fit the physics of the analysis. Adapting the mesh is done by either increasing or decreasing the density of the mesh in localized regions. Before the analysis is run, the user often knows the approximate regions within the model where variable gradients will be largest. An example is shown on the mesh of a cracked two-dimensional domain in Figure 1.1(a). If this model is subjected to a tension force from top and bottom, the stresses will be greatest in the immediate vicinity of the crack tip. In the regions away from the tip, the magnitude of the stresses drops dramatically. The
mesh can be adapted by increasing the mesh density in the region of highest stress gradient as shown in Figure 1.1(b).

![Figure 1.1: A coarse mesh of a cracked surface (a) and localized refinement in region of high stress gradient (b).](image)

If the characteristics of the gradient are not known at the outset, a preliminary analysis could be done on a coarse mesh of the model. An adaptive, refined mesh can then be generated as dictated by the results. After adaptation, the analysis is performed again and the process repeated until the results converge.

Over the years, much attention has been given to 2D and all-tetrahedral mesh adaptation. Refinement and coarsening techniques have been developed that effectively increase or decrease mesh density in localized regions. However, due to several inherent complexities, few refinement methods have been developed for all-hex meshes.

This thesis presents a method for localized refinement of all-hex meshes based on a twist plane insertion technique that will be presented in Chapter 6. This method has been implemented within CUBIT, a meshing tool kit written and maintained by Sandia National Laboratories* [1].

---

*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.
The contributions made by this thesis are (1) the ability to conformally refine all entities of an all-hexahedral mesh, (2) the simplification of template-based refinement into a general method and (3) the application of template insertion to multi-directional refinement.
2 HEX MESHING

In many situations, hexahedral elements are superior to tetrahedra. For example, for a given number of degrees of freedom, hexahedral elements have been shown to converge more quickly in analysis. Also, hexahedral elements may incorporate directionality into a mesh and more accurately capture geometry [2].

Despite years of effort, the automatic generation of all-hex meshes in arbitrary geometries is still not a reality. Several unique challenges inherent in creating all-hex meshes have kept researchers from attaining this elusive goal. These challenges are described in detail in section 2.1. Many useful techniques for creating all hex meshes through significant user intervention or automatically for simple geometries have been developed and are discussed in section 2.2.

2.1 Hex Meshing Challenges

All-hex mesh generation faces three major challenges dealing with connectivity, quality and geometric sensitivity [3].

2.1.1 Connectivity Constraints

The structure of a hexahedron can be viewed as three pairs of opposing quadrilateral faces. Each face of a hexahedral element within a conforming mesh must either be located on a boundary or border an equally dimensioned face of a neighboring element. The opposite face of the neighboring element then follows the same restriction and so on to form a stack of elements as seen in Figure 2.1(a). Such stacks must begin and end at geometric boundaries or else form completely closed loops within a mesh. These hex stacks can be extended to hex sheets by expanding the connectivity of each element in the stack in a second direction as shown in Figure 2.1(b).
All-hex meshes, like the mesh in Figure 2.2(a), are composed of inter-laced hex sheets that are a feature of the hex element connectivity. A single hex sheet is highlighted in the mesh shown in Figure 2.2(a) and displayed alone in Figure 2.2(b). The hex sheet can be described by its dual representation, the twist plane. The twist plane shown in Figure 2.2(c) is a three-dimensional surface that represents the hex sheet and is useful for describing the orientation of the hex sheet in space and its interconnectedness with other sheets in the mesh. The dual of the mesh is described in further detail in Chapter 5.
Due to the nature of the element connectivity, an individual hex element cannot be inserted or removed from an all-hex mesh without inserting or removing an entire sheet. Furthermore, any modification to an element, such as refinement or removal, that extends to the border of a neighboring element also modifies that element. For these reasons, the connectivity characteristics of all-hex meshes have significant refinement and coarsening implications.
2.1.2 Quality Constraints

The quality of the individual elements within the mesh impact the accuracy of the analysis. The ideal hex element, shown in Figure 2.3(a), has interior angles of $\pi/2$ between all adjacent faces. Should any interior angle approach or exceed $\pi$, as shown in Figure 2.3(b), the accuracy of the element becomes unacceptable.

![Figure 2.3: An ideal hex element, interior angles equal to $\pi/2$ (a), inverted element with interior angle greater than $\pi$.](image)

2.1.3 Geometric Constraints

Hex elements are sensitive to the geometry of the model. All surfaces of the model are meshed with quadrilateral meshes that must conform to the geometry. Some features, such as the small angle shown in Figure 2.4, are difficult to fit with good quality elements. Poor surface mesh characteristics may then be propagated into the volume mesh forming layers of distorted elements.

![Figure 2.4: Poor quality elements in the region of a small angle.](image)
2.2 All-Hex Meshing Techniques

Despite the challenges discussed above, several techniques for creating all-hex meshes have been developed. The following discussion of hex-meshing schemes is not comprehensive, but does reflect the state of the art. Understanding the principles upon which these schemes are based and the characteristics of the meshes they produce are necessary to recognize the capabilities and limitations of the refinement technique presented in this work.

Mapping and sub-mapping schemes, discussed in sections 2.2.1 and 2.2.2 respectively, create structured meshes. On structured surface meshes, all nodes not located on surface boundaries are connected to exactly four quadrilaterals. In structured volume meshes, all nodes not located on volume boundaries are connected to exactly eight hex elements. The hex sheets that compose structured meshes, and the dual twist planes that represent them, are in parallel layers, a feature important to the refinement technique herein discussed.

The hex generation schemes discussed in sections 2.2.3 to 2.2.5 generally produce unstructured meshes. In unstructured meshes, the nodes may be connected to more or less elements than in structured meshes. The hex sheets that compose unstructured meshes are not always parallel but instead intersect and diverge within the mesh.

2.2.1 Mapping

Mapping algorithms create regular meshes on volumes with six logical surfaces. Such algorithms identify the logical boundaries defining each surface, then divide opposing surface boundaries into equal intervals. After the intervals are correctly set, the algorithms fit two-dimensional grids to all surfaces of the model followed by a three-dimensional grid to the volume. The mesh in Figure 2.5 was created using a mapping algorithm. As depicted in the figure, the model boundaries need not be linear. However, highly irregular shapes can distort the grid, thereby introducing poor element quality.
2.2.2 Sub-mapping

Sub mapping schemes take mapping a step further by breaking complex geometries, either manually or automatically, into several virtual mappable volumes [4]. Each volume is then meshed as described above. The model in Figure 2.6 was meshed automatically using a sub-mapping scheme.
2.2.3 Sweeping

Sweeping schemes produce meshes on prismatic geometries by creating a mesh on a source surface (labeled “s” in Figure 2.7), then projecting the mesh parallel to the axis of the prism. Layer after layer is added to the prism until the target surface (labeled “t” in Figure 2.7) is reached. The surfaces that connect source and target surfaces are called linking surface (labeled “l” in Figure 2.7) and are always meshed with structured meshes.

![Figure 2.7: A swept volume mesh.](image)

Sweeping algorithms dominate current hex meshing schemes. Much research and effort has expanded state of the art algorithms to effectively mesh geometries with non-planar surfaces, variable cross sections [5] and multiple source and target surfaces [6][7][8][9][10].

2.2.4 Whisker Weaving

The whisker-weaving scheme [11] produces an all-hex volume mesh from quadrilateral meshes initially constructed on all geometric surfaces. This technique constructs elements within the model using the properties of the mesh dual that will be discussed in chapter 5. Whiskers are projected from the centroids of the surface elements to the interior of the volume where they are connected to other whiskers. After cleanup procedures on the connectivity of the whiskers are performed, the hexahedral elements are created based on the interconnectedness of the whiskers. The whisker weaver can
create meshes on non-sweepable geometries, but the quality of the resulting mesh cannot be guaranteed.

2.2.5 THexing and HMorphing

The THex [12] and HMorph [13] schemes are both examples of indirect hex meshing. The schemes discussed above form hexahedral elements directly within the volumes. Indirect meshing methods initially form other types of elements, tetrahedral elements in the case of THexing and HMorphing, and then transform these elements into hexahedrons. THexing replaces each tetrahedron, Figure 2.8(a), with the template shown in Figure 2.8(b). Each tetrahedron is divided into four hexahedra.

![Figure 2.8: A tetrahedral mesh element (a) and THex template (b).](image)

The model in Figure 2.9(a) is difficult to mesh using current hex-meshing schemes. Instead, the model is first meshed automatically with tetrahedra, Figure 2.9(b), and then "THexed" to create an all-hex mesh in Figure 2.9(c).
Figure 2.9: A complex geometry (a), a tetrahedral mesh (b), a THexed mesh (c).

Instead of using a division template, H Morphing uses transformations to combine tet elements to form hexes. This process begins at the surface working inwards until no more transformations can be accomplished. The resulting mesh may consist of multiple element types.
3 HEX REFINEMENT

Several techniques have been developed for the refinement of finite element meshes. These techniques can be divided into three major categories that are presented in Section 3.1. Section 3.2 discusses several hex refinement methods and then concludes with a summary of the contributions made in this work.

3.1 Classification

Finite element mesh refinement methods can be divided into the following three categories:

1. h-refinement: Add additional nodes to increase the number of elements in the mesh.
2. r-refinement: Reposition the nodes to form denser mesh in regions of interest. Node and element counts are equal before and after refinement.
3. p-refinement: Change the order of the elements by adding additional nodes to all the elements within the mesh (for example an 8-node hex to a 27-node hex). The mesh element count is equal before and after refinement.

As required for inclusion within CUBIT, the refinement scheme presented in this work is h-refinement. Chapter 4 discusses the other requirements that governed the direction of this research. Another requirement is that the refinement be performed on meshes that are conforming both before and after the refinement process. To understand mesh conformity, note the non-conforming mesh in Figure 3.1. The mesh has been refined in the lower right quadrant by dividing the original elements into four new elements. The new element edges are non-conforming at points a through d. The nodes
that exist at these points are called hanging nodes. Figure 1.1(b) from Chapter 1 is an example of a conformal refinement of a surface mesh.

In three-dimensional conforming meshes, all element faces not on boundaries must be bordered entirely by a single face of a neighboring element. In the finite element analysis process, non-conforming meshes can cause incorrect analysis results in the vicinity of the hanging nodes or, for some solvers, be unsolvable.

3.2 Current Refinement Techniques

Several h-refinement methods have been developed for hex meshes and are discussed in the following sections.

3.2.1 Octrees

In terms of hex refinement, octrees are the division of a single element into eight new elements by splitting all twelve edges of the element at their midpoints. The mesh is refined by iteratively inserting octrees into the mesh until the desired element size is reached in the area of interest. Figure 3.2 shows refinement at the tip of the model by inserting octrees in two levels to increase mesh density. While this method allows for excellent control over the scale and breadth of the refinement, the resulting mesh is non-conforming and therefore unsuitable in many cases [14].
3.2.2 Dicing

Dicing is used to divide individual sheets within the mesh along the plane of the sheet [15]. All the sheets within the mesh of Figure 3.3(a) are split to form the mesh in Figure 3.3(b).
While the resulting mesh is conforming, this method does not allow for local refinement to minimize the total number of elements in the mesh.

3.2.3 Sheet Insertion

Sheet insertion refinement is based on pillowing. Pillowing, or buffer insertion, is used to add elements into a mesh by inserting a layer of elements around a selected target region, shown in dark gray in Figure 3.4(a) [16]. This is accomplished by first, separating the elements to be pillowed from their neighboring elements, shrinking them, then adding a new layer of elements into the void, shown in white in Figure 3.4(b). Pillowing within an all-hex volume mesh inserts a sheet of elements which, as discussed in Section 2.1.1, is required to maintain mesh conformity.

![Figure 3.4: A mesh with region selected for pillowing in dark gray (a) shrunken region with new pillow elements in white (b).](image-url)
The cleave-and-fill tool [17], inserts new conforming elements into a mesh through pillowing semi-structured elements in the mesh regions between source and target surfaces of swept meshes. The mesh in Figure 3.5(a) was created by generating a mesh on the top surface of the model and sweeping it downward to create four layers (sheets) of elements. Because of the tapered nature of the source and target surfaces, the elements on the left side of the mesh are larger than those on the right. To remove this size difference, portions of two layers in the mid-plane of the mesh have been pillowed in Figure 3.5(b).

While local, conformal refinement can be accomplished with the cleave-and-fill tool, much user intervention is required. In addition, because only the swept portion of the mesh can be pillowed automatically, the control and scale of refinement are limited.

![Figure 3.5: A swept mesh (a) and mesh with inserted sheet (b).](image)

### 3.2.4 Template-based Refinement

Several geometry division templates proposed by Esmaeilian [18] divided six-sided regions into multiple subregions that transition between coarse and fine. The subregions are then fitted with mapped meshes to create a conformal refined mesh. This method offers local conformal refinement, but is limited to geometries with six surfaces. Many of the proposed templates are very complex and the number of templates that could be constructed is unlimited.

An all-hex, octree-based mesh generator introduced by Schneiders et al. [19] produces a mesh by first overlaying a geometric model with an octree then iteratively refining the octree in regions where the geometry is not captured by the initial coarse elements. Elements within the overlay that cross geometric boundaries are removed and a
buffer layer is inserted between the surfaces and the inner volume mesh. The most interesting feature of this technique with respect to this work is the method of octree refinement used. Several templates were proposed that could replace elements of the octree to create a conformal mesh. Though these templates were proposed for use in the initial mesh generation, they can be used for the local, conformal refinement of structured meshes.

3.2.5 Multi-directional Refinement with Pillowing

Schneiders' proposed a directional refinement method [20] that produces a conformal mesh by pillowing in alternating i, j and k directions. Each direction of pillowing reduces element size in the pillowed region and increases mesh density. This method was proposed for refinement of structured meshes.

The 3D anisotropic refinement scheme presented by Tchon et al. [21] expands Schneiders' multi-directional refinement to initially unstructured meshes by pillowing layers (sheets) of elements according to a size specification map. This method is capable of local, conformal refinement, offers excellent control of the refinement region and is very robust. However, this method does not allow for immediate refinement around individual nodes, element edges or element faces.

3.2.6 Contributions of this Thesis

The refinement technique presented in this work makes the following contributions to hex mesh refinement:

1. The ability to conformally refine all entities of an all hexahedral element mesh.
2. The simplification of template-based refinement into a general method.
3. The application of template insertion to multi-directional refinement.

The technique presented in this work takes full advantage of the robustness of multidirectional refinement by dividing the process into independent refinement directions. This technique improves upon the 3D anisotropic scheme discussed above by adding the ability to refine regions of the mesh that are smaller than entire hex elements or are more
easily described as the regions around nodes, element edges and element faces of the mesh.

Only six simple templates are necessary for the refinement of any mesh feature. These are used in a general method that is capable of refining any mesh regardless of its degree of unstructuredness. The insertion of templates into the mesh replaces pillowing of the 3D anisotropic scheme and is accomplished using a “node-based” technique where a template is chosen and oriented within the mesh based on the number and location of flagged nodes on the element to be replaced.
4 REQUIREMENTS

The refinement technique described in this work was developed to meet the following requirements necessary for inclusion within the CUBIT framework.

1. H-refinement scheme
2. All-hex refinement
3. Refinement of unstructured meshes
4. Conforming refinement
5. User specified or analysis driven
6. Local refinement
7. Node-based refinement

The refinement tool herein presented was developed using an h-refinement scheme for use on meshes that are all-hex, unstructured and conforming both before and after refinement. Both h-refinement and conforming meshes were discussed in Chapter 3. It was developed for use both within CUBIT's interactive user interface and within a mesh adaptivity module intended to be invoked during computational analysis.

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 would be 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 (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 required. These parameters are displayed in the examples of Chapter 10.

1. Numsplit: The number of times to subdivide each element.
2. Size: The target element size specified by the user.
3. Bias: The maximum change in element size from the specified size to the pre-existing background size.
4. Sizing Function: The automatic 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 in triangle, quadrilateral and tetrahedral refinement within the existing framework.
5 OVERVIEW

Before presenting the refinement technique introduced in this work, the concepts of a target refinement region and the dual of a mesh must be understood. The following sections discuss these then present an overview of the algorithm.

5.1 Target Regions

To initiate refinement, a target region is first specified. The target region is the portion of an existing mesh where the density is to be increased. The target is specified by selecting one or more geometric or mesh entities. Selection may be done either interactively, as discussed in Chapter 4, by selecting individual mesh or geometric entities, 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.

5.2 The Dual

The refinement method presented in this thesis is based on the modification of the dual of a mesh. The dual is a geometric representation that explicitly defines the connectivity characteristics of a mesh [22]. The dual is first presented for a two-dimensional mesh and then expanded to three-dimensions. In this explanation, the dual of the mesh and the spatial twist continuum (STC) are used interchangeably.

5.2.1 The Dual In Two Dimensions

The dual of a quadrilateral mesh is composed of zero-dimensional centroids, one-dimensional chords and two-dimensional polygons called 2-cells. All of these components are visible in Figure 5.1. Centroids are depicted with gray circles at the
center of each element. Chords are shown as dashed lines interconnecting the centroids. 2-cells are the polygons bounded on all sides by dual chords.

The mesh is related to the dual in the following manner. Nodes within the mesh are located at the center of each 2-cell. Each element edge crosses a single chord between nodes. The number of edges of the 2-cell dictates the valency (the number of edges attached) of the node.

![Figure 5.1: The dual features of a two dimensional mesh.](image)

Murdoch presented a set of constraints based on the dual that are used to determine the validity of a quadrilateral mesh [22].

1. A chord that begins on a boundary must terminate on a boundary. An example of this is seen in the chord labeled “1” in Figure 5.1.
2. A chord that does not begin on a boundary must form a closed loop within the mesh. Chord “2” in Figure 5.1 is an example.
3. A given chord may cross another particular chord multiple times, but crossings may not be consecutive. This constraint ensures that two quadrilaterals will not share two edges as shown in Figure 5.2.
4. A chord may cross itself provided each self-intersection is separated by four other centroids as shown in Figure 5.3.
5. Only two chords may intersect a centroid. This constraint ensures that each element has only four edges.
6. Chords are nowhere tangent.

Figure 5.2: Consecutive dual chord crossing.

Figure 5.3: Self-crossing of a dual chord.
5.2.2 The Dual in Three Dimensions

The concept of dual centroids, chords and 2-cells can be directly expanded to three dimensions. Dual chords and centroids of a group of hexahedral elements are shown in Figure 5.4. The centroids are located at the centers of the hexahedral elements. Chords pass through the three pairs of opposing faces on each hex and intersect at the centroid (an unseen third chord intersects the centroid of each element in Figure 5.4, passing through the element's top and bottom faces). As with two-dimensional duals, centroids and chords form the corners and sides of 2-cell polyhedra. A four-sided 2-cell is visible within the crossing chords in Figure 5.4. In three dimensions, 2-cells are grouped to form the faces of polyhedrons called 3-cells.

![Figure 5.4: Four hex elements with chords and centroids.](image_url)

In addition to 3-cells, another component of the dual unique to three dimensions is the twist plane. The interweaving plane of chords in Figure 5.4 forms such a twist plane. A twist plane is a continuous three-dimensional surface that represents a layer (or sheet) of elements within the mesh. Twist planes are illustrated in Figure 5.5. Figure 5.5(a) is a simple structured mesh. The white lines that cross through the faces of the elements in Figure 5.5(b) show the intersections between the twist planes and the model boundaries. All twist planes that represent the mesh are shown in Figure 5.5(c). The individual planes clearly show the inter-connectedness of the hex sheets.
Figure 5.5: A simple structured mesh (a) and its dual (b and c).
Using twist planes as the basis, 3-cells can be defined as the division volumes bounded by twist planes, chords are defined as the intersection of two twist planes and centroids are defined as the intersection of three twist planes. Except for constraint 5, the chord constraints presented in Section 5.2.1 also govern twist planes.

Figure 5.6 depicts the intertwined nature of twist planes in an unstructured mesh. Visualization of the twist planes emphasizes the connectivity constraints discussed in Chapter 2 which stated that individual elements can only be added to or removed from a mesh by adding or subtracting an entire sheet of elements.

The refinement method presented in this work was developed by determining the optimal locations to insert twist planes into the spatial twist continuum.

Figure 5.6: A simple unstructured mesh (a) and its dual (b).

5.3 Quadrilateral Refinement

The original focus of this work was to develop a hexahedral-mesh refinement scheme that would mirror the quadrilateral mesh refinement scheme currently implemented within CUBIT. This technique uses two templates (Figure 5.7) to replace original quadrilateral elements within the target refinement area.
The quadrilateral mesh in Figure 5.8(a) is to be refined in the target region consisting of several elements shown in dark gray. Refinement is done by first replacing each target element with the template shown in Figure 5.7(a). The algorithm then selects an element from the layer of elements bordering the target area and marches from element to element through the layer until a boundary is reached or it makes a complete loop through the mesh, returning to the starting location. The layer of elements just outside the target region is called the transition zone. This layer forms the transition between coarse and fine mesh. The transition zone is shown as the white layer of elements in Figure 5.8(b).

As it marches through the transition zone, the algorithm either replaces each element with one of the templates from Figure 5.7 or leaves it unchanged. For the mesh in Figure
5.8(b), two distinct transition zones exist, one above the target region (labeled with a “1” at beginning and end) and one below the target region (labeled “2”). To illustrate how a template is chosen for insertion into the transition zone, follow zone 1 from the right-side boundary. The first element, shown darkened in Figure 5.9(a), must be split at its bottom edge. Template b can be inserted with its orientation as shown in Figure 5.7(b) or rotated from the above position 90 degrees counter-clockwise and inserted. In this case, the latter is chosen arbitrarily. The next element, shown in Figure 5.9(b), has two adjacent sides split (left and bottom). Only one orientation of template b will replace this element. The third element along the path, shown in Figure 5.9(c) is labeled with an “n”. This element requires no transition element because none of its edges have been split. For an element in the transition zone to be left unchanged, it may not have an edge that borders a target element.

![Figure 5.9: Insertion of templates into transition zones.](image-url)
After the algorithm finishes traversing the first transition zone, it begins again at the right side of zone 2. Three elements into the transition zone from the right, an element is encountered that has three split edges, Figure 5.9(d). In this region, template “a” of Figure 5.7 must be inserted. This situation always exists when a transition element nested within a concave region of the target area is also bounded by the split side of an adjacent template.

The algorithm continues to march along the transition zones replacing elements as needed. As long as the path along each continuous string of transition elements is not interrupted, the transition elements will always align the coarse and fine mesh regions to produce a conforming mesh.

A hexahedral mesh refinement scheme that mirrors this technique would divide the face of each target hex in the same manner that template 5.7(a) divides quadrilaterals, thus dividing the hex into eight. This method would be superior to techniques that divide the edges of target hexes into three, which results in a one-to-twenty-seven split, because it would give the user greater control of the element size.

Chapter 7 discusses the fact that a refinement scheme for hexahedral meshes that always produces the same results on mesh surfaces as the quadrilateral refinement scheme just presented cannot be constructed. A hexahedral refinement scheme, called parallel sheet refinement, was developed that can produce similar results in structured meshes. However, a more robust scheme that expands parallel sheet refinement was developed that, like the quad-refinement method, is capable of local conformal refinement of unstructured all-hex meshes.

5.4 Hex Refinement Algorithm Overview

The hex refinement algorithm is divided into two distinguishing procedures; single and parallel hex sheet operations. Hex sheets have been defined as all the elements on a single twist plane of the spatial twist continuum. A single hex sheet is highlighted in the mesh of Figure 5.10(a) and displayed in Figure 5.10(b). Single sheet refinement operations refer to the execution of one level of refinement within a single hex sheet while each level of refinement in parallel sheet operations takes place within two neighboring sheets. Parallel sheet operations perform refinement similar to that seen in
the quadrilateral refinement technique discussed above, however, because of all-hex mesh connectivity characteristics, it is too constrained for unstructured meshes.

![Image](image_url)

**Figure 5.10:** 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, Figure 5.11(a), into twenty-seven hexes as shown in Figure 5.11(b). Parallel sheet refinement divides each edge into two, splitting a single element into eight hexes as shown in Figure 5.11(c).

![Image](image_url)

**Figure 5.11:** A hexahedral element (a) one-into-twenty-seven split (b) one-into-eight split.
Table 5.1 compares the refinement procedures presented in this paper based on the mesh entities each can refine. Mesh entities, labeled across the top of the table, are the constituents that compose a mesh. Nodes are the basic mesh entities. Edges are defined by two nodes. Faces can be defined by either four nodes or four edges. Hexes are defined by eight nodes, twelve edges or six faces.

Table 5.1: Comparison of refinement techniques for unstructured meshes. An “X” indicates that the method is capable of refining the mesh entity.

<table>
<thead>
<tr>
<th></th>
<th>Hex</th>
<th>Face</th>
<th>Edge</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Sheet</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parallel Sheet</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Combination</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

As summarized in the table, single and parallel sheet refinement techniques are limited as to the entities each is capable of refining. By introducing a third “combination” approach and coupling this with single sheet refinement, all mesh entities are then capable of refinement.

The steps of the complete algorithm are outlined as follows:

Step 1. Select target region.
Step 2. Check for targets consisting of nodes, edges or faces.
   If yes, go to Step 3.
   If no, go to Step 5.
Step 3. Refine around single nodes by inserting appropriate parallel sheet templates.
Step 4. Insert parallel sheet refinement templates around target edges and faces.
Step 5. Locate sheets that contain elements in target region.
Step 6. Loop through sheets:
   Remove concavities in sheet.
   Insert single sheet refinement templates.
6 SINGLE HEX SHEET OPERATIONS

This chapter introduces the theory supporting single hex sheet refinement, the templates used during the refinement process, the restrictions of this method and examples of refinement.

6.1 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 6.1(b).

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{6_1}
\caption{A 2D mesh with target elements selected (a) and the chords that define the mesh (b).}
\end{figure}

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 6.2(a) and (c). Each new intersection between the inserted chord and an original chord defines a new element, Figure 6.2(b) and (d).

In two dimensions, two directions of refinement divide each target quadrilateral into nine new quadrilaterals shown in Figure 6.2(d). Each inserted chord loop represents a single level of refinement. Note that each level 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.

Figure 6.2: 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 level 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 6.3(a) shows an individual hex sheet wherein one level of refinement has occurred.

![Figure 6.3](image)

Figure 6.3: An extracted sheet showing a single level of refinement on the target center hex (a) and the mesh after three levels of refinement (b).

Three directions of refinement divide the central target hex of Figure 6.3(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. The refinement region is extended by inserting additional twist planes into neighboring sheets.

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

### 6.2 Templates

Only three templates are needed to perform each direction of refinement, a main template, Figure 6.4(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 will be refined. These selected nodes are, for each template, marked with black dots in Figure 6.4. 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).

![Figure 6.4: Single hex sheet refinement templates.](image)

6.3 Restrictions

A single hex sheet refinement target region is restricted to a convex shape. An example of this restriction is shown in Figure 6.5. Figure 6.5(a) shows the hexes selected for refinement. In Figure 6.5(b), one twist plane has been inserted in the illustrated hex sheet. The template necessary to fill the concave region indicated requires the surface characteristics of the mock template in Figure 6.5(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 6.5(d).
If the target refinement region happens to correspond with an area of geometric concavity, like the highlighted target region seen in Figure 6.6, then the resulting refinement will also be concave.
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 need to be extended to all hexes attached to the node to maintain at least one layer of good quality hexes, i.e., no transition hexes, immediately surrounding the node. As shown in Figure 6.7(a) the refinement is spread into the neighboring elements. Extending the refinement region in this manner may not be desirable. Template (c) could be used to refine elements around a selected edge, as shown in Figure 6.7(b), but would result in lower quality elements within the target refinement region.
6.4 Algorithm Description and Examples

Due to the above restrictions, refinement using single sheet operations is best 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 6.8 is composed of 1,258 hex elements and is conforming. Figure 6.9 shows the refinement of a group of selected hexes. The total number of elements has been increased to 4,448. Figure 6.10 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 6.10. The element count in this mesh is increased to 2,112. Figure 6.11 shows the refinement of a single surface with 4,212 total elements.

Figure 6.7: Concavity restrictions of selection region (a).
Figure 6.8: Single sheet operations: Original mesh

Figure 6.9: Single sheet operations: Selected elements.
Figure 6.10: Single sheet operations: Selected curve and vertex.

Figure 6.11: Single sheet operations: Selected surface.
7 PARALLEL HEX SHEET OPERATIONS

This chapter introduces the theory supporting parallel hex sheet refinement, the templates used during the refinement process, the restrictions of this method and examples of refinement. The limitations of parallel hex sheet refinement discussed in Section 7.3 dealing with concave target regions and the non-parallel nature of hex sheets within unstructured meshes are described to explain why parallel hex sheet refinement is too constrained to mirror the quadrilateral refinement method discussed in Chapter 5.

7.1 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 7.1(a) shows examples of parallel sheet refinement reduced to two dimensions. In the figure, a single chord loop (the black dashed line) is inserted that circumscribes the nodes of the target elements shared between two parallel columns of elements representing sheets. The corresponding mesh, Figure 7.1(b), shows that two sheets of elements are modified in this single level of refinement.

Figure 7.1(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 7.2(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 elements affected by the refinement can be increased by inserting additional twist planes into the mesh.
Figure 7.1: Single chord insertion (a) and resulting mesh (b) and multi-chord insertion (c) and resulting mesh (d).
7.2 Templates

The refinement process is divided into multiple refinement levels. In each level of refinement, a single twist plane loop is inserted into the mesh. At each level, the refinement zone can be described by a set of selected nodes shared by two neighboring hex sheets, shown in solid black dots in Figure 7.2(b). 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 7.3(a), is defined by four selected nodes on a single 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 7.2(b).
7.3 Restrictions

As with single hex sheet refinement, no simple template can be constructed to effectively accommodate target region concavity, seen in Figure 7.4(a). Such a template would have the surface characteristics shown in figure 7.4(b). The dashed lines in Figure 7.5 represent the intersections between the twist planes and the element faces. Choosing any point on a dashed line and then following the line around the template will eventually trace back to the starting point. This template attempts to merge the two twist planes that enter the bottom face of the original element. Modifying the twist planes in such a manner results in unacceptable element quality within the template.
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. An example of parallel sheet refinement is shown in Figure 7.6. The mesh on the simple cube model in Figure 7.6(a) is structured. Figure 7.6(b) shows completed refinement of the highlighted region of Figure 7.6(a). Though the meshes produced with this technique are legitimate, the parallel sheet restriction limits its usefulness because many complex models cannot be fitted with a structured mesh. Furthermore, once a single refinement iteration has taken place within the mesh, the mesh is no longer structured. Further refinement in the unstructured region could not be accomplished. In contrast, single sheet refinement operations can effectively refine both structured and unstructured meshes.
Figure 7.6: A structured hex mesh with highlighted refinement target region (a) and refined mesh with darkened refined region and highlighted transition zone (b).
In unstructured meshes, neighboring hex sheets, defined by the dashed lines in Figure 7.7, can intersect (a) or diverge (c) within the mesh. In the case of intersecting sheets, the inserted twist plane, shown with the black dashed loop in Figure 7.7(b), 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 7.4(b). 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, Figure 7.7(d), confines the transition region to the fewest number of elements.

Figure 7.7: Intersecting and diverging hex sheets (a, c) and corresponding twist plane insertion restrictions (b, d).
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.

Because of the above restrictions, parallel sheet refinement is only suited for mesh regions where hex sheets are parallel (i.e. structured regions) or where sheet orientation is not an issue. Such conditions occur for the refinement of nodes and element edges.

7.4 Examples

The unstructured mesh in Figure 7.8(a) is to be refined at the node marked with the black dot. Template (c) of Figure 7.3 is used to refine the elements surrounding the node, Figure 7.8(b). As evident in the figure, template (c) can refine the node regardless of the number of elements connected to it. The mesh is then smoothed in Figure 7.8(c). In the case of single node refinement, hex sheet orientation has no influence on refinement pattern. The results of this refinement method are identical to those generated by the quadrilateral mesh refinement scheme presented in Chapter 5 for single node refinement.

![Figure 7.8: An unstructured hex mesh with target node selected for refinement (a), inserted templates in elements adjacent to node (b) and the final smoothed mesh (c).](image)

Refinement of element edges using the parallel sheet 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 7.9. Several edges
located at an abrupt change in geometry of the model have been refined. Figure 7.10 shows the same model meshed with only a quadrilateral surface mesh for comparison of the refinement schemes. In both figures, the refined regions are shown in dark gray and the transition zones are shown in white. Though some differences are present at the ends of the refinement region, the two refinement techniques produce very similar results.

Though refinement of nodes and edges using parallel sheet refinement appears to produce results that closely resemble those of the surface refinement technique described earlier, the inability to refine concave target regions and elements in unstructured meshes makes creating a matching hex refinement scheme impossible. Refinement of element faces using parallel sheet refinement is only done by refining the hexahedrons connected to the faces with a one-into-eight split, which this technique is not well suited to do for unstructured meshes.
Figure 7.9: Parallel sheet refinement of selected edges.
Figure 7.10: Refinement of a quadrilateral surface mesh.
8 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.

8.1 Templates

For use in the combination refinement technique, the parallel sheet templates shown in Figure 7.3 are modified by splitting the original element edges at one-third the length from the marked node to form those shown in Figure 8.1.

![Figure 8.1: Modified parallel hex sheet templates.](image)

8.2 Implementation

Figures 8.2 and 8.3 display how the two techniques are combined. Frame (a) of both Figure 8.2 and Figure 8.3 show the entities to be refined. The target regions are first reduced to the nodes that define them as shown in frame (b) of both figures. These nodes are flagged for refinement. The templates of Figure 8.1 are then inserted into the mesh as dictated by the flagged nodes to produce the mesh in frame (c) of both figures. This is essentially a single level of parallel sheet refinement. Only templates (b) and (c) of
Figure 8.1 were needed to perform this step of refinement for the target edges in Figure 8.2(c). All three modified parallel sheet templates were used to perform this step of refinement for the target face in Figure 8.3(c).

Following a single level of modified parallel sheet refinement, all eight nodes on the hexes adjacent to the target entities are flagged as depicted by the black nodes in frame (d) of Figures 8.2 and 8.3. Single sheet refinement then follows as described previously with the individual hex sheets to be refined defined by the target edges, shown with the heavy black line in Figure 8.2(e), or two adjacent edges of a target face, shown in Figure 8.3(e). 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 entities and the transition elements as seen in frame (f) of both figures.

For individual nodes, refinement is completed precisely as was described in Chapter 7 for parallel sheet refinement.
Figure 8.2: Combination refinement of two selected edges.
Figure 8.3: Combination refinement of a selected face.
8.1 Examples

Figure 8.4(a) shows the refinement of a single selected node. Each element surrounding the selected node, marked with black, is replaced with template (c) from Figure 8.1. Figure 8.4(b) shows the mesh after smoothing.

Figure 8.4: Single node refinement: refined mesh (a), smoothed mesh (b).

Figure 8.5 is an all-hex conforming mesh. Figure 8.6 shows combination refinement of several edges after smoothing. Figure 8.7 shows combination refinement of a group of faces after smoothing.
Figure 8.5: Combination refinement: Original mesh.
Figure 8.6: Combination refinement: Selected edges.

Figure 8.7: Combination refinement: Selected faces.
9 RESULTS AND DISCUSSION

The following sections discuss element quality and sizing issues that are shared by single sheet, parallel sheet and combination refinement methods.

9.1 Element Quality

The refinement process always introduces elements with quality lower than the original elements into the transition regions. The quality is degraded most in areas of overlapping transition templates. Figure 9.1 displays a common situation in single sheet refinement where two (b) templates from Figure 6.4 intersect the same transition element from two directions. This occurs when two closing twist planes, the dashed lines labeled 1 and 2 in Figure 9.1(a), intersect perpendicularly. The exploded view of the transition region in Figure 9.1(b) shows several lower quality elements.

![Figure 9.1: Overlapping transition zones in single sheet refinement (a) and exploded view of transition region (b).](image-url)
The quality of the elements can be degraded further by inserting transition elements into the poor quality elements of Figure 9.1. Figure 9.2 shows how this situation could occur. In Figure 9.2(a) a target refinement region is selected. A group of eight new elements is then selected for further refinement in Figure 9.2(b). Refinement of these elements results in overlapping the transition zones with those of the previous refinement to produce several sliver-like elements in Figure 9.2(c). Though the shapes of the elements in Figure 9.2(d) are improved through smoothing, the quality of several elements is still poor.

Figure 9.2: Transition zone overlapping.
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 [23] work best for quality improvement of refined meshes. The examples contained in this paper were smoothed with Mesquite’s mean ratio smoother within CUBIT [24]. Figures 9.3 through 9.6 compare the three refinement methods along a selected geometric boundary. The figures are all displayed after applying surface and volume smoothing. Table 9.1 gives the value of the “shape quality” metric [23] of the meshes in Figures 9.3 through 9.6. These metrics provide a standard for comparison of the meshes before and after refinement and before and after execution of smoothing. The shape quality metric equals one if the element is ideal and zero if the element is degenerate [23].

Figure 9.3 is the original all-hex, conforming mesh. The geometric curve that marks an abrupt change in geometry is a region where high gradients might occur during a stress analysis of the model under certain loading conditions. Figure 9.4 uses combination refinement to refine the curve. In Figure 9.5 the curve is refined using parallel sheet refinement. Figure 9.6 shows the curve 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 spread over the depth of two original elements. Refinement using the combination and parallel sheet methods 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.
Figure 9.3: Refinement Comparison: Original mesh.

Figure 9.4: Refinement Comparison: Single-parallel sheet combination refinement.
Figure 9.5: Refinement Comparison: Parallel sheet refinement.

Figure 9.6: Refinement Comparison: Single sheet refinement.
The quality of the individual techniques can be compared by observing the "Min." column of Table 9.1. This column displays the quality of the worst shaped element in the mesh. In most cases the poorest shaped elements are few in number when compared to the total number of elements as indicated by a significantly higher average shape for the whole mesh.

The minimum element shape metric for the original mesh in Figure 9.3 is 0.612. After combination refinement, Figure 9.4, the minimum is degraded by a factor of 2 to 0.306. However much is recovered through smoothing (new minimum element shape metric = 0.447). Smoothing does not improve the mesh refined with the parallel sheet method (Figure 9.5 minimum shape metric = 0.417 before smoothing and 0.416 after), however the quality of the mesh after refinement without smoothing is better than the other methods. With parallel sheet refinement, the intersection of transition regions is simpler, thus making the quality in the transition regions better than that produced with single sheet refinement. The quality of the mesh drops dramatically using single sheet refinement (Figure 9.6, minimum shape metric = 0.166). However, after smoothing, the quality is improved to a level that is acceptable in most analyses.

Table 9.1: Comparison of refined meshes in Figures 9.3 to 9.6 using the "shape quality" metric.

<table>
<thead>
<tr>
<th>Refinement Scheme</th>
<th>Figure</th>
<th>Num. Hexes</th>
<th>Ave.</th>
<th>Std. Dev.</th>
<th>Min.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Mesh</td>
<td>Figure 9.3</td>
<td>1116</td>
<td>0.844</td>
<td>0.070</td>
<td>0.612</td>
<td>0.934</td>
</tr>
<tr>
<td>Combination</td>
<td>Pre-smoothing</td>
<td>1962</td>
<td>0.721</td>
<td>0.208</td>
<td>0.306</td>
<td>0.952</td>
</tr>
<tr>
<td>Combination</td>
<td>Figure 9.4</td>
<td>1962</td>
<td>0.775</td>
<td>0.107</td>
<td>0.447</td>
<td>0.947</td>
</tr>
<tr>
<td>Parallel Sheet</td>
<td>Pre-smoothing</td>
<td>2135</td>
<td>0.711</td>
<td>0.174</td>
<td>0.417</td>
<td>0.954</td>
</tr>
<tr>
<td>Parallel Sheet</td>
<td>Figure 9.5</td>
<td>2135</td>
<td>0.742</td>
<td>0.128</td>
<td>0.416</td>
<td>0.955</td>
</tr>
<tr>
<td>Single Sheet</td>
<td>Pre-smoothing</td>
<td>5828</td>
<td>0.715</td>
<td>0.218</td>
<td>0.166</td>
<td>0.995</td>
</tr>
<tr>
<td>Single Sheet</td>
<td>Figure 9.6</td>
<td>5828</td>
<td>0.754</td>
<td>0.198</td>
<td>0.257</td>
<td>0.998</td>
</tr>
</tbody>
</table>

No matter which refinement technique is used, due to their lower quality, it is best to keep the transition regions out of the area of interest to avoid negative impacts on analysis accuracy.

Table 9.1 also compares the number of total hex elements in the mesh after refinement. Because it modifies the fewest number of elements, combination refinement
adds the lowest number of new elements to the mesh while single sheet refinement increases the total hex count significantly.

Tables 9.2 and 9.3 give the shape quality metrics of the meshes in Figures 6.7 to 6.11 and 8.5 to 8.7 that appeared previously.

Table 9.2: Comparison of single sheet refined meshes in Figures 6.8 to 6.11 using the “shape quality” metric.

<table>
<thead>
<tr>
<th>Target Region</th>
<th>Figure</th>
<th>Num. Hexes</th>
<th>Ave.</th>
<th>Std. Dev.</th>
<th>Min.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Mesh</td>
<td>Figure 6.8</td>
<td>1248</td>
<td>0.932</td>
<td>0.070</td>
<td>0.726</td>
<td>0.999</td>
</tr>
<tr>
<td>Selected Elements</td>
<td>Pre-smoothing</td>
<td>4448</td>
<td>0.789</td>
<td>0.128</td>
<td>0.294</td>
<td>0.999</td>
</tr>
<tr>
<td>Selected Elements</td>
<td>Figure 6.9</td>
<td>4448</td>
<td>0.818</td>
<td>0.184</td>
<td>0.312</td>
<td>0.999</td>
</tr>
<tr>
<td>Selected Curve</td>
<td>Pre-smoothing</td>
<td>2112</td>
<td>0.851</td>
<td>0.190</td>
<td>0.278</td>
<td>0.999</td>
</tr>
<tr>
<td>Selected Curve</td>
<td>Figure 6.10</td>
<td>2112</td>
<td>0.867</td>
<td>0.161</td>
<td>0.319</td>
<td>0.999</td>
</tr>
<tr>
<td>Selected Surface</td>
<td>Pre-smoothing</td>
<td>4212</td>
<td>0.851</td>
<td>0.205</td>
<td>0.207</td>
<td>0.999</td>
</tr>
<tr>
<td>Selected Surface</td>
<td>Figure 6.11</td>
<td>4212</td>
<td>0.863</td>
<td>0.180</td>
<td>0.266</td>
<td>0.999</td>
</tr>
</tbody>
</table>

Table 9.3: Comparison of combination refined meshes in Figures 8.5 to 8.7 using the “shape quality” metric.

<table>
<thead>
<tr>
<th>Target Region</th>
<th>Figure</th>
<th>Num. Hexes</th>
<th>Ave.</th>
<th>Std. Dev.</th>
<th>Min.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Mesh</td>
<td>Figure 8.5</td>
<td>3093</td>
<td>0.942</td>
<td>0.061</td>
<td>0.572</td>
<td>0.997</td>
</tr>
<tr>
<td>Selected Curves</td>
<td>Pre-smoothing</td>
<td>3333</td>
<td>0.919</td>
<td>0.113</td>
<td>0.354</td>
<td>0.997</td>
</tr>
<tr>
<td>Selected Curves</td>
<td>Figure 8.6</td>
<td>3333</td>
<td>0.925</td>
<td>0.088</td>
<td>0.535</td>
<td>0.995</td>
</tr>
<tr>
<td>Selected Faces</td>
<td>Pre-smoothing</td>
<td>3940</td>
<td>0.883</td>
<td>0.163</td>
<td>0.128</td>
<td>0.997</td>
</tr>
<tr>
<td>Selected Faces</td>
<td>Figure 8.7</td>
<td>3940</td>
<td>0.895</td>
<td>0.134</td>
<td>0.163</td>
<td>0.998</td>
</tr>
</tbody>
</table>

9.2 Size Control

The refinement scheme presented offers little control over the size of the refined elements. Because parallel sheet refinement, which was developed for the one-into-eight refinement of hex elements, has many limitations single sheet and combination refinement schemes are used. These schemes split the element edges within the target region into three, resulting in one-into-twenty-seven refinement of hex elements. Such refinement results in a very fine mesh inside the target region. Furthermore, the edge size of the refined elements are limited to $1/3^n$ times the size of the original element, where $n = 1, 2, 3...$ Multiple refinement iterations in a single target region cause the element size to
decrease exponentially. Several elements in Figure 9.7(a) are refined to produce the mesh in Figure 9.7(b). Several new elements are then refined again to produce the mesh in Figure 9.7(c). The edge sizes of the refined elements in Figure 9.7(c) are 1/9 the size of the original elements. Possibilities for overcoming size limitations are discussed briefly in the future research section of Chapter 11.

Figure 9.7: Original mesh (a), refinement of selected hexes (b and c).
10 EXAMPLES

The parameters that control the scale and scope of refinement within CUBIT were presented and described in Chapter 4. They included numsplit, size, bias, sizing function, depth and radius. For interactive refinement, the user must first select a target refinement region by specifying the geometric or mesh entities to refine. Then the number of times to iteratively refine the region is specified by choosing the numsplit, size with optional bias or sizing function commands. The default command is set at numsplit = 1, which is one complete refinement of the target region. Finally, the user may select either depth or radius to specify the spread of the target region. If neither depth nor radius is specified, a default depth of 1 is used for target regions consisting of geometric entities, which includes all the hex elements bordered by that entity in the refinement region. A default depth of zero is used for target regions consisting of mesh entities, which does not add any additional entities into the target region.

The refinement algorithm interprets the target region and the selected parameters to perform either single sheet refinement or combination refinement. If the target region contains any full hex elements, then only single sheet refinement is necessary. If the target region is composed of nodes, element edges or element faces, then combination refinement is necessary. If a depth of 1 or greater or a radius greater than zero is selected, then full hex elements will always be included in the target region and single sheet refinement will be performed regardless of the selected entities. If the target region consists of geometric entities and a depth of zero has been selected, only the nodes, element edges and element faces will be included in the refinement region and combination refinement will be performed.

The following figures present examples of twist plane insertion refinement based on these parameters. The original mesh in Figure 10.1 is all-hex and conforming. It was produced by automatically sweeping the left surface through the volume with mesh size
set to 1.0 units. This caused the edge lengths of all the elements to be approximately equal to 1.

In Figure 10.2, the geometric curve defining the top of the front surface was refined with the default numsplit equal to 1 and depth parameter equal to 3 to create the mesh.
Setting depth to three spreads the refinement into all elements within three elements from the selected curve.

Again the curve is refined to produce the mesh in Figure 10.3. For this example, the default numsplit is again selected but the radius parameter is set to 1.5. The radius parameter includes all hexes within the radial distance of 1.5 units from the selected curve. Each edge on the original mesh is approximately 1 unit long, so one would expect one and one-half layers of elements around the curve to be included in the target region. As seen in Figure 10.3, two layers of elements were added to the target refinement region. The elements that were only partially within the radial distance from the curve were completely included within the target region.

![Figure 10.3: Refinement examples: Selected curve with radius = 1.5.](image)

The mesh in Figure 10.4 was made by refining the upper left vertex of the front surface of the original mesh with the size parameter set to 0.125 and bias set to 1.3. When used together, size and bias set a sizing map for the entire mesh. The size at the selected vertex is 0.125 and gradually increases at a rate 1.3 times the number of element layers from the vertex. The algorithm then iteratively refines the elements with edge sizes larger than the specified size. At the elements near the vertex, a single refinement iteration reduces the element edge sizes to approximately 0.333. The elements are then
refined a second time to give edge lengths of approximately 0.111. At this point all elements in the region are beneath the 0.125 size limit and the refinement process terminates.

![Refinement examples: Selected vertex size = 0.125 and bias = 1.3.](image)

Figure 10.4: Refinement examples: Selected vertex size = 0.125 and bias = 1.3.

Though the size parameter gradually increases as the distance from the vertex increases, the changes in actual element size are abrupt. This is a result of the size control limitations discussed in Chapter 9, that the size or refined elements is equal to $1/3^n$ times the original element size, where $n = 1, 2, 3, ...$. The blocky shape of the refinement region is due to the convex-only restriction of the refinement technique.
The mesh in Figure 10.5 shows an example of refinement of a target vertex with a depth of 3 selected. The size of the refinement elements has been set to 0.125 and bias is left as zero. Because no bias has been selected, all the elements within the target region have been refined twice.

The upper front curve on the original mesh is again refined, this time with the numsplit parameter set to 2 to produce the mesh in Figure 10.6. Numsplit is the number of times to iterate the refinement over the target region. As the default, the depth is automatically set equal to 1 and all the elements along the curve are refined. The algorithm then selects the new elements along the curve and refines these a second time to produce the mesh seen in Figure 10.6. Setting the default depth equal to 1 prevents the transition zones from overlapping to create poor quality elements as discussed in Chapter 9.
Figure 10.6: Refinement examples: Selected curve with numsplit = 2.

Figure 10.7 shows an example of refinement similar to Figure 10.6 except a depth of zero has been selected to invoke combination refinement.

Figure 10.7: Refinement examples: Selected curve with depth = 0 and numsplit = 2.

Figure 10.8 is an example of a group of element faces selected for refinement with numsplit set to 2. The first iteration of refinement splits each edge of the target faces into three. The second iteration splits each new edge into three, dividing each original edge
into nine new edges. The original face is then split into 81 new faces creating a dramatic increase in mesh density.

Figure 10.8: Refinement examples: Selected faces with numsplit = 2.
11 CONCLUSION

11.1 Summary

A refinement technique based on the insertion of twist planes into the STC has been presented. This technique is divided into two distinct portions, single sheet operations and parallel sheet operations. In the former, individual refinement steps occur entirely within a single hex sheet. Single sheet refinement is effective for dividing single and groups of elements but is incapable of effectively refining nodes, element edges and element faces. Using parallel sheet refinement, two neighboring sheets are modified in each level of refinement. This method is limited by the intersecting and diverging nature of hex sheets in unstructured meshes but, when used in conjunction with single sheet operations, forms a combination method which is well suited for refinement of single nodes, element edges and element faces. Both methods use basic templates defined by the number of flagged nodes on the original element to replace the element.

The contributions of this thesis to hexahedral finite element mesh refinement are:

1. The ability to conformally refine all entities of an all hexahedral element mesh.
2. The simplification of template-based refinement into a general method.
3. The application of template insertion to multi-directional refinement.

11.2 Future Research

11.2.1 Parallel Sheet 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.

### 11.2.2 Size Control

The size constraints discussed in Chapter 9 could be overcome by controlling the number and location of twist planes inserted into the target region and then smoothing the mesh with a smoothing algorithm that adjusts the size of the elements according to a predetermined size metric. Figure 11.1(a) shows a mesh refined with single sheet refinement. The mesh in Figure 11.1(b) was creating by reversing the refinement in several of the refinement sheets of Figure 11.1(a). After smoothing, the size of the elements within the target region increased.

![Figure 11.1: Refined hexes using single sheet refinement (a), refinement by inserting fewer twist planes into the refinement region (b).](image)

### 11.2.3 Constraint Reduction

The constraints that limit parallel sheet refinement discussed in Chapter 7 could be lessened by controlling the number of refinement directions. Instead of forcing three
directions for each target element, in highly unstructured mesh regions, one or two
directions of refinement could effectively split the elements.

11.2.4 Expanded Combination Refinement

Combination refinement can be expanded further by first refining the target region,
shown in Figure 11.2(a), in one (or two) directions with parallel sheet refinement, shown
in Figure 11.2(b). Then the remaining directions can be refined with single sheet
refinement as in Figure 11.2(c). This method would produce refinement that is less dense
than the single sheet refinement method discussed in this work.

![Figure 11.2: Target refinement region (a), parallel sheet refinement (b), single sheet refinement (c).](image)

11.2.3 Mesh Adaptivity Tool

Sheet extraction [25] can be used to undo twist plane insertion refinement (as is seen
in Figure 11.1). Coupling the two techniques could form an adaptivity package, executed
manually or automatically, that could be used to locally modify finite element meshes
through both coarsening and refinement.
REFERENCES


