2007-07-13

A Selective Approach to Hexahedral Refinement of Unstructured Conformal Meshes

Michael Hubbard Parrish
Brigham Young University - Provo

A SELECTIVE APPROACH TO CONFORMAL REFINEMENT

OF UNSTRUCTURED HEXAHEDRAL MESHES

by

Michael H. Parrish

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 2007

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Michael H. Parrish

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date	Steven E. Benzley, Chair
Date	Richard J. Balling
Date	Steven J. Owen

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Michael H. Parrish 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.

Date

Steven E. Benzley
Chair, Graduate Committee

Accepted for the Department

E. James Nelson
Graduate Coordinator

Accepted for the College

Alan R. Parkinson
Dean, Ira A. Fulton College of Engineering and Technology

ABSTRACT

A SELECTIVE APPROACH TO CONFORMAL REFINEMENT

OF UNSTRUCTURED HEXAHEDRAL MESHES

Michael H. Parrish

Department of Civil and Environmental Engineering

Master of Science

Hexahedral refinement increases the density of an all-hexahedral mesh in a specified region, improving numerical accuracy. Previous research using solely sheet refinement theory made the implementation computationally expensive and unable to effectively handle multiply-connected transition elements and self-intersecting hexahedral sheets. The Selective Approach method is a new procedure that combines two diverse methodologies to create an efficient and robust algorithm able to handle the above stated problems. These two refinement methods are: 1) element by element refinement and 2) directional refinement. In element by element refinement, the three inherent directions of a hexahedron are refined in one step using one of seven templates. Because of its computational superiority over directional refinement, but its inability to handle multiply-connected transition elements, element by element refinement is used in all areas of the specified region except regions local to multiply-connected transition

elements. The directional refinement scheme refines the three inherent directions of a hexahedron separately on a hexahedron by hexahedron basis. This differs from sheet refinement which refines hexahedra using hexahedral sheets. Directional refinement is able to correctly handle multiply-connected transition elements. A ranking system and propagation scheme allow directional refinement to work within the confines of the Selective Approach Algorithm.

ACKNOWLEDGMENTS

The pages that follow represent the combined effort of many people. I wish to thank my graduate committee and especially my advisor Dr. Steven E. Benzley for his time, counsel, and funding of this research. I also wish to thank Matt Staten, Mike Borden and others at Sandia National Laboratories for their help and contributions in the methodology and implementation of this work. Finally, I would like to thank my family for without them I would not be where I am today.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

As computing power continues to increase, the finite element method has become an increasingly important tool for many scientists and engineers. An essential step in the finite element method involves meshing or subdividing the domain into a discrete number of elements. Mesh generation has therefore been the topic of much research. Tetrahedral or hexahedral elements are commonly used to model three dimensional problems. Tetrahedral elements have extremely robust modeling capabilities for any general shape while hexahedral elements provide more efficiency and accuracy in the computational process [1].

Within the realm of hexahedral mesh generation, mesh modification is an area of research that attempts to improve the accuracy of an analysis by locally modifying the mesh to more accurately model the physics of a problem. Hexahedral refinement modifies the mesh by increasing the element density in a localized region.

Several schemes have been developed for the refinement of hexahedral meshes. Methods using iterative octrees [2] have been proposed, however these methods result in non-conformal elements (see Appendix A) which cannot be accommodated by some solvers. Other techniques insert non-hexahedral elements that result in hybrid meshes or require uniform splitting to maintain a consistent element type [3]. Schneiders proposed an element by element refinement scheme [4] in connection with an octree-based mesh generator; however this technique is limited in that it is unable to handle multiply-connected transition elements (see Chapter 3). Schneiders later proposed a sheet refinement method [5] which produces a conformal mesh by pillowing layers in alternating i, j, and k directions but relies on a Cartesian initial octree mesh. Tchon et al. built upon Schneiders' sheet refinement in their 3D anisotropic refinement scheme by expanding the refinement capabilities to unstructured meshes [6][7] however this scheme still has poor scalability inherent in all sheet refinement schemes. Harris et al. further expanded upon Schneiders' and Tchon's work by using templates (see Appendix E) instead of pillowing to refine the mesh and included capabilities to refine element nodes, element edges, and element faces [8]. While the refinement scheme introduced by Harris is robust in many aspects, it is limited by self-intersecting hexahedral sheets (see Chapter 4), multiply-connected transition elements, and poor scalability. The refinement process developed in this paper combines the element by element method proposed by Schneiders and the sheet refinement method proposed by Harris to create a method that overcomes the limitations of using either method alone.

2 Background

A hexahedron, the finite element of interest in this paper, has a dual representation defined by the intersection of three sheets called twist planes [9][10]. The direction normal to each sheet is a unique and inherent direction within a hexahedron. Figure 2-1 shows a hexahedron with its three dual twist planes. Each refinement direction is indicated with an arrow normal to each plane.

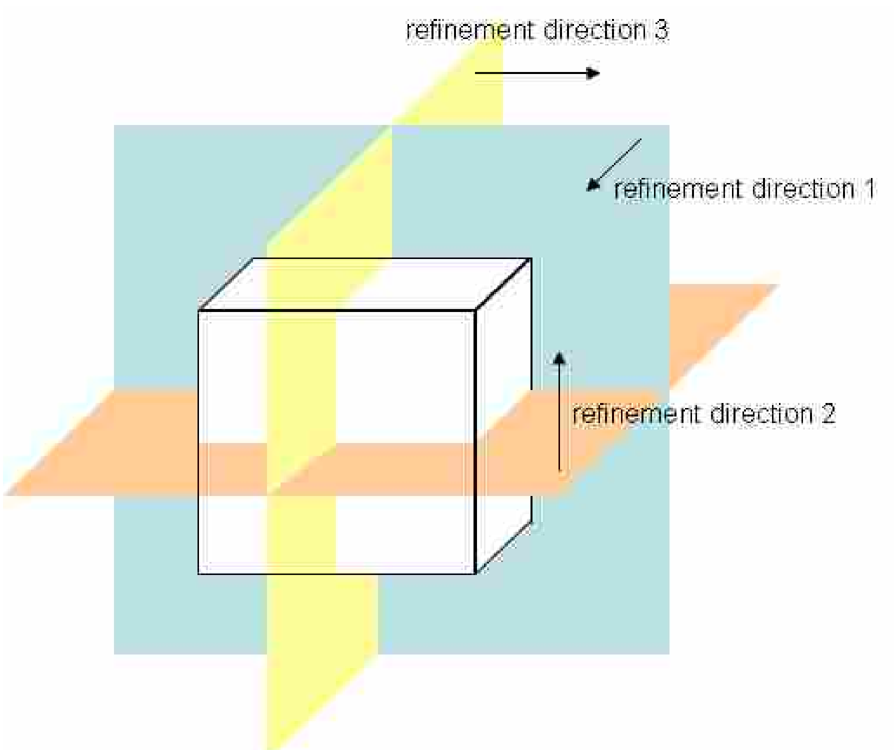


Figure 2-1: A hexahedron with its twist planes - arrows normal to twist planes represent directions of refinement

Element by element refinement replaces a single hexahedron with a predefined group of conformal elements effectively refining all three directions of the hexahedron at the same time. As such a non conformal mesh is temporarily created until all templates have been inserted. Only one template is applied to any initial element thus increasing the efficiency of the refinement process. Figure 2-2 shows how a mesh is refined using element by element refinement.

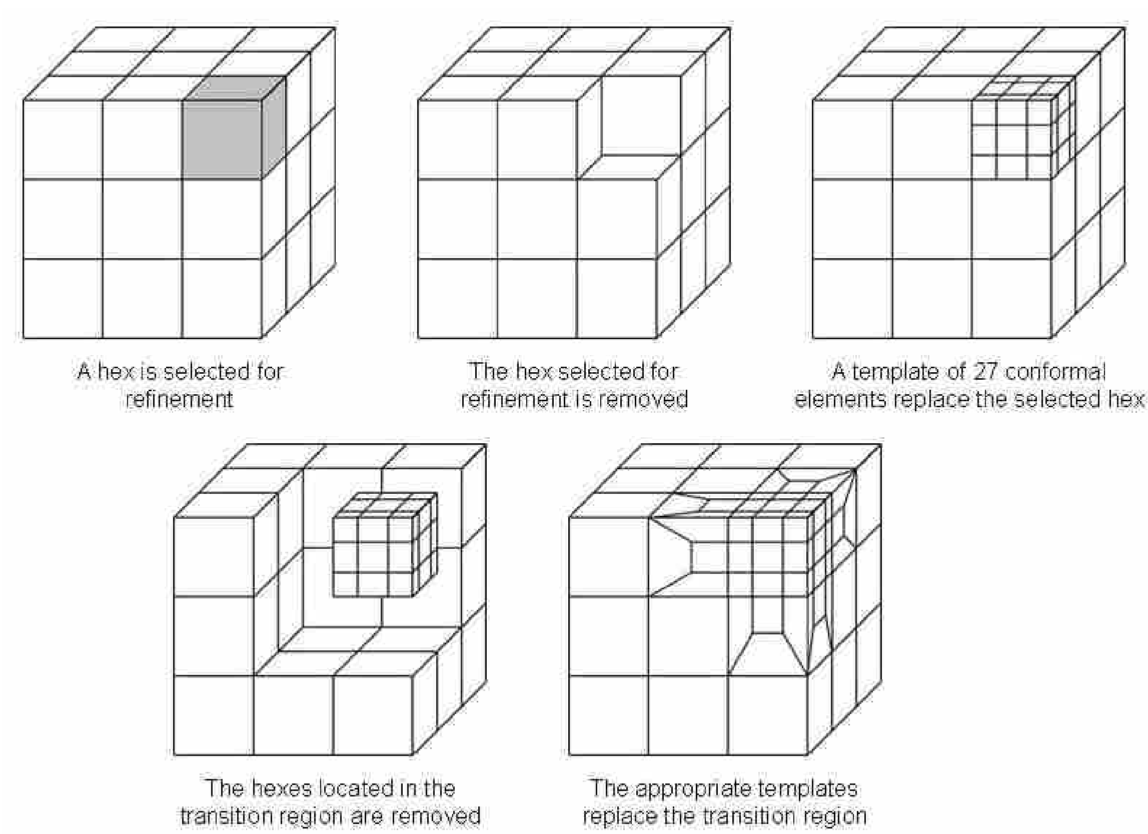


Figure 2-2: Element by element refinement

The sheet refinement method refines a hexahedron one direction at a time. The refinement region is processed in hexahedral sheets allowing unstructured meshes to remain conformal throughout the entire process. Since conformity is maintained, sheet refinement inherently produces a conformal mesh. Figure 2-3 shows how a mesh is refined using sheet refinement.

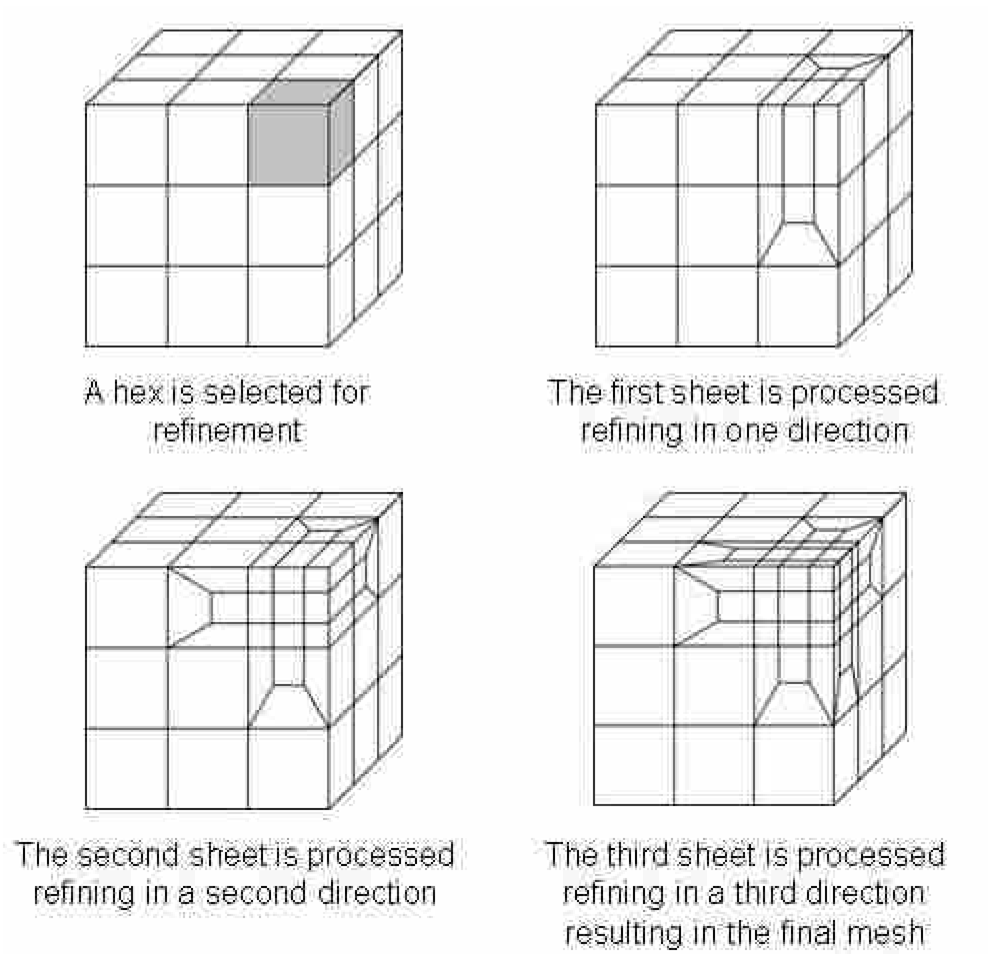


Figure 2-3: Sheet refinement

3 Limitations of Element by Element Refinement

Element by element refinement is limited by its inability to produce a conformal mesh where multiply-connected transition elements are present. In hexahedral refinement, a multiply-connected transition element refers to any hexahedral element that is not selected for refinement but shares more than one adjacent face with hexahedra that are selected for refinement (see Figure 3-1). This limitation stems largely from missing or unidentified templates. These templates are often unknown or cannot be created with reasonable quality thus limiting the effectiveness of the element by element refinement scheme.

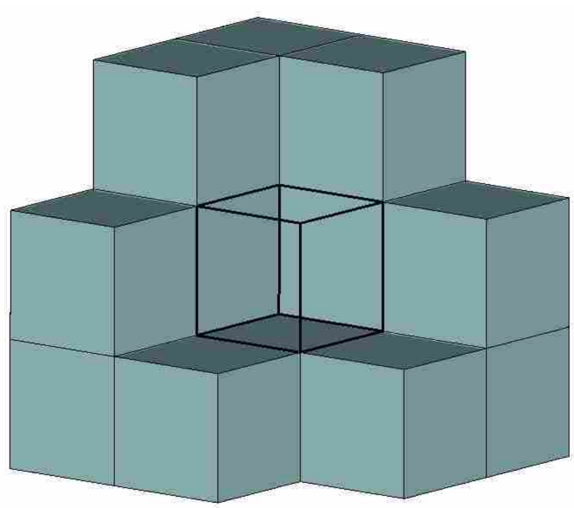


Figure 3-1: Example of multiply-connected transition element- hexahedron outlined in black is a multiply-connected transition element and shaded elements are selected for refinement

Figure 3-2 is an example where element by element refinement would produce a non-conformal mesh. Templates are successfully applied to the region selected for refinement and most of the transition elements. However, an adequate solution for the multiply-connected transition elements does not exist. Thus, the resulting mesh is non-conformal. A solution for this particular example has been proposed [11], however it produces too many elements and results in a low mesh quality.

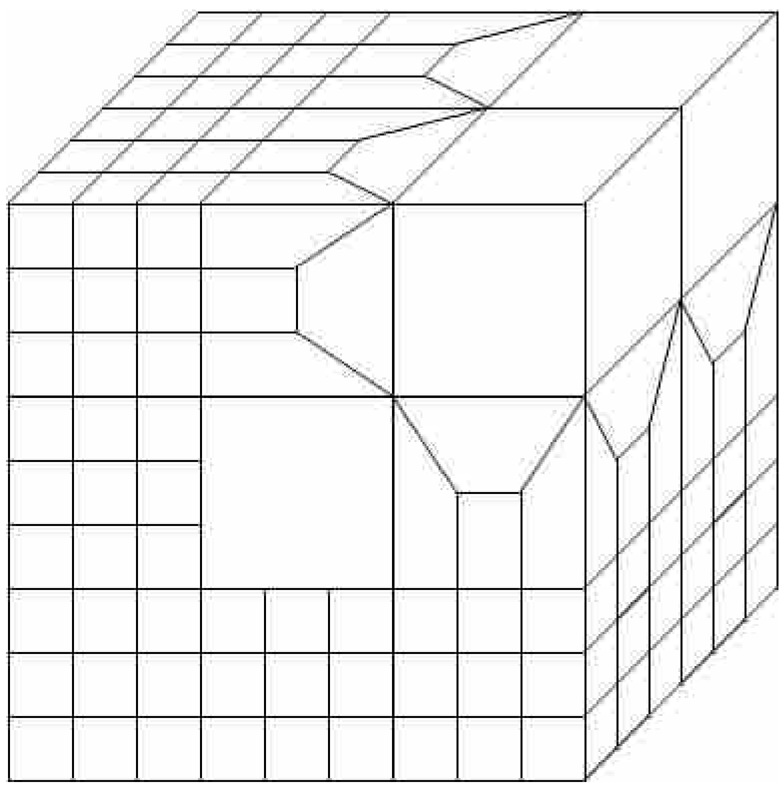


Figure 3-2: Example of non-conformal mesh where the refinement region contains multiply-connected transition elements

4 Limitations of Sheet Refinement

While sheet refinement is robust in its capabilities, it has three serious limitations. These limitations are: 1) the inability to effectively treat self-intersecting hexahedral sheets, 2) the inefficiency in refining multiply-connected transition elements, and 3) scalability.

4.1 Self-Intersecting Hexahedral Sheets

For conformal, all-hexahedral meshes, a hexahedral sheet must either initiate at a boundary and terminate at a boundary or form a closed surface. Sometimes meshing algorithms will create self-intersecting hexahedral sheets as shown in Figure 4-1. A self-intersecting hexahedral sheet is defined as any hexahedral sheet that passes through the same stack of elements multiple times (i.e. any dual twist plane that intersects itself). Hexahedra at the intersection of a self-intersecting hexahedral sheet must be handled as a special case because they need to be processed more than once. Recognizing all the cases where a sheet intersects with itself is a difficult and error prone procedure.

4.2 Multiply-Connected Transition Elements

Sheet refinement is able to produce a conformal mesh where multiply-connected transition elements are present however early implementations dealt with these transition elements inefficiently. Initially, hexahedra were added to the region until all multiply-connected transition elements were removed. While this produces a conformal mesh, it leads to excessive refinement. Excessive refinement increases the computational load for both mesh generation and analysis. Templates were later proposed to handle multiply-connected transition elements[12] but these templates were never implemented into any sheet refinement scheme.

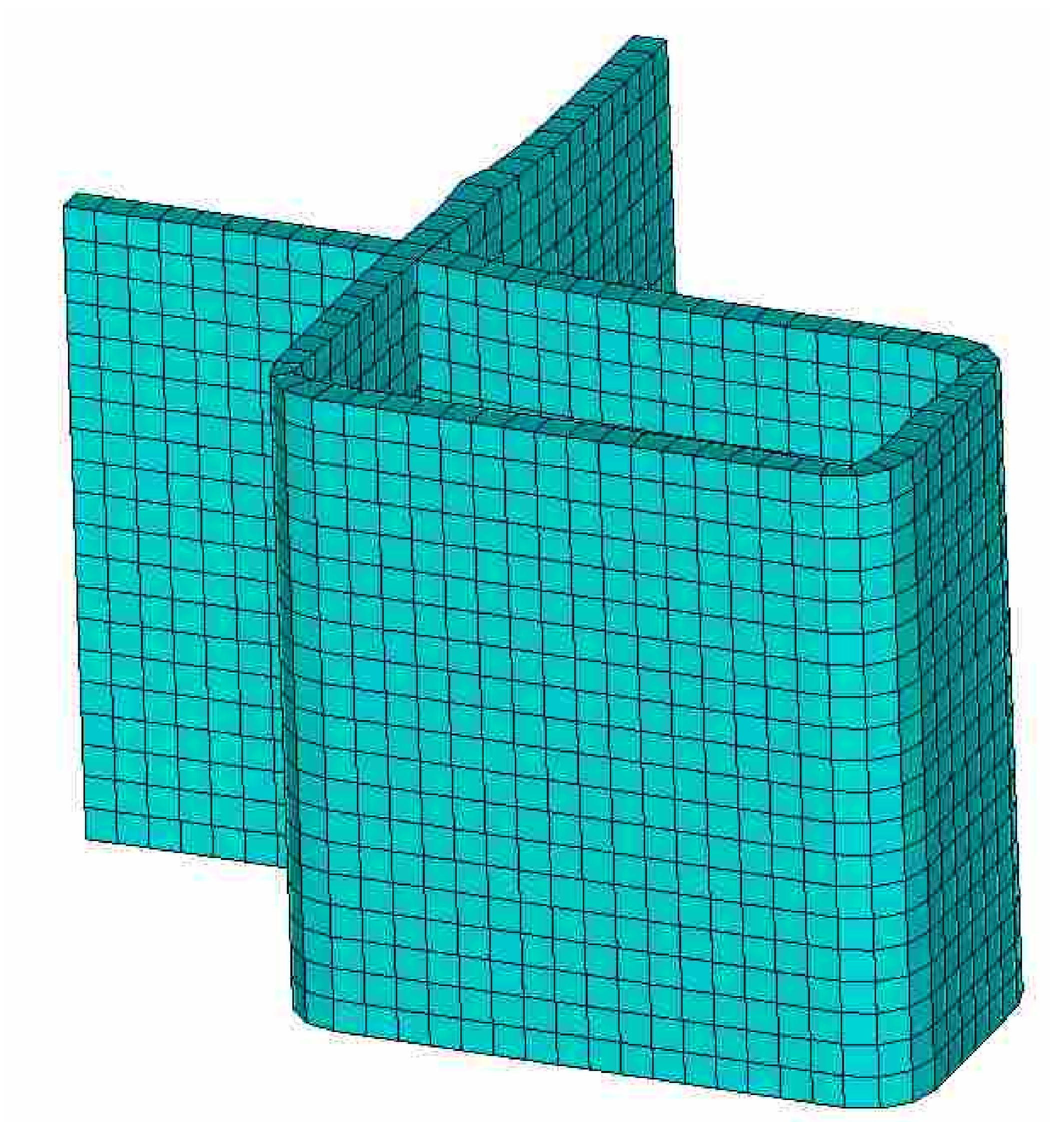


Figure 4-1: Example of self-intersecting hexahedral sheet

4.3 Scalability

Empirical studies show that the time requirement of sheet refinement grows exponentially as the number of initial elements increases. In Harris' implementation, a major contributor to this problem is the process of creating and deleting intermediate hexahedra (see Figure 4-2).

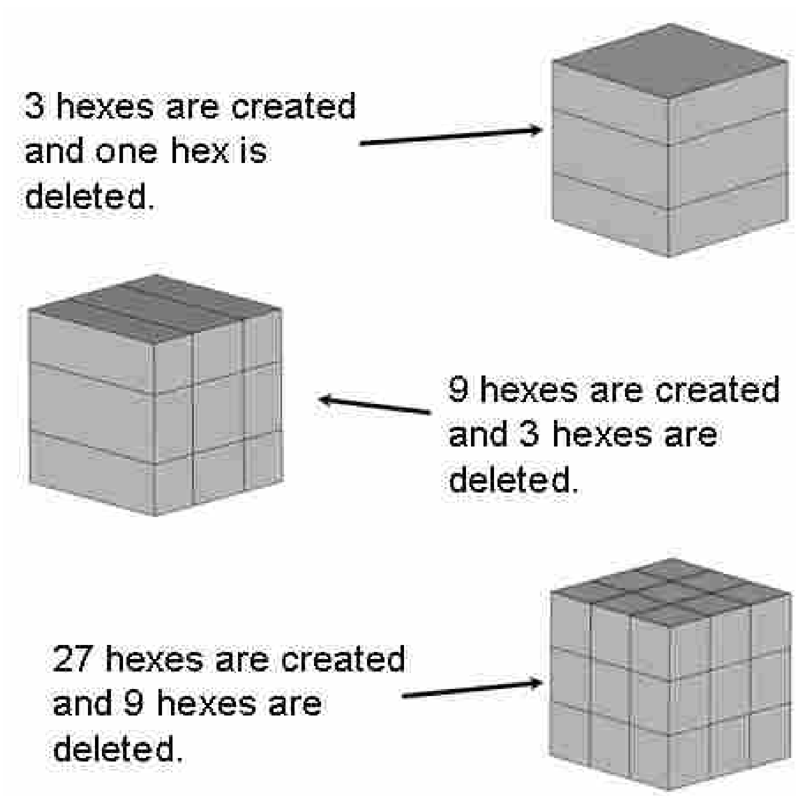


Figure 4-2: Harris' sheet refinement process

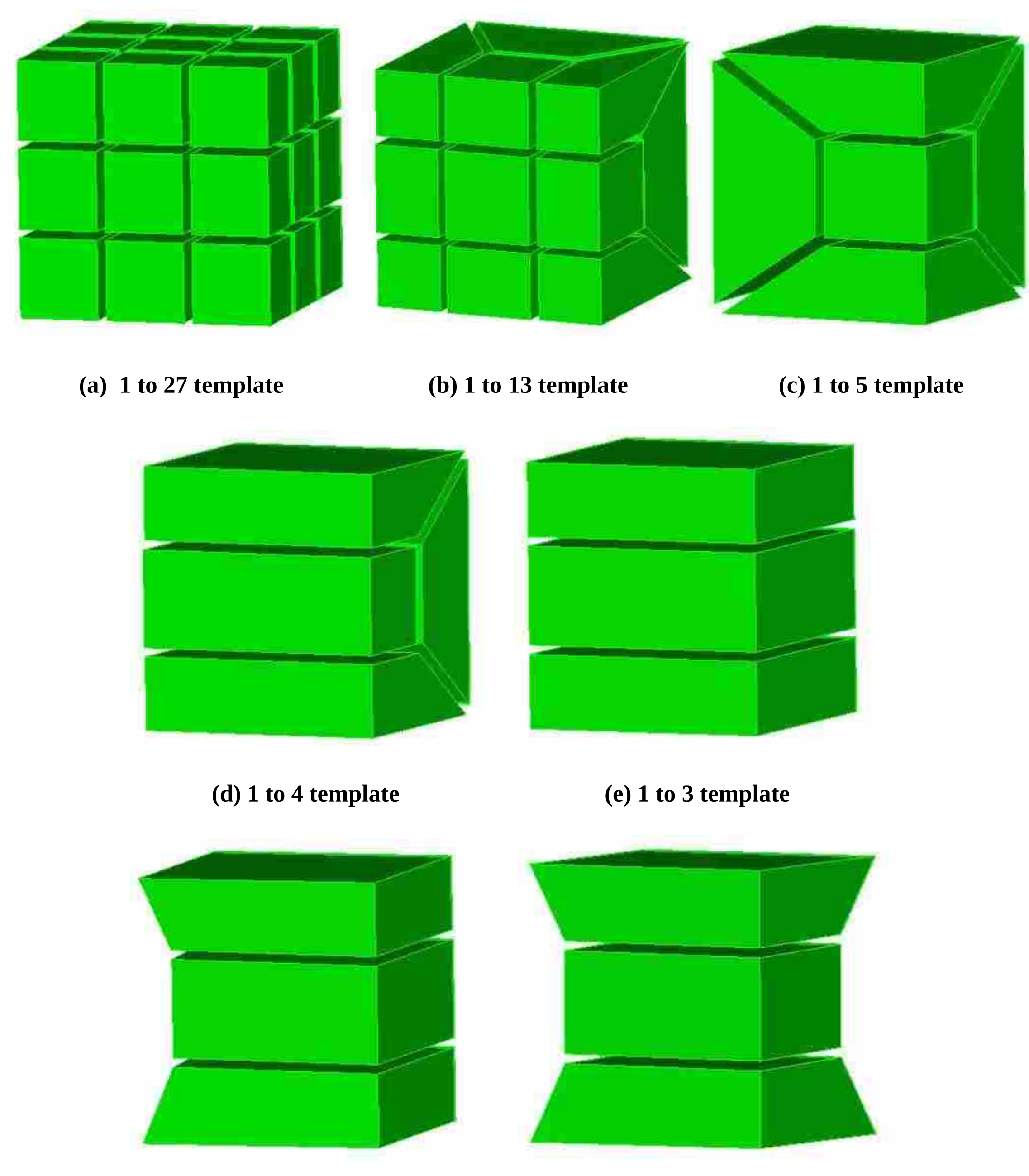
The process occurs in the following manner. The first sheet is processed, deleting the original hexahedron and creating three intermediate hexahedra. The second sheet is then processed, deleting the three intermediate hexahedra created by the first sheet and creating nine new intermediate hexahedra. Finally, the third sheet is processed, deleting the nine intermediate hexahedra created by the second sheet and creating the final 27 hexahedra. In total, 13 hexahedra are deleted and 39 hexahedra are created to obtain the desired refinement. Also, each creation and deletion requires a data base query further increasing the computational time.

5 A Selective Approach

The Selective Approach Algorithm is a new robust refinement scheme. This procedure (as its name suggests) automatically selects the more appropriate of two different refinement schemes for each hexahedron within a target region. A target region is defined as the elements selected for refinement and the transition elements connecting elements selected for refinement and the coarse mesh. The two refinement schemes used in the Selective Approach Algorithm are element by element (see Section 5.2) and directional (see Section 5.3) refinement. The combination of these two methods allows the Selective Approach Algorithm to overcome the limitations of both element by element and sheet refinement discussed previously.

5.1 Templates

Seven templates [4][12][13] are used within the Selective Approach Algorithm (see Figure 5-1). Both element by element refinement and directional refinement use templates. The 1 to 27 template and the 1 to 13 template are only used in the element by element refinement scheme while the other five templates are used in both element by element and directional refinement. Figure 5-1(f) and Figure 5-1(g) are the templates required to handle any multiply-connected transition element. Figure 5-2 explains how the 1 to 3 template with 1 adjustment is constructed. The 1 to 3 template with 2 adjustments is constructed in a similar fashion.



(a) 1 to 27 template **(b) 1 to 13 template** **(c) 1 to 5 template**

(d) 1 to 4 template **(e) 1 to 3 template**

(f) 1 to 3 template with one adjustment **(g) 1 to 3 template with two adjustments**

Figure 5-1: Templates used in the Selective Approach Algorithm

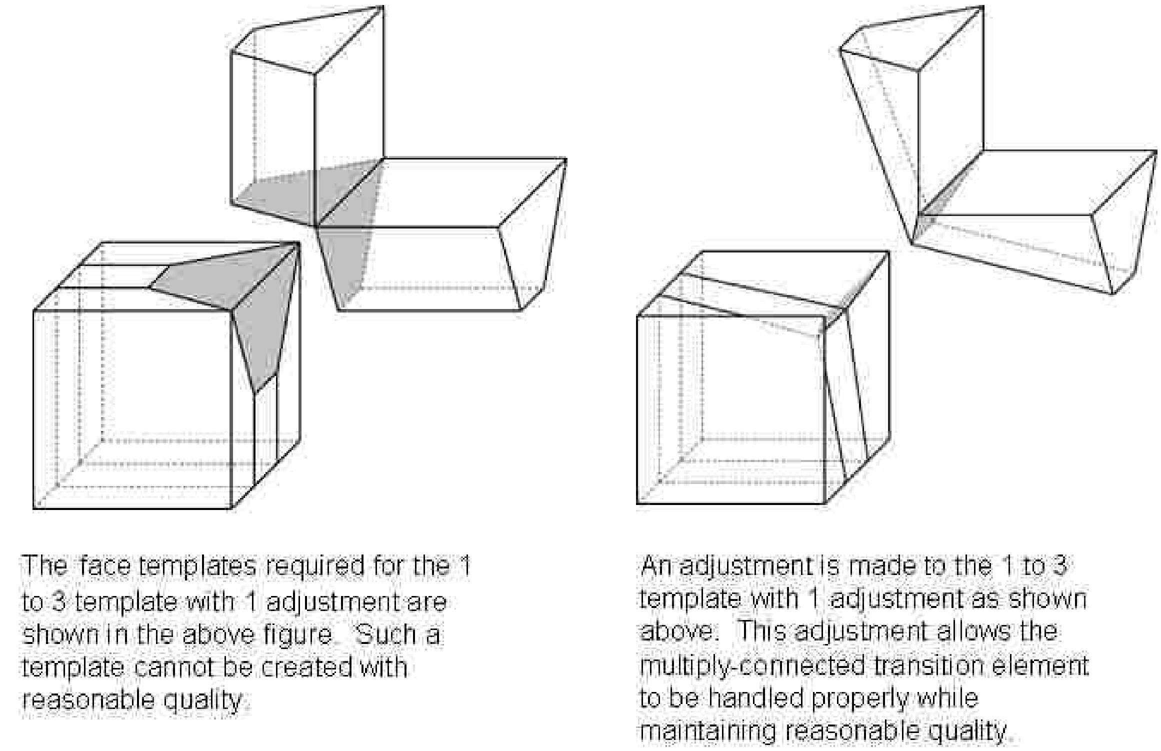


Figure 5-2: Adjustment to handle multiply-connected transition elements

5.2 Element by Element Refinement

The general process of performing element by element refinement was discussed in Chapter 2. Here element by element refinement is discussed in connection with the Selective Approach Algorithm. As stated previously, the element by element refinement method refines all three directions of a hexahedron in one step. A single hexahedron is deleted and the final group of elements is created using one of the seven templates described previously. Since no intermediate hexahedra are created or deleted, the computational efficiency of element by element refinement is far superior to that of sheet refinement. The limiting factor then, of the element by element refinement method is its inability to handle multiply-connected transition elements. Therefore, the Selective

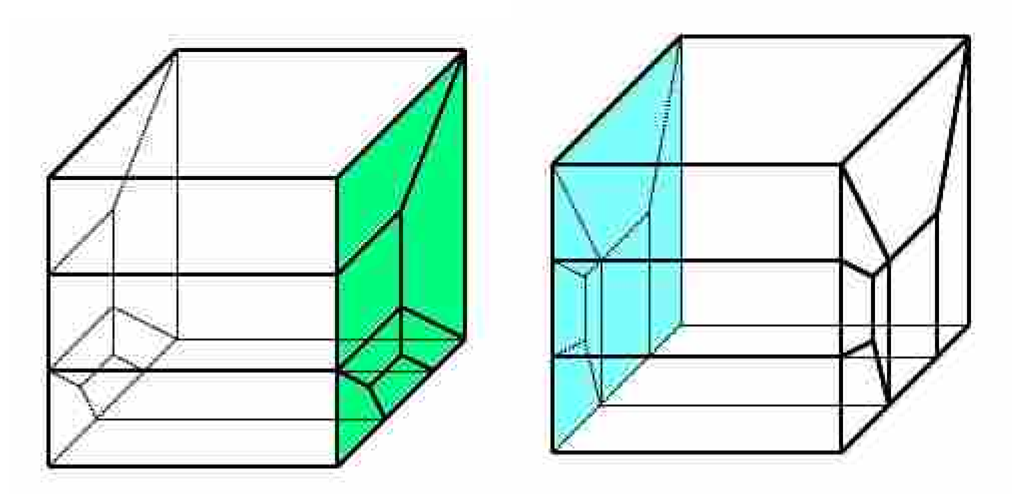
Approach Algorithm uses element by element refinement in all areas of the target region except areas local to multiply-connected transition elements.

5.3 Directional Refinement

Like sheet refinement, the directional refinement scheme refines each inherent direction of a hexahedron separately; however hexahedra are processed individually like element by element refinement. A ranking system and propagation scheme are new techniques used in directional refinement and will be discussed hereafter. While directional refinement requires more computational effort, it is able to produce a conformal mesh in regions local to multiply-connected transition elements. Directional refinement is therefore used in areas of the target region that contain multiply-connected transition elements.

5.3.1 The Conformity Problem and Ranking System

Conformity is a significant problem for the directional refinement scheme when hexahedra are processed element by element. An example of the conformity problem is shown in Figure 5-3 with two hexahedra that share a single face. The common face for both hexahedra is shaded in the figure. These two hexahedra share two common directions. These directions must be refined in the same order in both hexahedra, otherwise a non conformal mesh will be created. In Figure 5-3, both hexahedra contain valid refinement schemes yet the shared face is not conformable. This problem could potentially occur often since each hexahedron is refined independently of its neighbors. A method is therefore required so that refinement directions in adjacent hexahedra are refined in the same order.



(a) **(b)**

Figure 5-3: Conformity issues

To solve the conformity problem, the functionality of dual twist planes is used. The direction normal to each twist plane in this refinement scheme represent unique directions of refinement. In the Selective Approach method, connected elements receiving directional refinement are grouped together. Typically there is a single group for region containing multiply-connected transition elements. Each group is then processed separately by taking an initial arbitrary edge and giving it a rank of 1. All opposite edges of adjacent faces are located for the selected edge. If these new edges need to be directionally refined, they are given the same rank and become selected edges themselves. The rank propagates to all applicable edges intersecting and normal to the twist plane defined by the initial edge. The process repeats itself as another unranked edge is arbitrarily selected and given a rank of 2. The ranking scheme is finished when all applicable edges of the entire refinement region are ranked. The ranking system is described graphically in Figure 5-4. Refinement then occurs on an element by element basis starting in the direction with the lowest rank and continuing in ranked order until the hexahedron is completely refined and the algorithm moves onto the next hexahedron.

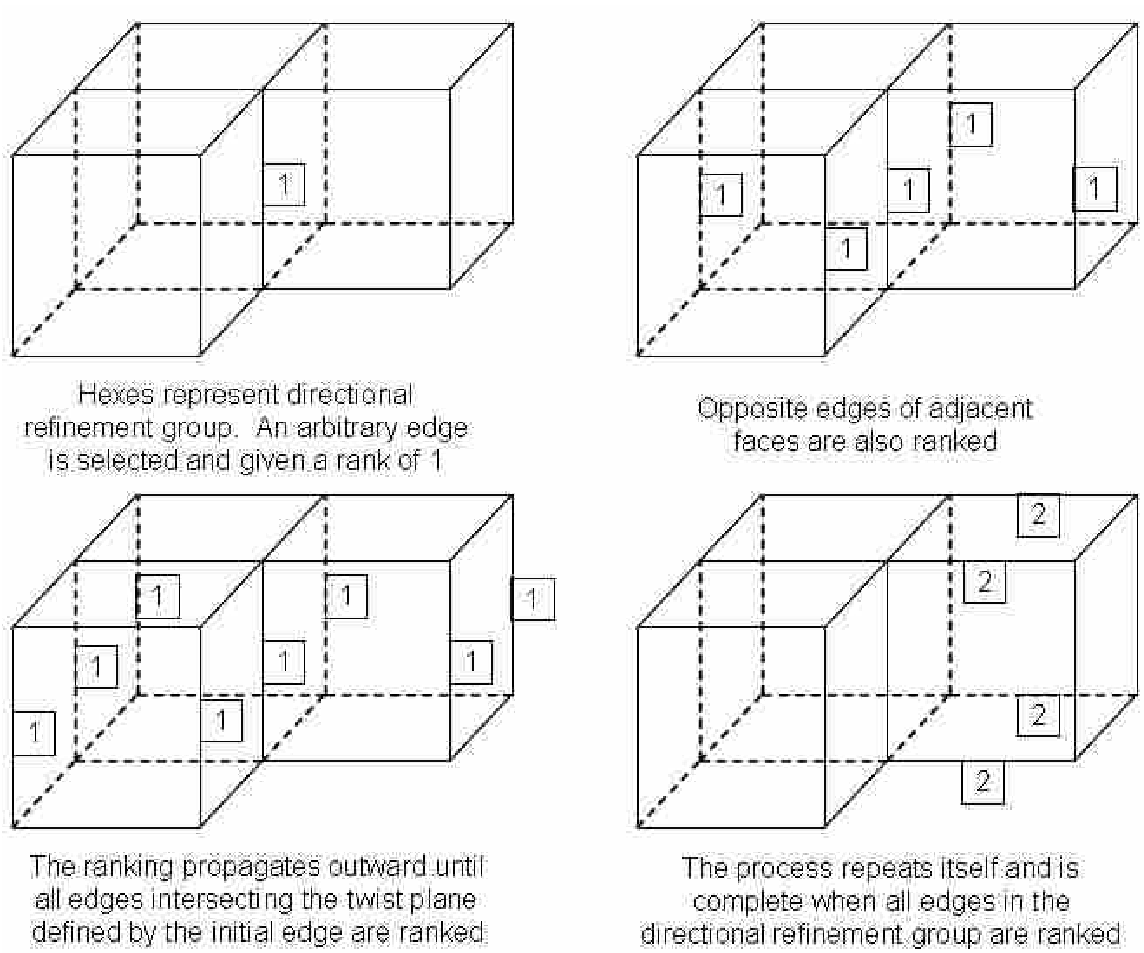


Figure 5-4: Ranking system

5.3.2 Propagation Scheme

After a hexahedron is refined in one direction using the directional refinement scheme, new edges exist that may need to be split in order to maintain element quality in the transition region. Only new edges perpendicular to the direction of refinement are considered in the propagation scheme. Figure 5-5 graphically shows how the propagation scheme works with a specific example.

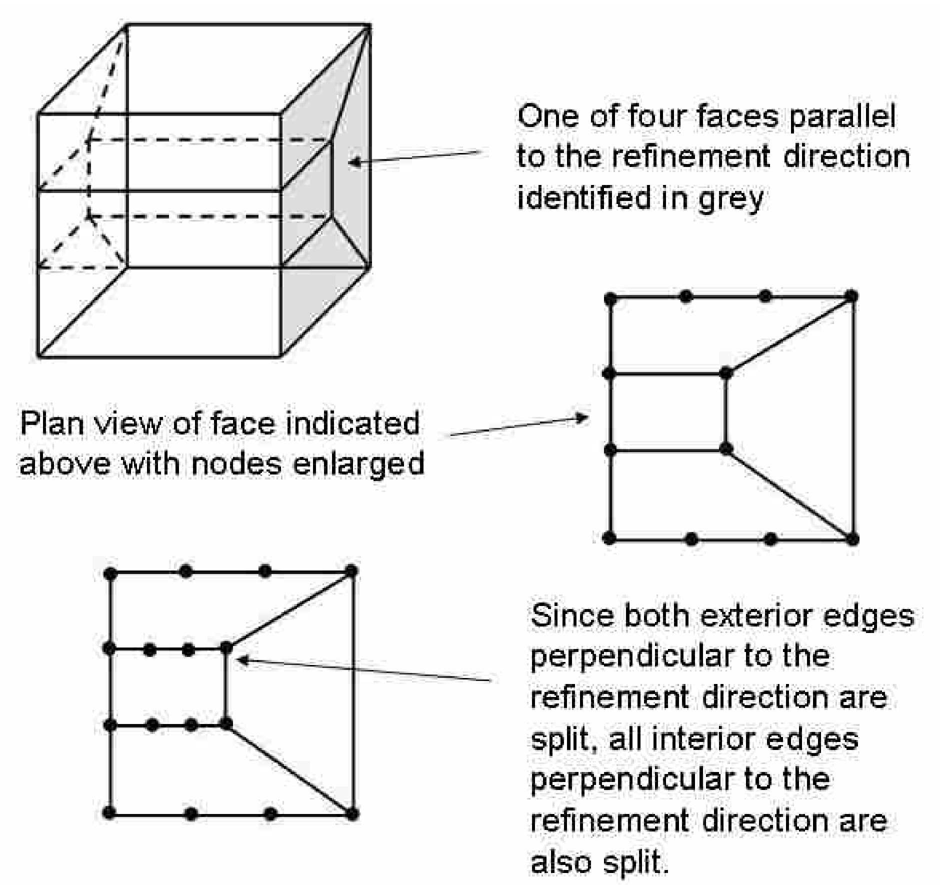


Figure 5-5: Propagation scheme

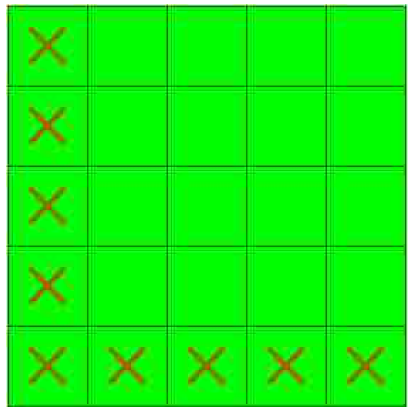
5.4 Algorithm

An outline of the Selective Approach Algorithm is given in Table 5-1. Figure 5-6 demonstrates the algorithm's logic with a specific example. The Selective Approach Algorithm starts by applying the 1 to 27 template to the elements selected for refinement (see Figure 5-6(b)). The transition hexahedra are all that remain after this step. Because element by element refinement is more efficient, it is applied first (see Figure 5-6(c)). The remaining hexahedra are then ranked as shown in Table 5-1 step 12. Finally, the remaining hexahedra are refined directionally in order of increasing rank. The propagation scheme is applied to each hexahedron during the directional refinement process (see Figure 5-6(d)).

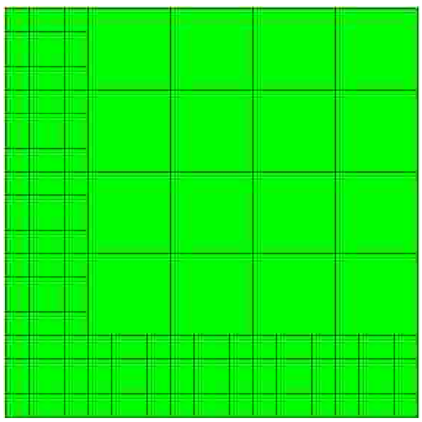
Table 5-1: The Selective Approach Algorithm

The Selective Approach Algorithm

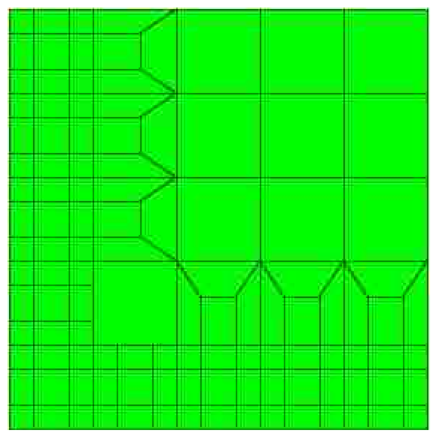
```
 1 :  loop target hexes
 2 :        apply 1 to 27 template to elements selected for refinement
 3 :  end loop
 4 :  loop transition hexes
 5 :        if template applies then
 6 :              refine hex using template
 7 :        else
 8 :              add to directional hex list
 9 :        end if
10 :  end loop
11 :  loop directional hex list
12 :        apply ranking system
13 :  end loop
14 :  loop directional hex list
15 :        loop refinement directions in order of increasing rank
16 :              apply template
17 :              apply propagation scheme
18 :        end loop
19 :  end loop
```



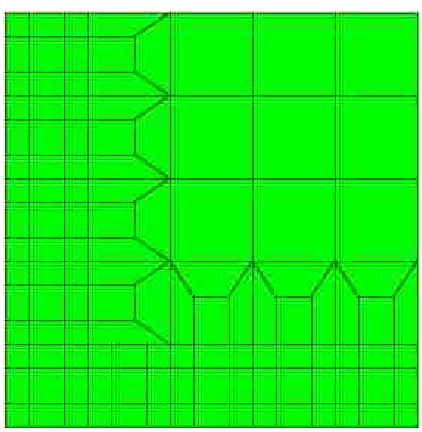
(a) Original mesh where left and bottom hexahedra selected for refinement



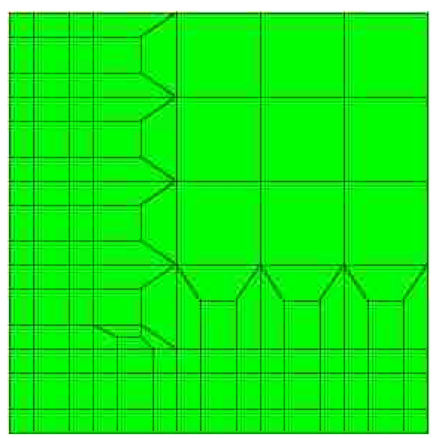
(b) 1 to 27 template applied to elements selected for refinement



(c) Element by element refinement is transition elements



(d) Element is refined in one direction followed by propagation scheme



(e) Element is refined in final direction resulting in the final mesh

Figure 5-6: Example of algorithm

6 Results and Example

The Selective Approach Algorithm solves the sheet refinement limitations of self-intersecting hexahedral sheets, inefficiently handled multiply-connected transition elements, and poor scalability. The following section considers the aforementioned limitations individually and discusses how the Selective Approach method eliminates them. Following this discussion, an example will be considered showing the robustness of this algorithm.

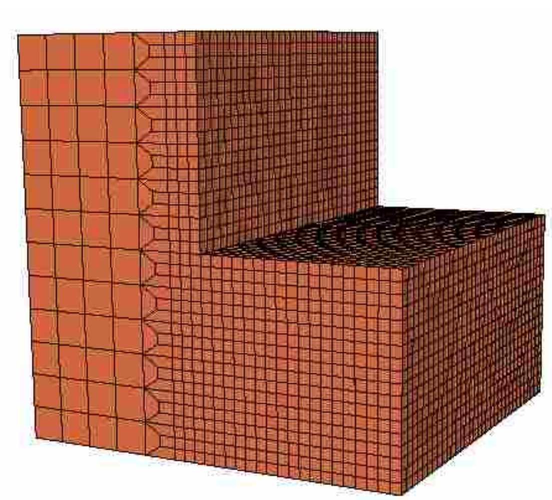
6.1 Self-Intersecting Hexahedral Sheets

The Selective Approach Algorithm automatically solves the limitation of self-intersecting hexahedral sheets because both element by element and directional refinement process the target region on a hexahedron by hexahedron basis.

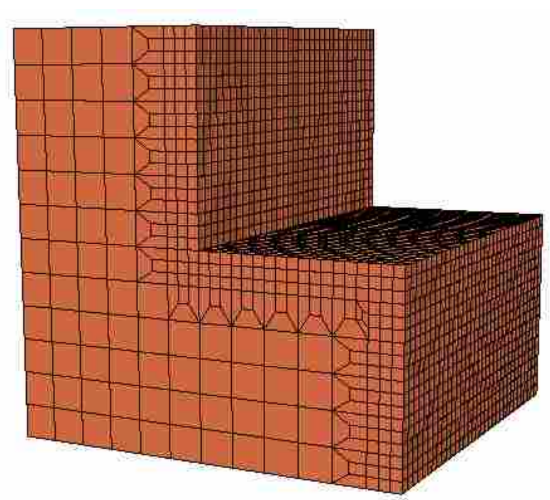
6.2 Multiply-Connected Transition Elements

To illustrate the new capabilities of the Selective Approach Algorithm when considering multiply-connected transition elements, a simple example problem is presented here. The Selective Approach Algorithm is compared with the sheet refinement scheme implemented by Harris.

The problem involves refining the surfaces composing the right boundary of the model. Figure 6-1(a) shows the model refined using the sheet refinement scheme implemented by Harris and Figure 6-1(b) shows the brick refined using the Selective Approach Algorithm. While sheet refinement could perform the refinement in a similar fashion to the Selective Approach Algorithm, the adjustment templates were never implemented. The sheet refinement scheme refined the entire bottom right section of the model in an attempt to remove the multiply-connected transition elements. Excessive refinement is not a problem with the Selective Approach method. The newly implemented adjustment templates eliminate the need to add hexahedra to the target region.



(a) Sheet refinement



(b) The Selective Approach Algorithm

Figure 6-1: Simple model where surfaces composing right boundary are refined

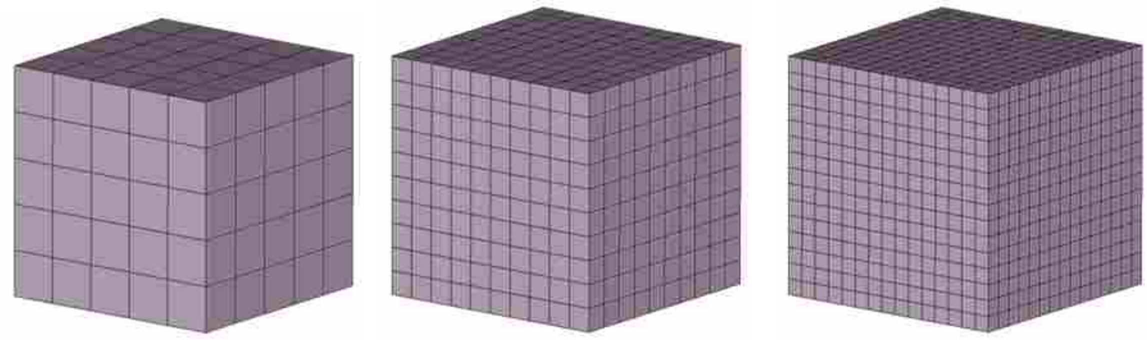
Values for the number of elements, time for both methods, and element quality are given in Table 6-1. For this example, the Selective Approach method is far superior in both element count and time required to perform the refinement. The Selective Approach Algorithm produced half as many elements and the time requirement was lower as well partially because fewer hexahedra were refined. Solving the mesh using the Selective Approach method would also require less time thus lowering the overall time required for a full analysis. The final minimum quality produced by both refinement schemes is the same and adequate for an accurate analysis.

Table 6-1: Numerical results of refining the surfaces composing the right boundary of the model

Measurement	Sheet Refinement	Selective Approach
Initial Element Count	1188	1188
Final Element Count	16500	8712
Time (sec)	5.359	0.859
Initial Minimum Quality	1.0	1.0
Final Minimum Quality	0.3143	0.3143

6.3 Scalability

To compare the scalability of the Selective Approach Algorithm to sheet refinement, a simple meshed brick was again used. The number of elements before refinement was increased incrementally by increasing the interval count of the brick as shown in Figure 6-2. Each meshed brick was completely refined and the required time recorded. Again, Harris' sheet refinement scheme was used for comparison in the analysis. The results are shown in Figure 6-3 and Figure 6-4. Figure 6-4 graphically shows the same data as Figure 6-3 however the y-axis has been reduced from 90000 seconds to 500 seconds to accurately portray the scalability of the Selective Approach Algorithm.



(a) Interval 5 **(b) Interval 10** **(c) Interval 15**

Figure 6-2: Interval determines element count

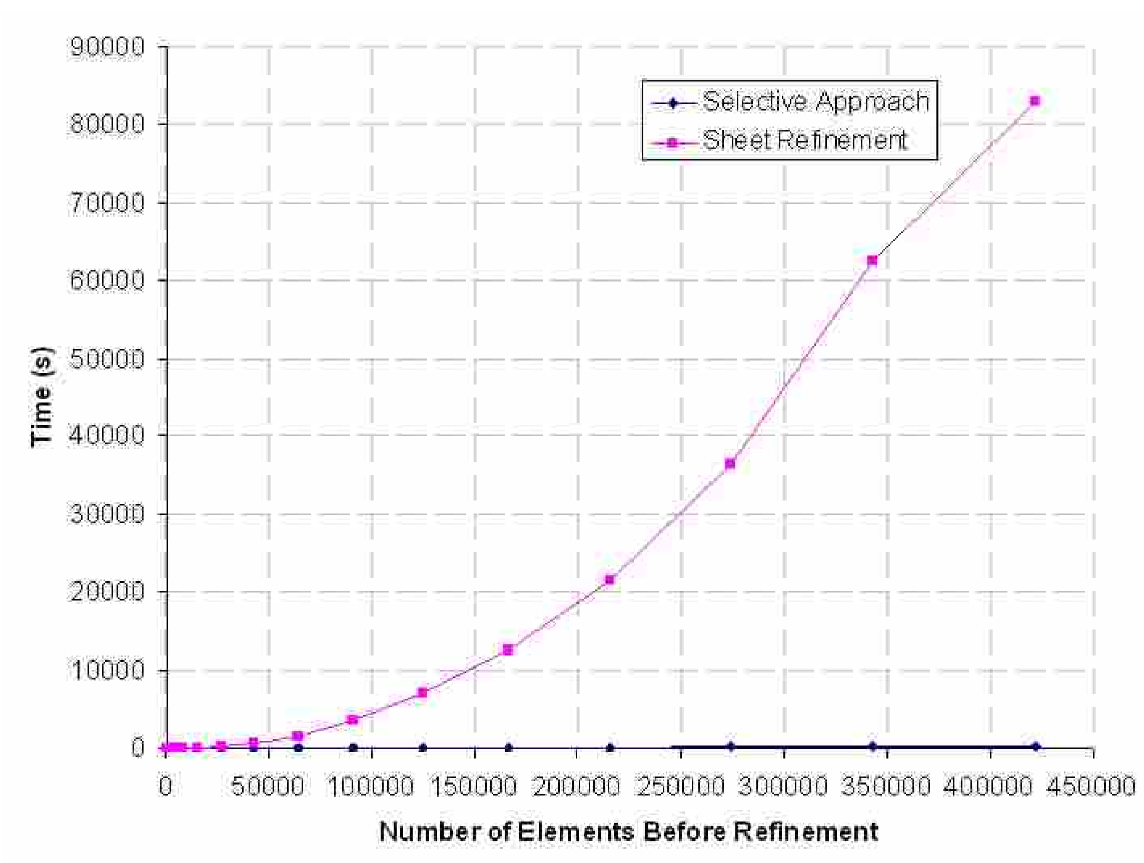


Figure 6-3: Comparison of scalability between sheet refinement and the Selective Approach Algorithm

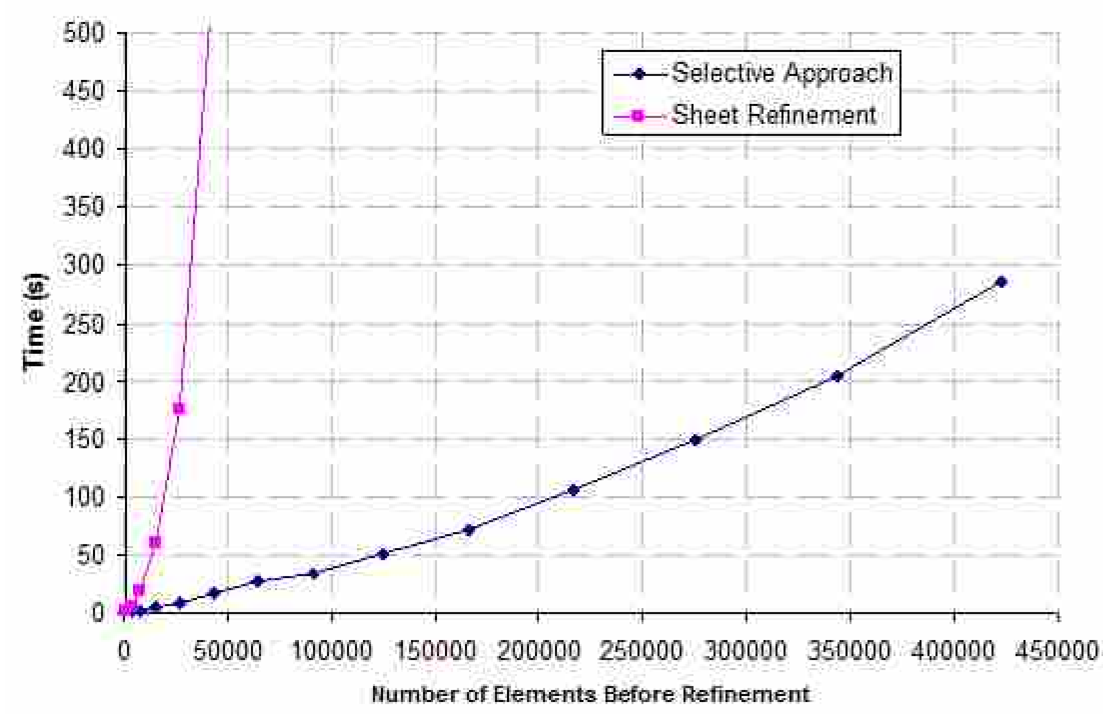


Figure 6-4: Comparison of scalability between sheet refinement and the Selective Approach Algorithm (y-axis reduced)

Arguably the greatest advantage of the Selective Approach method over sheet refinement is scalability. Figure 6-3 decisively shows the exponential increase in time for sheet refinement as the number of elements before refinement is increased. The scalability of the Selective Approach Algorithm is nearly linear in comparison (see Figure 6-4). The excellent scalability displayed in the Selective Approach Algorithm results from using element by element refinement as the primary refinement scheme.

It should be noted that in the above example, no elements required directional refinement within the Selective Approach Algorithm. A second scalability test was performed where the number of elements of a simple brick was increased incrementally by increasing the interval count as before. However, only elements within a constant radial distance from the top front vertex of the brick were refined instead of the entire brick (see Figure 6-5). This target region required directional refinement to be used in the refinement process. Using directional refinement will increase the overall computational time of the Selective Approach Algorithm. Figure 6-6 shows the results of the second scalability test where directional refinement is used. Figure 6-7 shows the same data with the y-axis reduced in order to determine the scalability of the Selective Approach Algorithm.

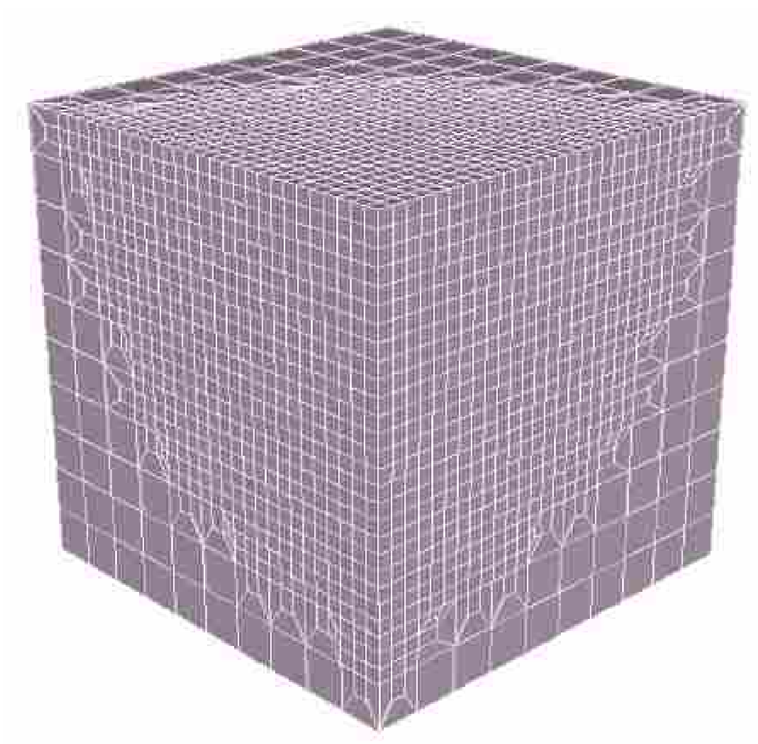


Figure 6-5: Refinement of elements within a radius of top front corner

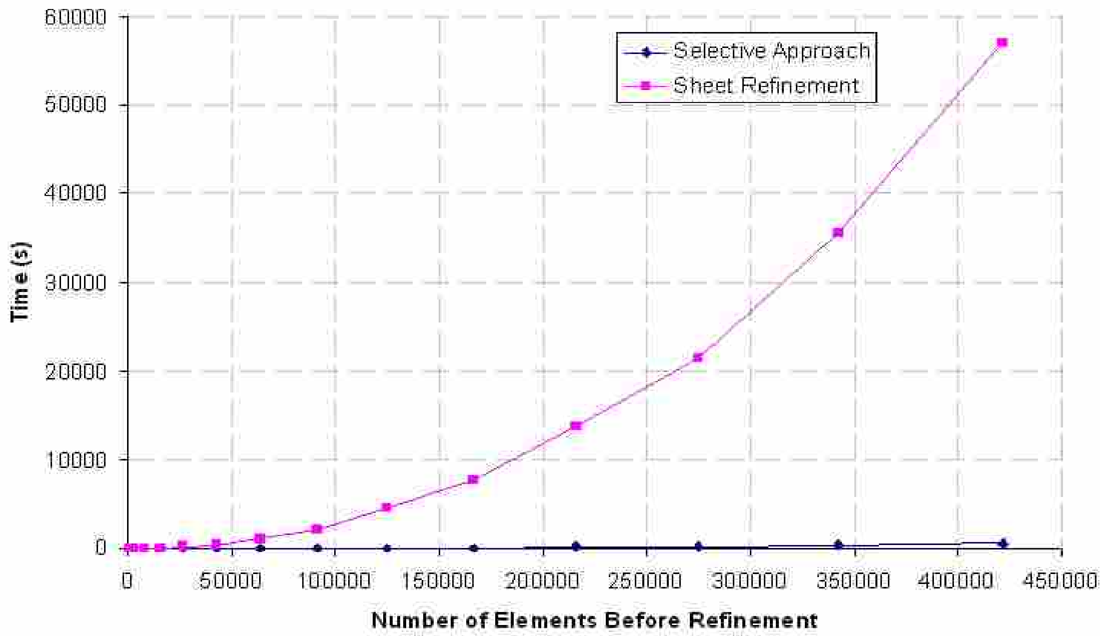


Figure 6-6: Comparison of scalability between sheet refinement and the Selective Approach Algorithm with some elements refined using directional refinement

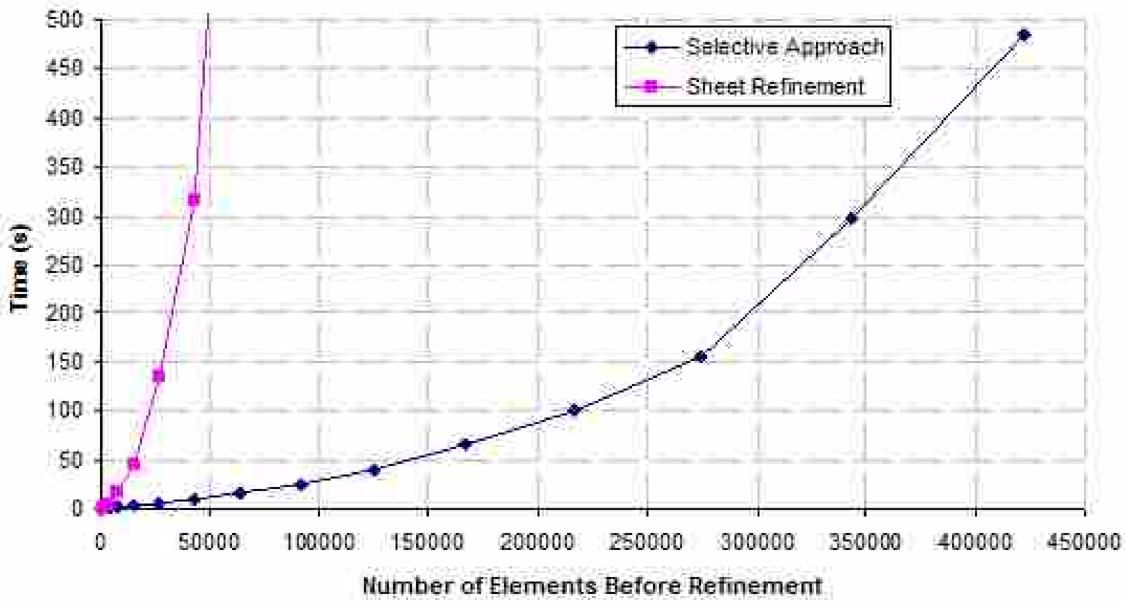


Figure 6-7: Comparison of scalability between sheet refinement and the Selective Approach Algorithm with some elements refined using directional refinement (y-axis reduced).

Like sheet refinement, the scalability of the Selective Approach Algorithm increases exponentially as the number of elements is increased when using directional refinement. However, the rate at which time increases is much less with the Selective Approach Algorithm. For example, the last data point taken in this scalability test required about eight minutes to complete with the Selective Approach Algorithm while the sheet refinement algorithm implemented by Harris required over 15 hours to complete.

6.4 Example

The example considered is a model of one quarter of a piston (see Figure 6-8). All top surfaces of the model were refined using both the sheet refinement algorithm implemented by Harris and the Selective Approach Algorithm. Number of elements, speed, and quality were considered in the analysis and the model was smoothed before calculating the final element qualities. Figure 6-9 contains snapshots of the model after both refinement schemes were preformed. The results are given in Table 6-2.

Table 6-2: Numerical results of refining the top surfaces of a piston

Measurement	Sheet Refinement	Selective Approach
Initial Element Count	1720	1720
Final Element Count	20660	17348
Time (sec)	7.735	1.984
Initial Minimum Quality	0.6286	0.6286
Final Minimum Quality	0.2269	0.1856
Final Minimum Quality (Smoothed)	0.3211	0.3201

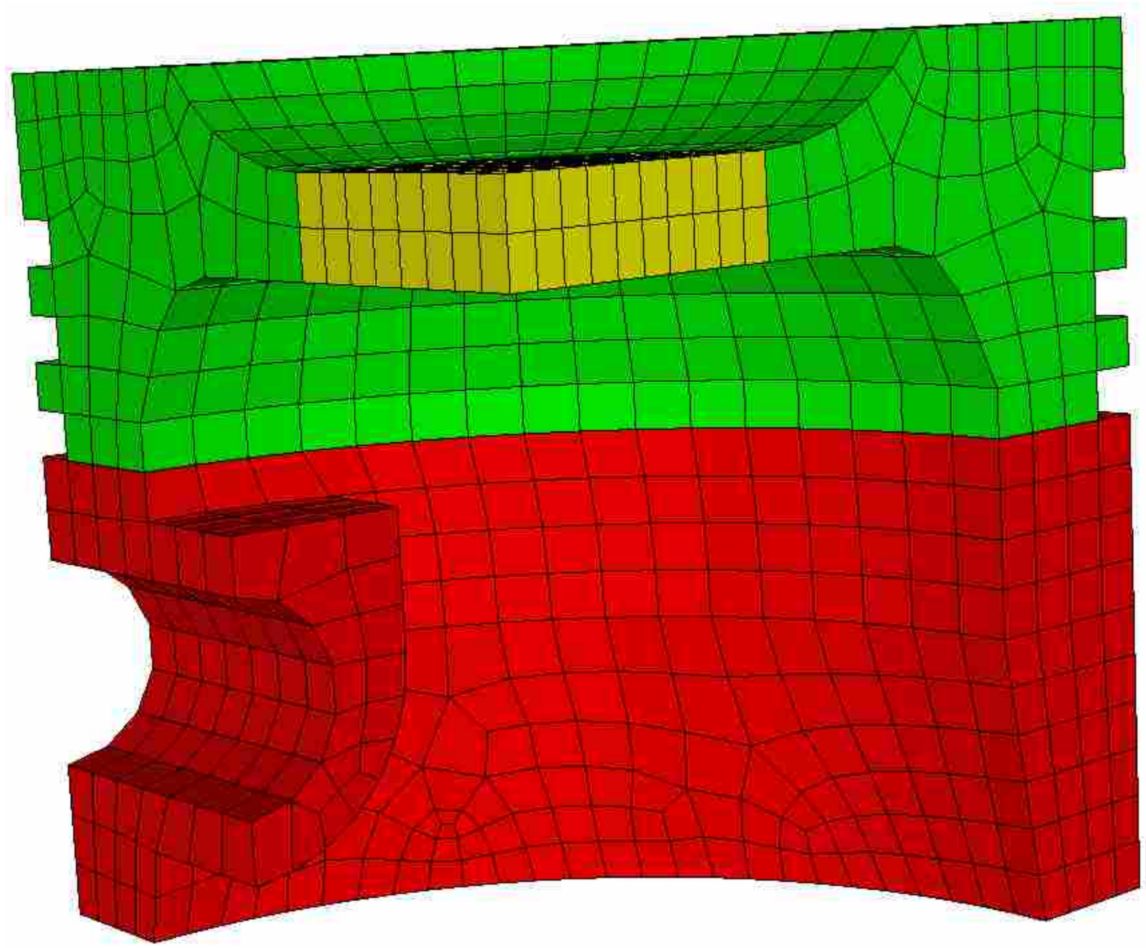
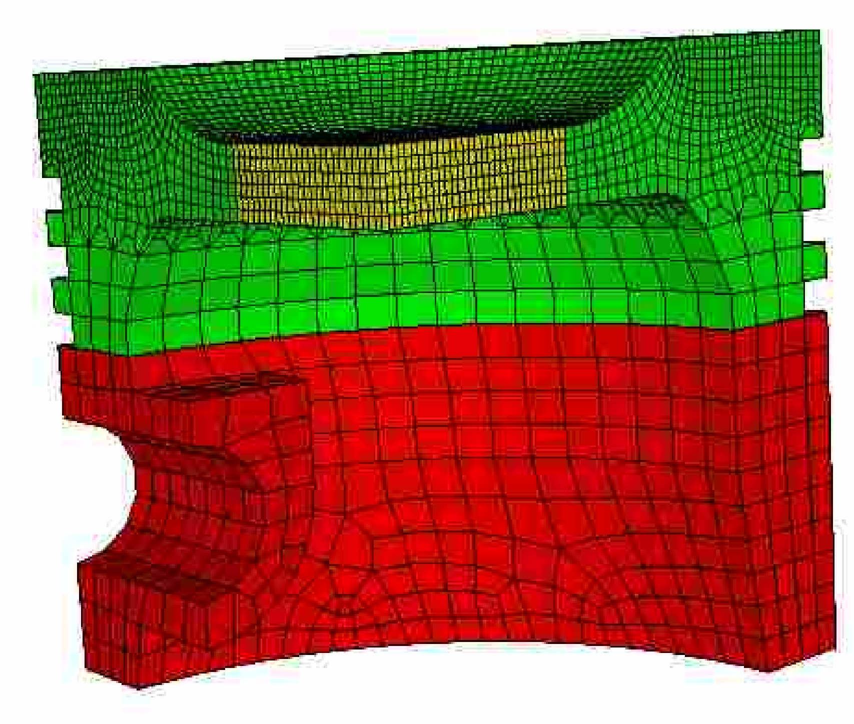
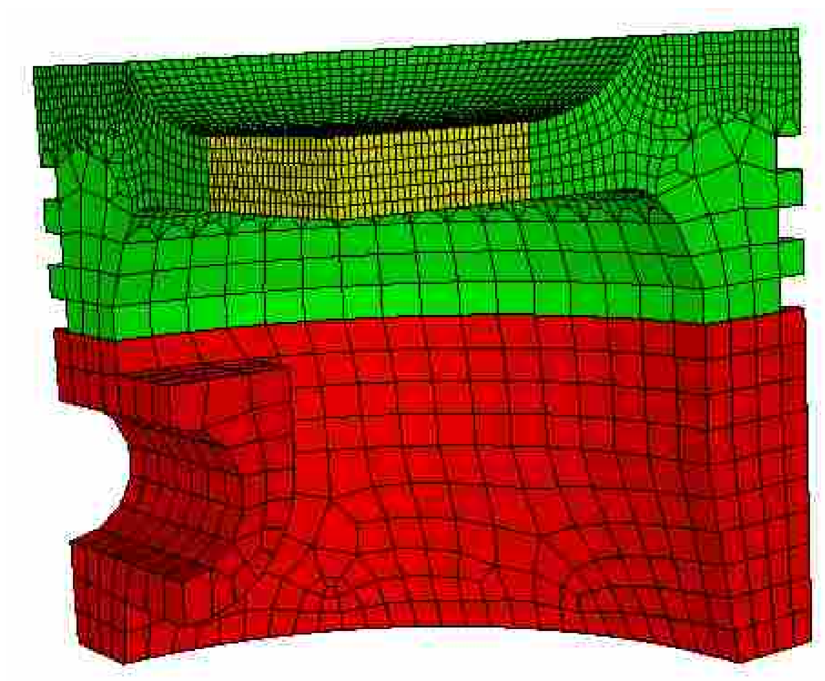


Figure 6-8: Quarter of a piston

In this example, the Selective Approach Algorithm outperforms sheet refinement in both final number of elements and time. The final quality using the Selective Approach method is also adequate for an analysis and comparable to the sheet refinement scheme. This example shows that the Selective Approach Algorithm maintains the robust features found in sheet refinement with improved speed and a lower element count.



(a) Sheet Refinement



(b) Selective approach

Figure 6-9: Snapshots of piston after refinement

7 Conclusion

The refinement scheme presented in this work is a powerful mesh modification tool. The Selective Approach Algorithm is able to handle self-intersecting hexahedral sheets, multiply-connected transition elements, and scalability issues by leveraging the advantages of both element by element and sheet refinement schemes. Directional refinement is a new refinement technique that refines the three inherent directions of a hexahedron sequentially while the target region is processed on a hexahedron by hexahedron basis. A ranking system that utilized the dual of the mesh and a propagation scheme allowed directional refinement to work properly within the confines of the Selective Approach Algorithm. The algorithm appears to have a scalability that is nearly linear when directional refinement is not needed. When directional refinement is required, the scalability of the Selective Approach Algorithm increases exponentially however it is on a much smaller scale then Harris' sheet refinement algorithm. Also, the robustness that existed in sheet refinement is not lost within the Selective Approach Algorithm. An example was also given that provided evidence of this new algorithm's power.

References

1. Benzley S, Perry E, Merkley K, Clark B, and Sjaardema G (1995) *A Comparison of All Hexahedral and All Tetrahedral Finite Element Meshes for Elastic and Elasto-plastic Analysis*. In: Proceedings 4th International Meshing Roundtable, pages 179-191. Sandia National Laboratories.

2. Tchon K.-F, Hirsch C, and Schneiders R (1997) *Octree-Based Hexahedral Mesh Generator for Viscous Flow Simulations.* In: 13th AIAA Computational Fluid Dynamics Conference, No. AIAA-97-1980. Snownass, CO.

3. Marechal L (2001) *A New Approach to Octree-Based Hexahedral Meshing*. In: Proceedings 10th International Meshing Roundtable, pages 209-221. Sandia National Laboratories.

4. Schneiders R (1996) *Refining Quadrilateral and Hexahedral Element Meshes*. In: 5th International Conference on Numerical Grid Generation in Computational Field Simulations, pages 679-688. Mississippi State University.

5. Schneiders R (2000) *Octree-Based Hexahedral Mesh Generation*. In: Int Journal Comput Geom Appl. 10, No. 4, pages 383-398

6. Tchon K.-F, Dompierre J and Camerero R (2002) *Conformal Refinement of All-Quadrilateral and All-Hexahedral Meshes According to an Anisotropic Metric*. In: Proceedings 11th International Meshing Roundtable, pages 231-242. Sandia National Laboratories.

7. Tchon K.-F, Dompierre J and Camerero R (2004) *Automated Refinement of Conformal Quadrilateral and Hexahedral Meshes.* In: Int Journal Numer Meth Engng 59:1539-1562

8. Harris N (2004) *Conformal Refinement of All-Hexahedral Finite Element Meshes*. MA Thesis, Brigham Young University, Utah

9. Murdoch P, Benzley S, Blacker T, and Mitchell S (1997) *The Spatial Twist Continuum: A Connectivity Based Method for Representing all-Hexahedral Finite Element Meshes.* In: Finite Elements in Analysis and Design 28:137-149

10. Murdoch P and Benzley S (1995) *The Spatial Twist Continuum*. In: Proceedings 4th International Meshing Roundtable, pages 243-251. Sandia National Laboratories.

11. Carbonera C. and Shepherd J (2006) *A Constructive Approach to Constrained Hexahedral Mesh Generation*. In: Proceedings 15th International Meshing Roundtable, pages. 435-452. Sandia National Laboratories

12. Benzley S, Harris N, Scott M, Borden M, and Owen S (2005) *Conformal Refinement and Coarsening of Unstructured Hexahedral Meshes*. In: Journal of Computing and Information Science in Engineering 5:330-337

13. Esmaeilian S (1989) *Automatic Finite Element Mesh Transitioning with Hexahedron Elements*. Doctoral Dissertation, Brigham Young University, Utah

14. Blacker T. (2000) *Meeting the Challenge for Automated Conformal Hexahedral Meshing*. In: Proceedings 9th International Meshing Roundtable, pages 11-19. Sandia National Laboratories.

15. Melander D. (1997) *Generation of multi-million element meshes for solid model-based geometries: The dicer algorithm*. MA thesis, Brigham Young University, Utah.

16. Borden M, Benzley S, Mitchell S, White D, and Meyers R. (2001) *The cleave and fill tool: An all-hexahedral refinement algorithm for swept meshes.* In: Proceedings 9th International Meshing Roundtable, pages 69-76. Sandia National Laboratories.

17. Harris N, Benzley S, and Owen S (2004) *Conformal Refinement of All-Hexahedral Element Meshes Based on Multiple Twist Plane Insertion*. In: Proceedings 13th International Meshing Roundtable, pages 157-167. Sandia National Laboratories.

18. Mitchell S.A. and Tautges T.J. (1995) *Pillowing doublets: refining a mesh to ensure that faces share at most one edge*. In: Proceedings 4th International Meshing Roundtable, pages 231-240. Sandia National Laboratories.

Appendix A. The Hexahedron and Hexahedral Meshing

Before one can delve into the realm of hexahedral refinement, one must understand the basic principles of hexahedral meshing. This appendix identifies basic characteristics of the hexahedron and explains three major constraints on all-hexahedral meshing [14].

The Hexahedron

The hexahedron is the basic element in an all-hexahedral mesh and can be viewed as three pairs of opposing faces. Though this definition of the hexahedron seems simple, the implications derived from it are significant. Collectively, the hexahedron contains six quadrilateral faces, twelve edges, and eight nodes as shown in Figure A-1.

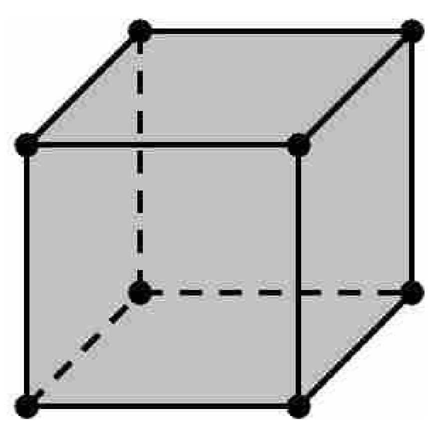


Figure A-1: The hexahedron

Connectivity Constraints

To maintain connectivity, each quadrilateral face of a hexahedron must border an equally dimensioned face of a neighboring hexahedron or be located on a boundary. While this constraint remains true, the hexahedral mesh is called a conformal mesh. Many finite element solvers require this constraint, therefore it is essential that conformity is maintained throughout the mesh.

By lining up hexahedral elements so that each element has two neighboring elements that are attached to opposing faces, a stack of hexahedral elements is formed as shown in Figure A-2. A stack of elements must begin and end at a boundary or be a closed loop of elements. Hexahedral sheets are formed by grouping stacks of elements in a second dimension as shown in Figure A-3. Each element in a hexahedral sheet has four neighboring elements that are attached to two orthogonal pairs of opposing faces. Similarly, hexahedral sheets must begin and end at a boundary or form closed loops. The dual of the mesh, as will be discussed later, represents these connectivity characteristics through chords and twist planes.

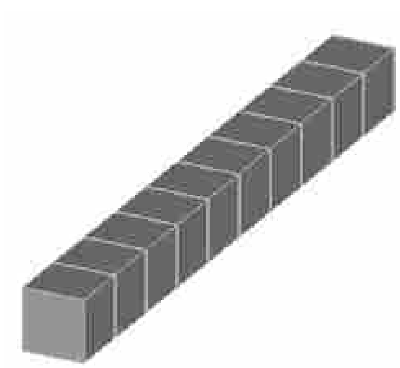


Figure A-2: Stack of elements

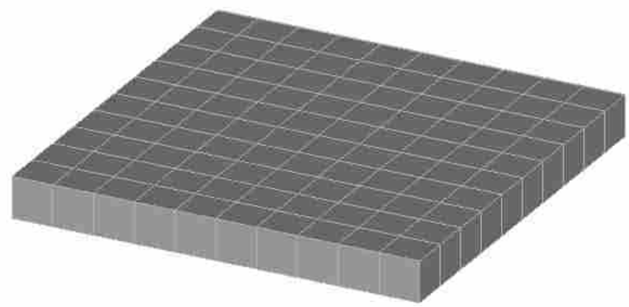
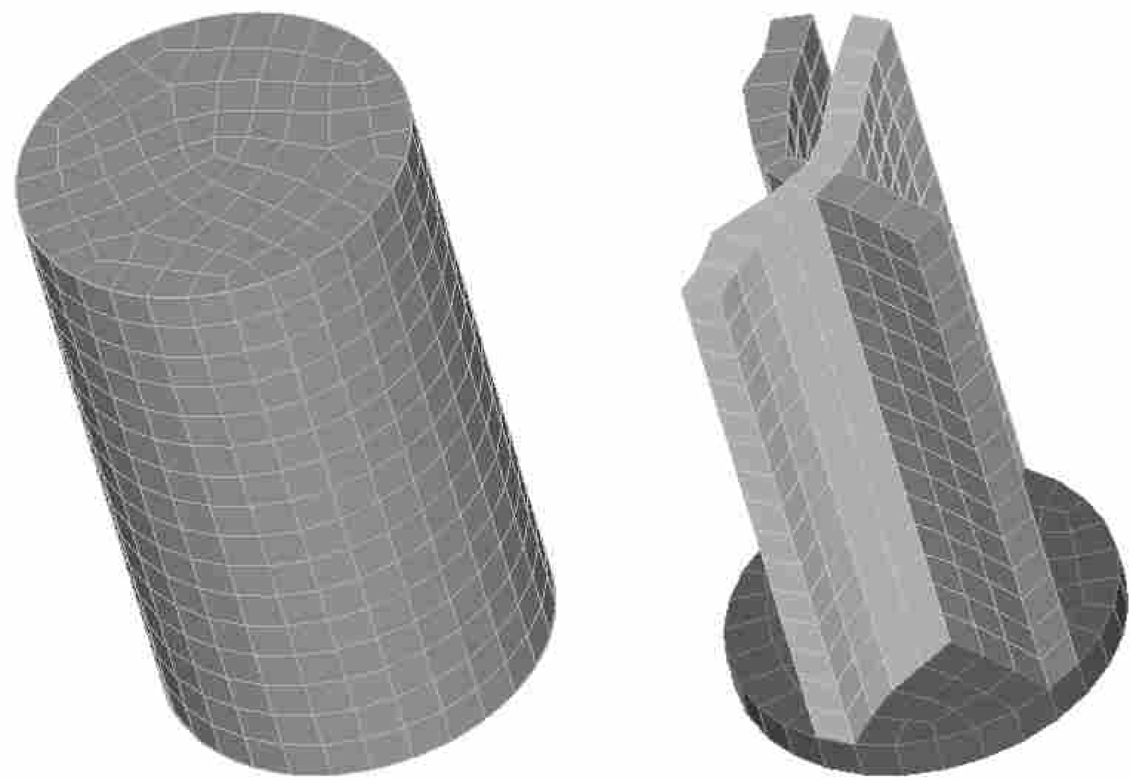


Figure A-3: Hexahedral sheet



(a) **(b)**

Figure A-4: (a) Meshed cylinder, (b) Three intersecting hexahedral sheets of cylinder mesh

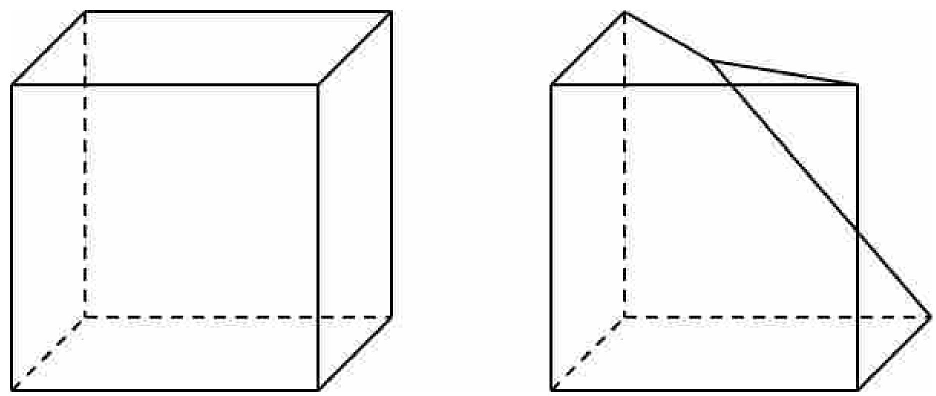
Conformal all-hexahedral meshes are composed of multiple intersecting hexahedral sheets which gives the mesh its characteristic connectivity as shown in Figure A-4. Because of this characteristic, it is impossible to insert or remove an individual element. An entire sheet must be inserted or removed to maintain a conformal mesh. This idea can be further extended in that whenever any modification occurs to an element that extends to the border of a neighboring element, the neighboring element must also be modified to maintain a conformal mesh. This has significant impact on localized hexahedral refinement.

Quality Constraints

The accuracy of a finite element analysis is directly correlated to the quality of individual elements within the mesh. If the quality is too poor, the analysis becomes unacceptable. The element quality becomes unacceptable when the Jacobian becomes negative. This usually occurs when the internal angles between faces are greater than 180 degrees. The best quality is obtained when all interior angles are 90 degrees. Figure A-5(a) depicts an ideal hexahedral element and Figure A-5(b) depicts an unacceptable element which is typically called an inverted element.

Geometric Constraints

An all-hexahedral mesh requires that all surfaces be meshed using quadrilateral elements. These quad meshes must conform to the geometry of the model and therefore become sensitive to geometry constraints. Small angles are particularly difficult to mesh with a high quality. For example, Figure A-6 shows a poor quality quad mesh of a triangle. Poor surface meshes can also be propagated to the interior of a hexahedral mesh thus causing unwanted distortions and poor hexahedral quality.



(a) (b)

Figure A-5: (a) An ideal hexahedral element, (b) An inverted element

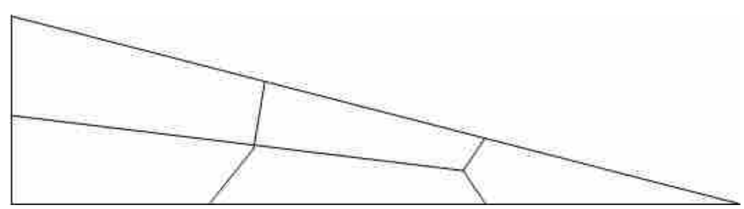


Figure A-6: Poor element quality with small angle

Appendix B. The Dual

The dual of the mesh or the spatial twist continuum (STC) is a powerful geometric representation of the inherent connectivity within a mesh [9][10]. Therefore, any type of mesh has a dual representation. For the purposes of this thesis the dual will be described for an all-quadrilateral mesh first and then be expanded to a three-dimensional all-hexahedral mesh.

The Dual of a Quadrilateral Mesh

Figure B-1 shows a quadrilateral mesh with its corresponding dual. In a two-dimensional mesh, the dual is composed of three components. These are:

- Centroids
- Edges
- 2-Cells

The black dots represent the centroids. The dotted lines connecting the centroids are representative of the dual edges. Lastly, the 2-Cells are polygons bounded on all sides by dual edges.

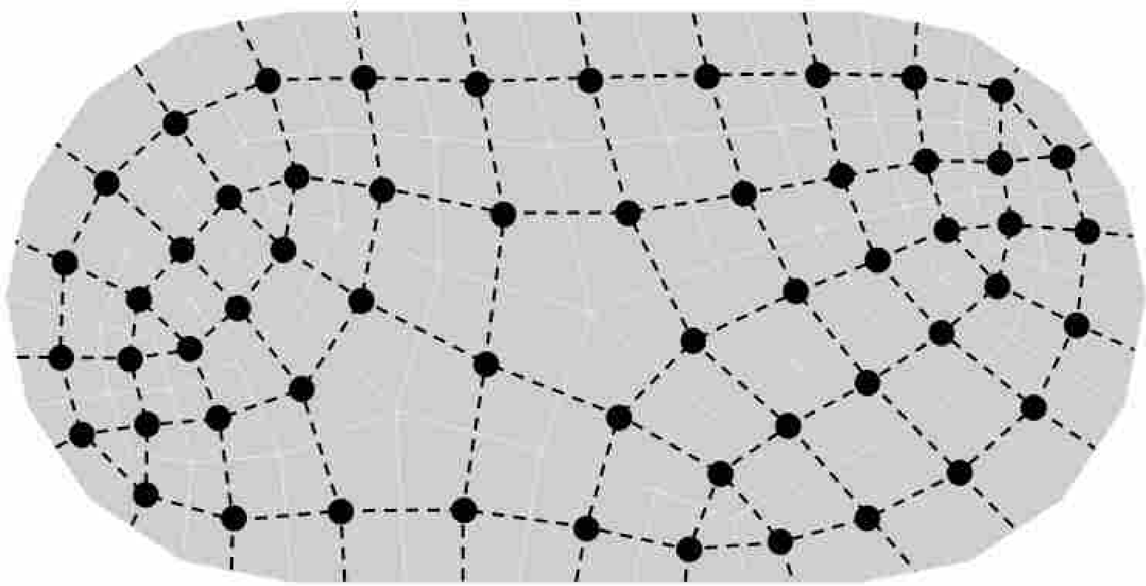


Figure B-1: The dual of a quadrilateral mesh

Construction of the dual of the mesh for Figure B-1 or any other quadrilateral mesh involves two steps.

1. Place a centroid in each quadrilateral element.
2. Whenever two elements share a face, add an edge to connect the two corresponding centroids.

An important aspect of the dual in two-dimensions is the relationship between mesh entities and dual entities. Table B-1 lists the mesh entities and their corresponding dual entities. A quadrilateral face of dimension two for example has a corresponding centroid dual entity which has a dimension of zero. This relationship can be extended to three dimensions and will be shown later.

Table B-1: Relationship between mesh entities and dual entities

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

Quadrilateral meshes contain a unique property in that dual edges that correspond to opposite sides of a quadrilateral element can be grouped together into a continuous curve called a chord. Dual chords describe the global connectivity of an all-quadrilateral mesh. A chord actually represents a stack of quadrilateral elements and a quadrilateral mesh can be viewed as an intertwining of dual chords. The validity and quality of an all-quadrilateral mesh is directly related to how these dual chords are intertwined. Murdock presented a list of six properties that dual chords must adhere to for a quadrilateral mesh to be valid. These six properties are listed below.

1. A chord that begins on a boundary must terminate on the boundary.
2. A chord that does not begin on the boundary must form a closed loop within the mesh.
3. Chords may cross each other multiple times, but such crossings may not be consecutive. This ensures that two quadrilaterals will not share two edges.
4. A chord is allowed to cross itself provided each self-intersection is separated by four other centroids.
5. Each centroid is passed through exactly twice, either by two distinct chords or one chord twice. This constraint ensures that each element has only four edges.
6. Chords are nowhere tangent.

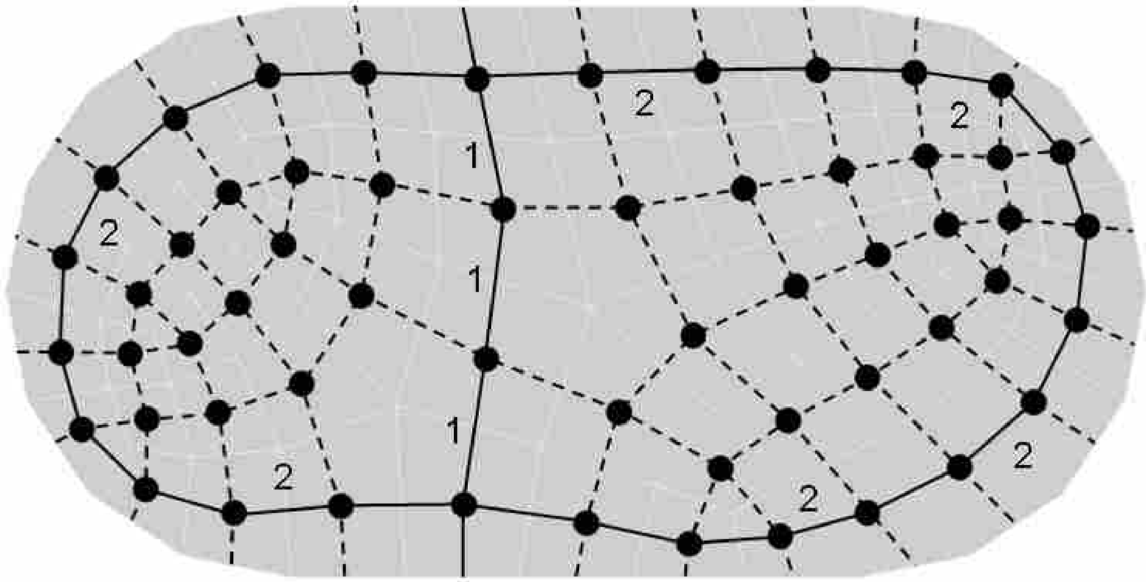


Figure B-2: Dual with chords

Figure B-2 shows some of the characteristics described in the list above. Chords are indicated with solid black lines and labeled for clarification. Chord 1 demonstrates a chord starting and finishing on a boundary while chord 2 demonstrates a chord that forms a closed loop. Notice also that no chord crosses another chord consecutively ensuring that no two quadrilaterals share two edges. Figure B-2 further demonstrates that each centroid is only passed through twice, a requirement for an all-quadrilateral mesh. Figure B-3 shows how a dual chord can self-intersect. Since the self-intersection is separated by at least four centroids, this is a valid mesh.

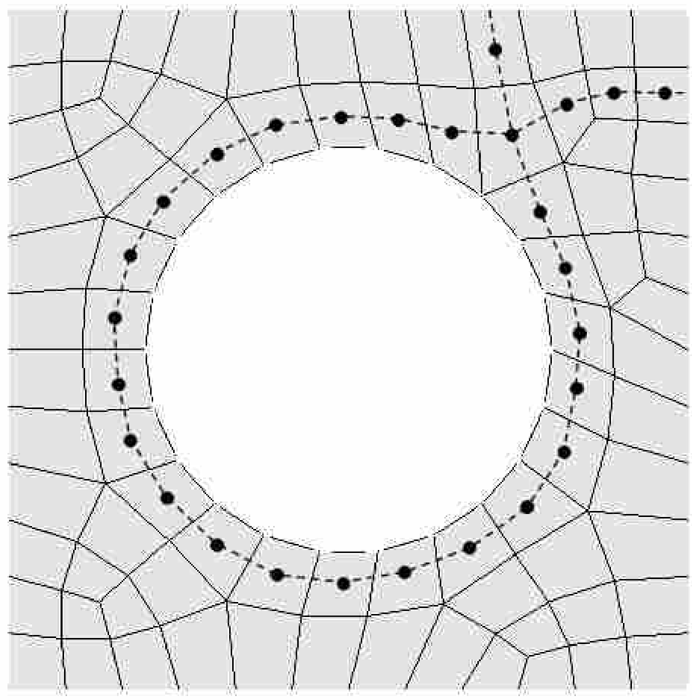


Figure B-3: Self-intersecting chord

The Dual of a Hexahedral Mesh

The ideas presented in the previous section can be directly expanded to all-hexahedral meshes. As in two-dimensions, basic components of the dual exist and are outlined below.

- Centroid
- Edge
- 2-Cell
- 3-Cell

As with the dual in two dimensions, each dual element directly relates to hexahedral mesh element as shown in Table B-2.

Table B-2: Relationship between hexahedral mesh entities and dual entities

Mesh Entity	Dimension	Dual Entity	Dimension
Hex	3	Centroid	0
Face	2	Edge	1
Edge	1	2-Cell	2
Node	0	3-Cell	3

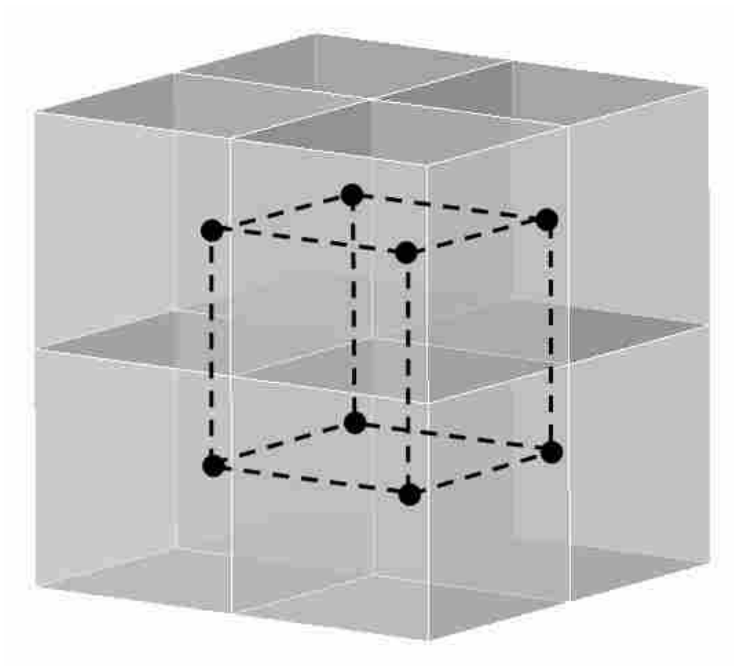


Figure B-4: Eight hexahedra with corresponding dual entities

Figure B-4 depicts eight hexahedra with some of their corresponding dual entities. All of the dual entities were not included for clarity in this discussion. Centroids are indicated with black circles in the center of each hexahedron. Dual edges are indicated by the dashed lines and connect centroids of elements that share a face. 2-Cells are polygons of dual edges that form a face similar to a mesh face. Six 2-Cells are shown in Figure B-4 each one representing an interior mesh edge. These 2-Cells are perpendicular to and intersect the mesh edge they represent. 3-Cells are three dimensional polyhedrons that represent a node. The single 3-Cell (area bounded by the 12 dual edges in Figure B-4) represents the interior node of the eight hexahedra shown. The dual can be constructed in a similar fashion to that of a two-dimensional dual. The steps are as follows:

1. A centroid is placed in each hexahedron
2. A dual edge is constructed by connecting the centroids of elements that share a face.

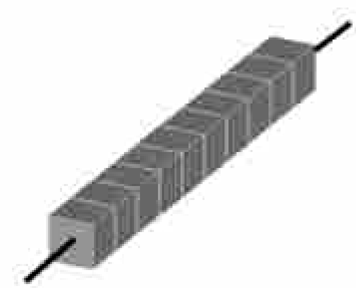


Figure B-5: Stack of elements with corresponding dual

As in two dimensions, dual elements can be combined to globally describe the connectivity of the mesh. Dual chords also exist in three dimensions. They are formed by combining the two dual edges that represent opposite faces in a hexahedron together and then propagating that connection throughout the entire mesh. A dual chord along with the hexahedral elements it represents is shown in Figure B-5. Notice that the dual chord in this case graphically represents a stack of elements. Also, as two chords intersecting in two dimensions can define a quadrilateral element, the intersection of three chords in three dimensions can define a hexahedron. The same constraints presented by Murdoch that were applicable in two dimensions are also applicable in three dimensions.

Another, and often more powerful, way to represent the connectivity of a hexahedral mesh is by the use of a twist plane. A twist plane is created by grouping the 2-Cells which are logically perpendicular to a chord at a centroid. Figure B-6 shows a twist plane and the hexahedra that it represents. A twist plane always represents a specific hexahedral sheet within a hexahedral mesh and the same principles discussed by Murdoch for dual chords apply to twist planes. Each hexahedron contains three such twist planes (see Figure B-7) and therefore each hexahedron has three inherent directions normal to these twist planes. This idea of three unique directions within a hexahedron is critical to an understanding of the work presented in this thesis.

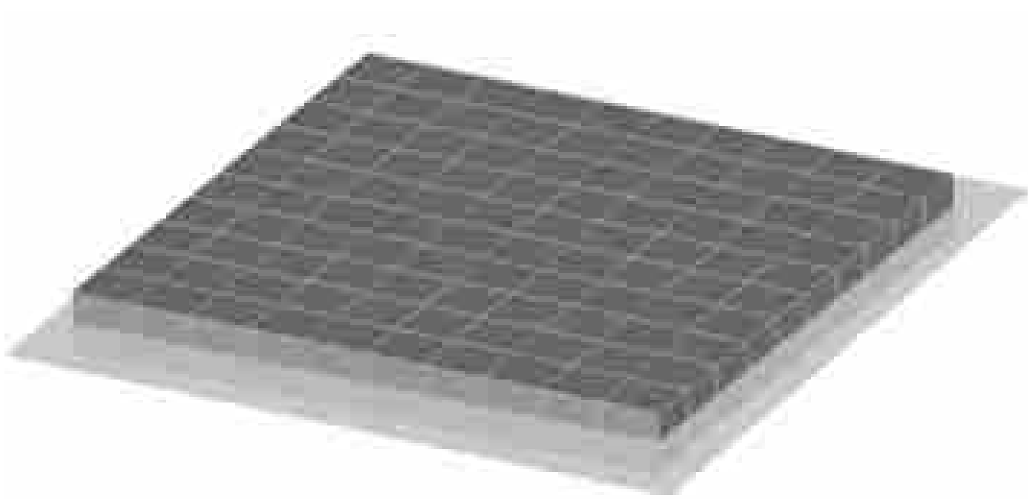


Figure B-6: Hexahedral sheet with twist plane

Both the dual chord and twist plane are powerful tools used to represent the connectivity of quadrilateral and hexahedral meshes. These tools have been used in previous refinement techniques and will be used in the hexahedral refinement algorithm discussed in this thesis.

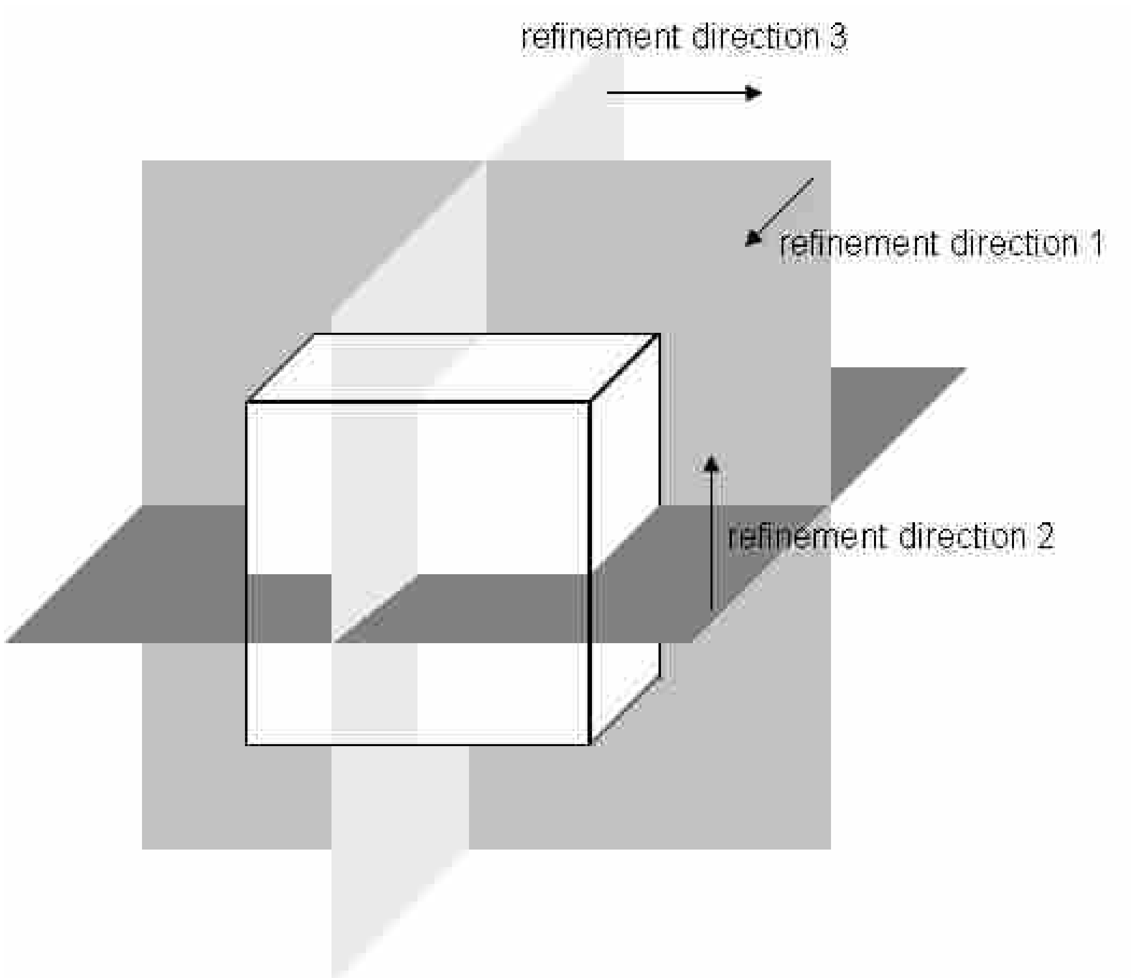


Figure B-7: A hexahedron with its three inherent directions normal to twist planes

Appendix C. Hexahedral Refinement Techniques

Refinement is not new to the meshing community. It has a myriad of applications and has therefore been the topic of much research. This appendix will describe in detail many refinement schemes currently found in the literature. Particular attention will be given to the element by element and sheet refinement schemes since they provide the foundation of this thesis. The list of refinement schemes presented here is by no means exhaustive. These refinement schemes were selected because of their importance to the meshing community and their relevance to this thesis. A general explanation of each refinement scheme will be given as well as a brief discussion of each scheme's strengths and weaknesses.

Octrees

Octrees, as the name suggests, refines one element into eight elements [2]. This is accomplished by splitting each edge at its midpoint. The refinement process involves iteratively inserting octrees into a mesh until the desired size is reached. This method is quick and provides excellent control over localization and element size. The major drawback to this type of refinement is that it can produce a non-conformal mesh. Many finite element solvers are unable to handle non-conformal meshes and thus octree refinement can be very limiting. Figure C-1 shows a simple cube that has been refined using octrees.

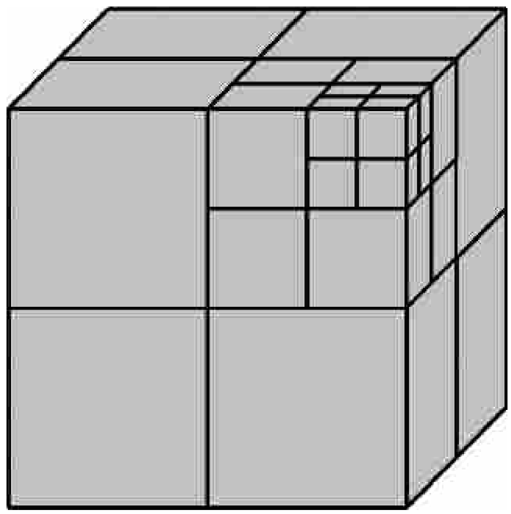


Figure C-1: Refinement using octrees

Dicing

The dicer algorithm was developed to create multi-million element meshes [15]. The algorithm uses parametric mapping to refine coarse elements allowing large numbers of elements to be generated quickly. An efficient storage scheme is also used taking advantage of the structured nature of the refinement. This allows the dicer algorithm to be both quick and efficient. The dicer algorithm is limited in that it can only refine full hexahedral sheets. This means that it cannot do any localized modification. Another limitation of the dicer algorithm is that all geometry features must be resolved with the coarser mesh. While these limitations are inconvenient, for the purposes for which it was designed, the dicer algorithm is an effective hexahedral refinement tool. Figure C-2 shows a hexahedral mesh refined using the dicer algorithm.

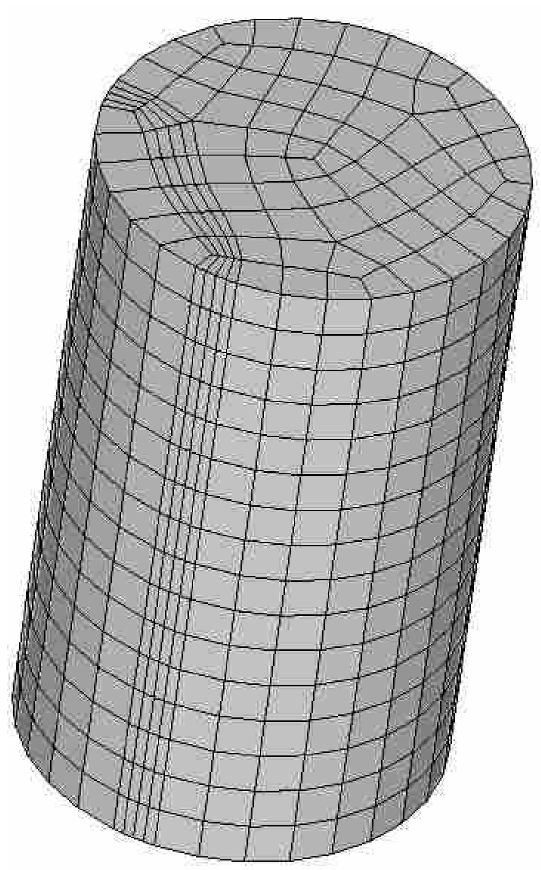


Figure C-2: Refinement using dicer algorithm

The Cleave-and-Fill Tool

The cleave-and-fill tool is an adaptation of sheet insertion and was designed to refine the region between source and target surfaces of swept meshes thus helping to improve the mesh quality in some cases [16]. This makes the cleave-and-fill tool too specific for a general refinement algorithm.

Element by Element Refinement

Element by element hexahedral refinement attempts to refine a hexahedral mesh by inserting a template that refines all three directions of a hexahedron in one step. These templates replace each of the original hexahedra within the target region. The difficulty then in the element by element refinement scheme is to maintain conformity by inserting the proper template with the proper orientation. The transition region is of most concern since templates inserted into this area must connect the fine mesh to the coarse mesh while maintaining conformity.

Schneiders introduced an element by element refinement scheme in connection with an octree-based mesh generator [4]. This refinement technique worked well in many cases; however, it is unable to create a conformal mesh where multiply-connected transition elements were present. In hexahedral refinement, a multiply-connected transition element refers to a hexahedral element that is not selected for refinement but shares more than one face with hexahedra that are selected for refinement. Currently, many of the templates to handle these transition elements are unknown. This fact limits the potential for conformal refined meshes using the element by element approach. Some of the known templates that handle multiply-connected transition elements are discussed in the appendix entitled Templates.

Figure C-3 graphically illustrates the element by element refinement process. The hexahedron selected for refinement is removed and replaced with the appropriate template. Next, the transition elements are removed and replaced with their appropriate templates. Notice that in element by element refinement the mesh is non-conformal for most of the refinement process. It is only when the refinement scheme is finished that conformity is restored.

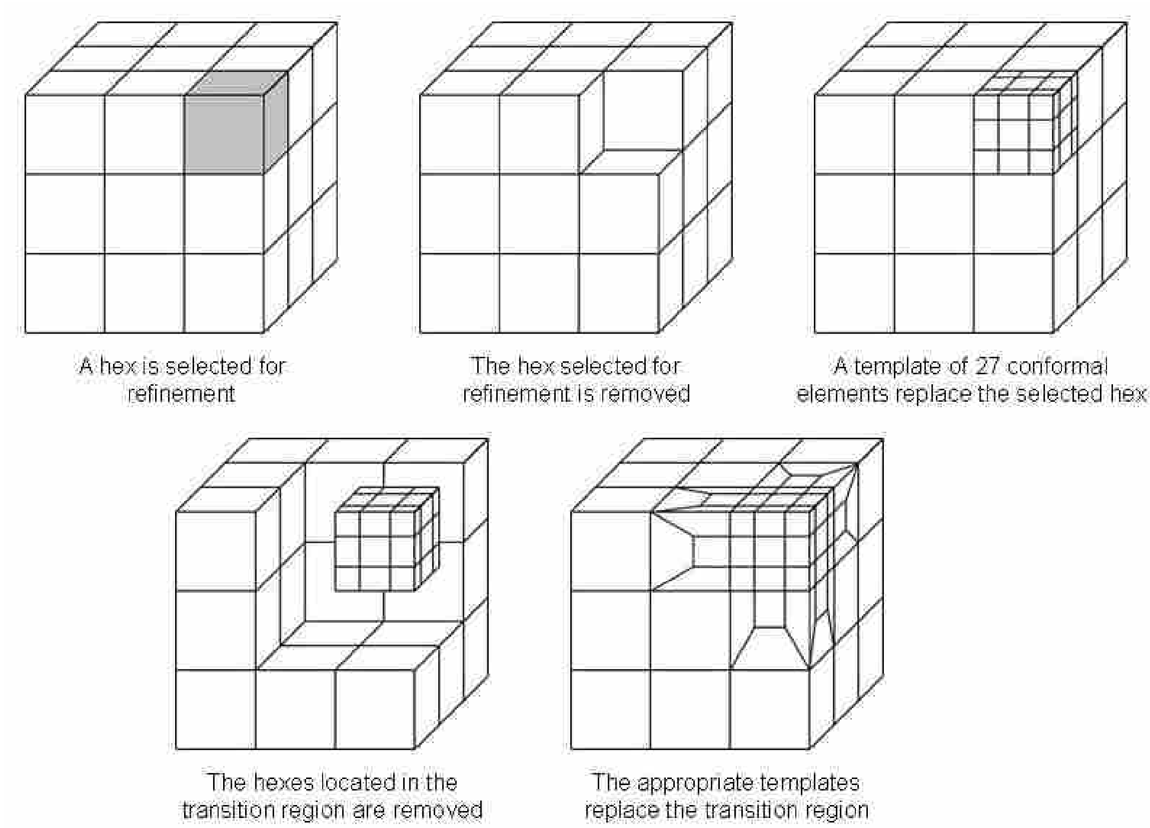


Figure C-3: Element by element refinement process

Sheet Refinement

Schneiders also proposed a sheet refinement method that refines a mesh by pillowing each inherent direction of a hexahedron separately [5]. This refinement scheme eliminates the multiply-connected transition element problem inherent in element by element refinement thus always producing a conformal mesh. This method was originally proposed for structured meshes.

Tchon expanded upon Schneiders multi-directional refinement to include refinement of unstructured meshes [6][7]. Sheet refinement thus occurs by pillowing hexahedral sheets according to an anisotropic size metric rather then refining individual elements. This refinement scheme is capable of local conformal refinement, and offers great user control over the target region.

Harris further expanded upon Tchon's work by using template insertion instead of pillowing in sheet refinement [8]. Only three templates are required to accomplish this type of refinement. Harris further generalized the refinement process to include nodes, edges, and faces as possible targets for hexahedral refinement. Again, this type of refinement in general offers refinement localization, produces a conformal mesh, and offers excellent user control of the refinement region.

Figure C-4 shows the general process of sheet refinement for a three-dimensional mesh. The first sheet is processed resulting in refinement in one single direction. The second sheet is then processed refining in a second direction. As the third sheet is processed, the third direction is refined resulting in the final mesh. It is important to note that the mesh is conformal during the entire refinement process, thus ensuring that the final mesh will also be conformal.

While the refinement algorithm proposed by Harris can be considered the most robust of all refinement schemes presented thus far, it is not without its limitations. In fact, the limitations of the Harris algorithm were the driving force of the work presented in this thesis. Since these limitations are vital to an understanding of capabilities of the Selective Approach Algorithm, a detailed description of each limitation will follow.

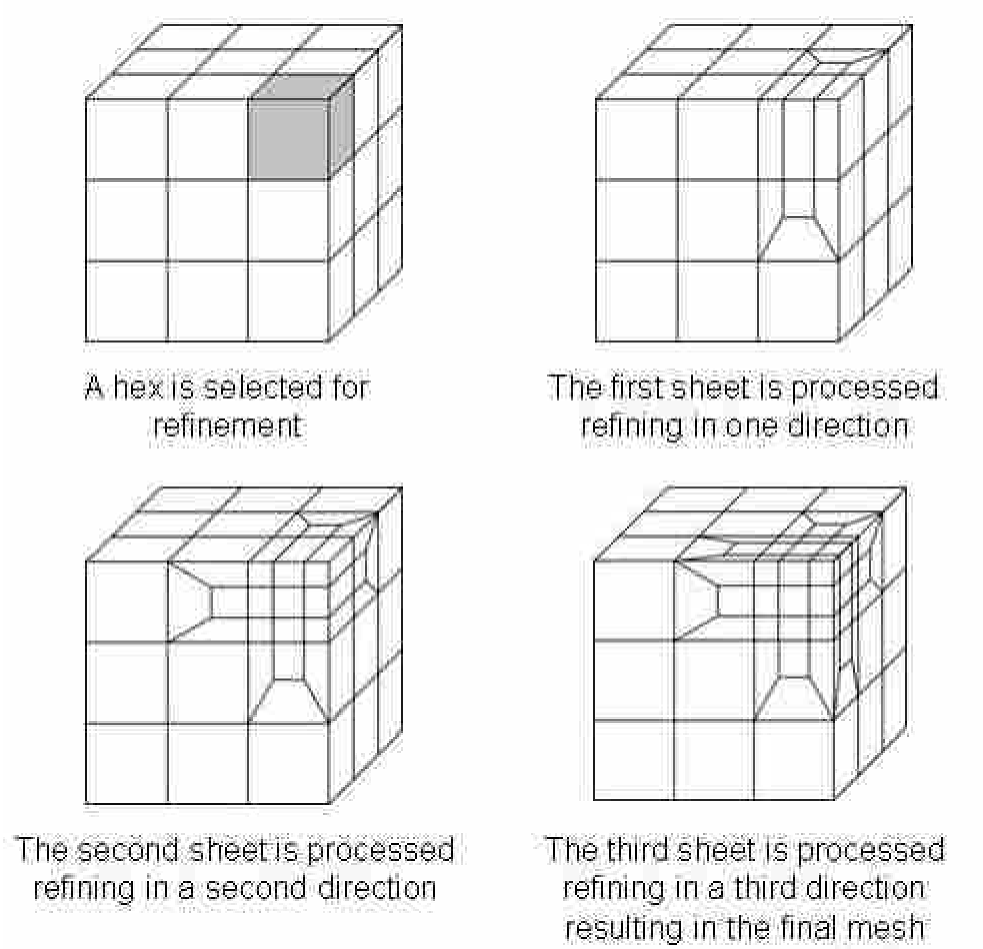


Figure C-4: Sheet refinement process

Self-Intersecting Hexahedral Sheets

Self-intersecting hexahedral sheets can occur anytime an unstructured hexahedral mesh is present. Appendix A discussed the connectivity and dual of a hexahedral mesh. In that appendix, it was shown that an all-hexahedral mesh can be described as the intertwining of hexahedral sheets where each hexahedral sheet must either form a closed loop or both ends must exit at boundaries of the mesh. Murdock proposed a set of criteria for chords as well as twist planes in order for the mesh to maintain its conformity. One of these criteria states that twist planes may self-intersect provided there are sufficient stacks of elements between the self-intersecting hexahedra so that no hexahedra share two adjacent faces. Figure C-5 shows an all-hexahedral mesh with a self-intersecting hexahedral sheet highlighted within the mesh.

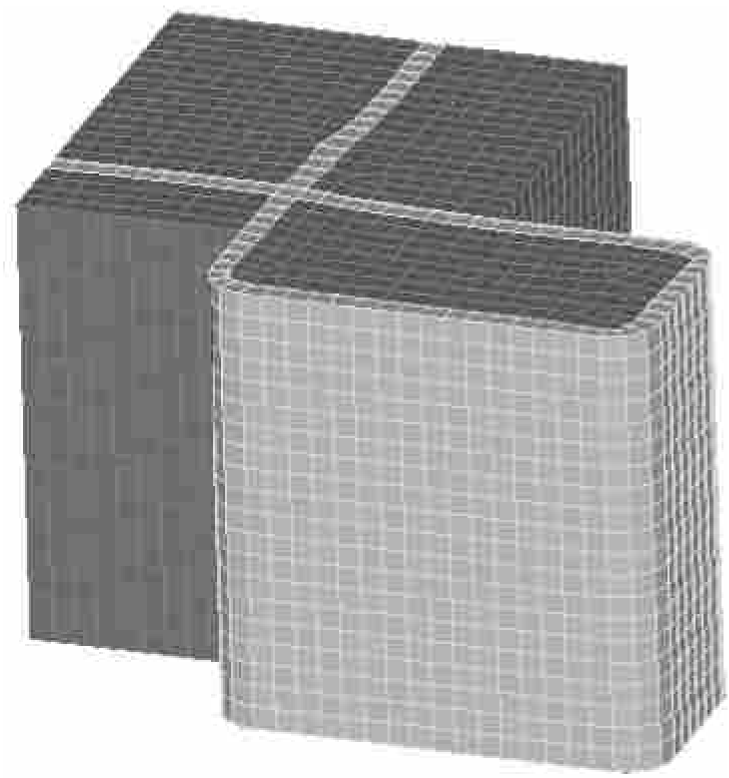


Figure C-5: Mesh containing self-intersecting hexahedral sheet

While self-intersecting hexahedral sheets are not common, they do occur. One of the limitations of Harris' algorithm is that self-intersecting hexahedral sheets must be handled as a special case. Recognizing every case where a hexahedral sheet intersects itself is difficult and error prone.

Multiply-Connected Transition Elements

As stated previously, a multiply-connected transition element in hexahedral refinement refers to a hexahedral element that is not selected for refinement but shares more than one face with hexahedra that are selected for refinement. An example of a multiply-connected transition element is shown in Figure C-6. An adjustment template was proposed for this case which is discussed in Appendix E. Sheet refinement is able to produce a conformal mesh in a region local to multiply-connected transition elements by two different methods. The first method involves adding hexahedra to the region selected for refinement in the region local to multiply-connected transition elements until these elements no longer exist. This resulted in excessive refinement which is not needed nor intended by most users. Figure C-7 shows a two-dimensional example of how the multiply-connected transition elements are currently handled. Figure C-7(a) shows the target hexahedra highlighted in dark grey. Nine hexahedra are added to the target region to remove the multiply-connected transition element as shown in Figure C-7(b). Figure C-7(c) shows the final mesh. In this example, the refinement region ended up being twice as large as was originally intended. This limitation increases computation time during refinement and will also increase the required analysis time later on because of the unintended over-densification of the mesh. The second method involves using templates that are specifically tailored to multiply-connected transition elements. These templates are discussed in detail in Appendix E. Though this method is superior to the former method, it was never implemented into any sheet refinement method.

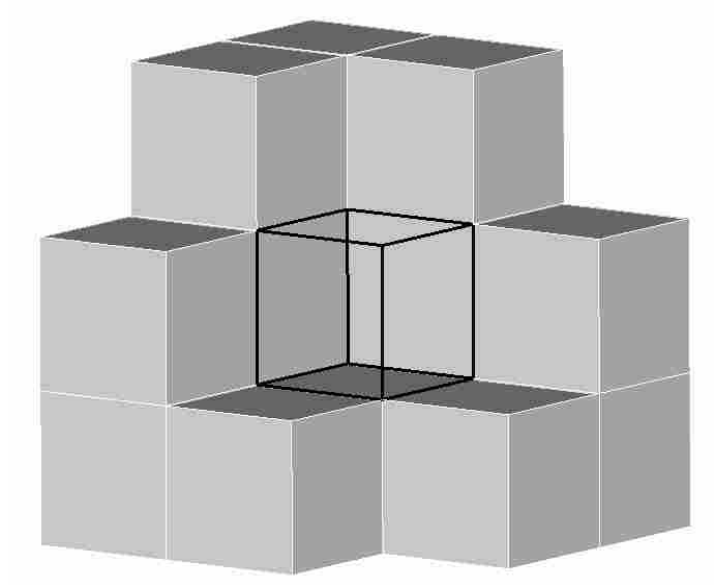
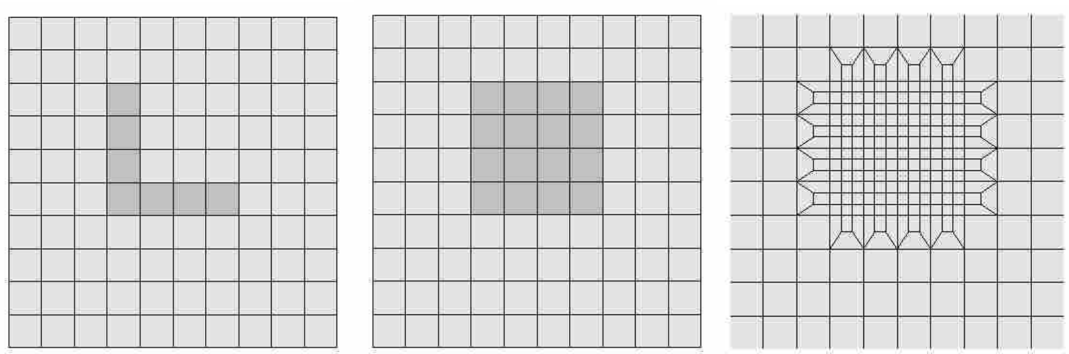


Figure C-6: Example of a multiply-connected transition element



(a) **(b)** **(c)**

Figure C-7: Example of excessive refinement

Scalability

Scalability is by far the biggest drawback to sheet refinement schemes because large meshes require too much time. Figure C-8 shows a graph comparing the number of initial elements to be refined versus time in seconds using the sheet refinement method implemented by Harris. Notice that the time increases exponentially as the number of elements increases.

The reason for the poor scalability is found in the theory of sheet refinement. Since the refinement takes place one direction at a time, many intermediate hexahedra are created and deleted to arrive at a fully refined hexahedron. Initially, one hexahedron is refined into three hexahedra. These three hexahedra are deleted and replaced with nine new hexahedra. Finally, nine hexahedra are deleted and replaced with twenty-seven hexahedra. This means that forty total hexahedra were created and thirteen total hexahedra were deleted to obtain the desired refinement. The creation and deletion of these intermediate hexahedra multiplied by sometimes millions of initial hexahedra results in poor scalability thus limiting the capabilities of sheet refinement for large meshes.

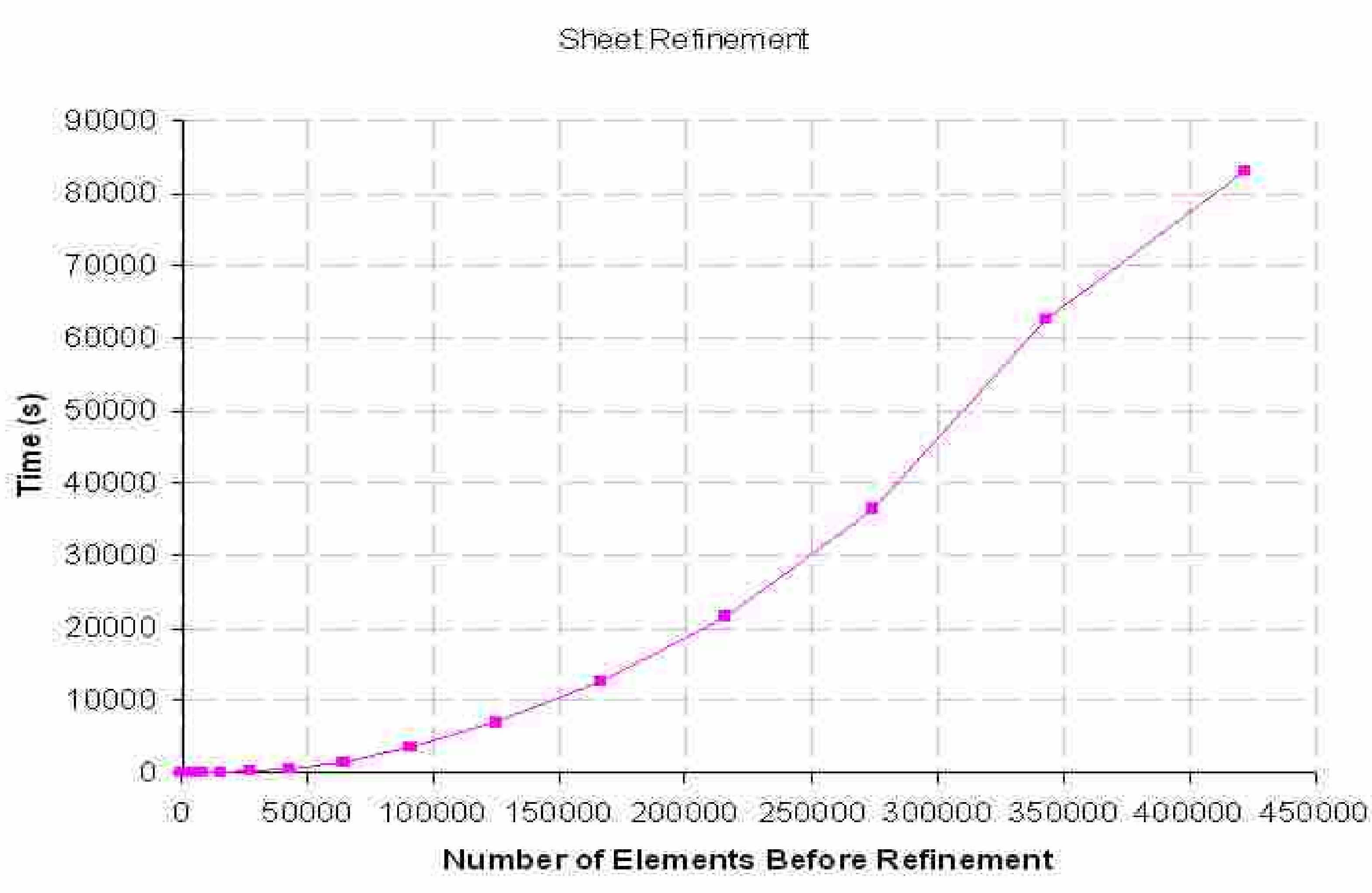


Figure C-8: Poor scalability of sheet refinement scheme implemented by Harris

Appendix D. Element by Element vs. Sheet Refinement

The previous appendix described some of the general hexahedral refinement schemes found in the literature and discussed in detail the element by element and sheet refinement schemes. Since element by element and sheet refinement provide the basis for the Selective Approach Algorithm, this appendix will compare these two schemes in order to determine the benefits of each scheme and how they are applied to the Selective Approach Algorithm.

Requirements and General Comparison

For any refinement algorithm, seven requirements exist which must be adhered to in order for the algorithm to be considered robust [8]. These requirements are listed below.

- Unstructured all-hexahedral refinement
- Localized refinement
- Conformal refinement
- Control over refinement region
- Handle self-intersecting hexahedral sheets
- Handle multiply-connected transition elements
- Scalability

Unstructured refinement, localized refinement, conformal refinement, and control over refinement region were capabilities of sheet refinement developed by Harris. Plausible fixes for self-intersecting hexahedral sheets have been proposed for sheet refinement, however, they are difficult to implement and error prone. Templates to handle multiply-connected transition elements have also been proposed [12] but never implemented for sheet refinement. Scalability, however, has never been addressed. This is because poor scalability is inherent within any sheet refinement scheme since each direction of a hexahedron is refined separately. Element by element refinement does not have inherently poor scalability, but introduces a conformity problem where multiply-connected transition elements exist. Thus, element by element and sheet refinement schemes each lack essential characteristics limiting their capabilities. Table D-1 compares element by element to sheet refinement in their ability to fulfill the requirements stated above.

Table D-1: Comparison of template-based and directional refinement

Requirement	Element by Element	Sheet
Unstructured All-Hexahedral Refinement	x	x
Localized Refinement	x	x
Conformal Refinement		x
Refinement Region Control	x	x
Self-Intersecting Hexahedral Sheets	x	
Handle Multiply-Connected Transition Elements		x
Scalability	x	

Combining Element by Element and Sheet Refinement

A refinement scheme utilizing the strengths of both element by element and sheet refinement is one possible solution. Element by element refinement is clearly the superior method when looking at hexahedral refinement in terms of scalability. However, any mesh involving multiply-connected transition elements would require sheet refinement to remain conformal. The Selective Approach Algorithm is the implementation of the combination of element by element and sheet refinement. The Selective Approach Algorithm (as its name suggests) selects either element by element or sheet refinement for any given situation. Element by element refinement is used in all areas of the refinement region not local to multiply-connected transition elements. Sheet refinement cannot be used directly in the Selective Approach method. Directional refinement is a modification of sheet refinement and is used in the Selective Approach Algorithm in areas local to multiply-connected transition elements. In directional refinement, each inherent “direction” of a hexahedron is still refined separately like sheet refinement however the mesh is processed on a hexahedron by hexahedron basis rather than in hexahedral sheets. A ranking system and propagation scheme discussed in the body of this thesis allow directional refinement to work correctly within the Selective Approach Algorithm.

Appendix E. Templates

Templates play a key role in the Selective Approach Algorithm. This appendix will first discuss templates in general followed by a detailed description of each of the templates used in this work.

Template Characteristics

A template can be defined as a guide or a pattern to create a group of conformal elements from a single original element. A valid template is bounded completely within a single hexahedron. Each of the six faces of the original hexahedron must contain a valid face template. Examples of valid face templates are given in Figure E-1. The proper connectivity must also be maintained within the template. Hanging nodes, for example, would render a non-conformal mesh and thus the template would be invalid. The final characteristic of valid templates is that all refined elements must be hexahedra meaning they have six faces, twelve edges, and eight nodes as discussed in Appendix A.

Valid Template Creation

Valid template creation is generally governed by face templates on each of the six faces of the original hexahedron. Once the face templates are properly applied, the objective then becomes creating all-hexahedral elements within the original mesh.

Template creation is further complicated by the connectivity requirement. Together these requirements make template creation for most cases difficult at best. Figure E-2 shows a specific example of the difficulties inherent in template creation. Esmaelian [13] produced some complex templates however many of these templates created too many elements with poor quality.

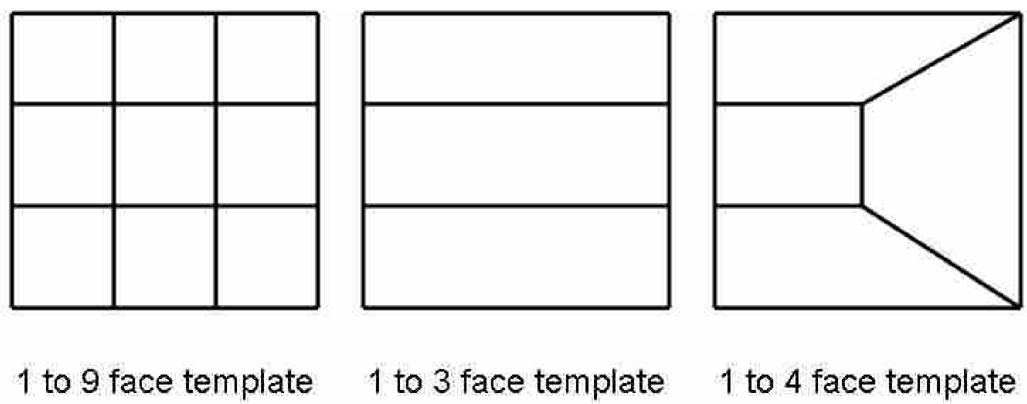


Figure E-1: Examples of valid face templates

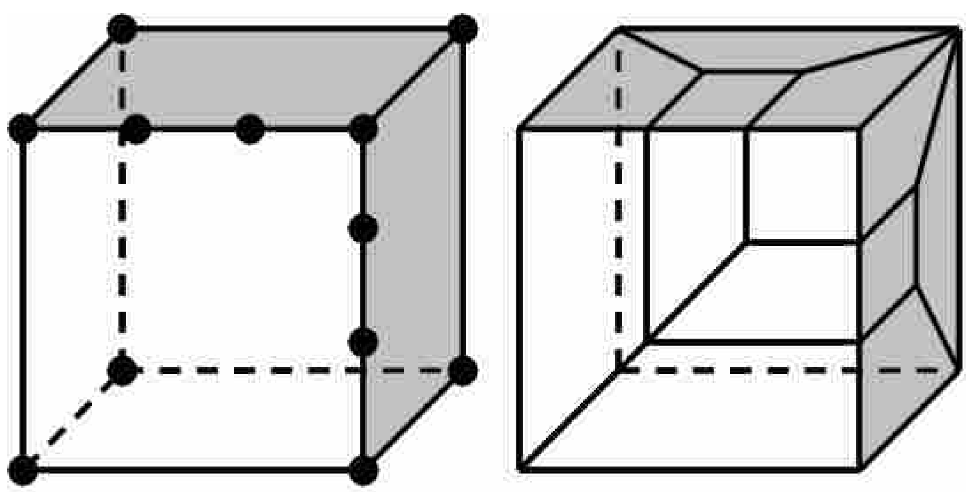
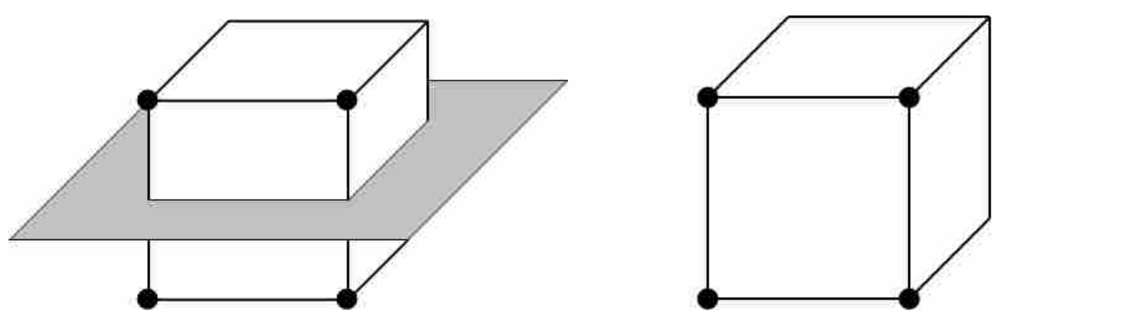


Figure E-2: Template creation - the split edges uniquely define each template (left) and the face templates are applied to the original hexahedron (right). Currently, this template cannot be created because no known configuration will satisfy the connectivity and all-hexahedral requirements in the interior of the template.

Determining Proper Template and Orientation

Selection of the proper template and correct orientation is crucial to maintain conformity in the mesh. In Harris' sheet refinement, nodes marked for refinement were used to determine the proper template and orientation [17]. Since the hexahedra are processed in hexahedral sheets, this method was satisfactory (see Figure E-3(a)). In element by element refinement, hexahedra are no longer processed in hexahedral sheets and therefore marking nodes is inadequate for the Selective Approach Algorithm (see Figure E-3(b)). Element edges must be used to uniquely define the required template of a given hexahedron. Figure E-4 shows edges that must be split to uniquely define each of the seven templates used in the Selective Approach method. Marking element edges also allows the templates to be oriented correctly. With the selection and orientation of the proper template using element edges, the Selective Approach method will produce a conformal mesh.



(a) Sheet refinement **(b) Element by element refinement**

Figure E-3: Using nodes to uniquely define required template. (a) Directional refinement uses nodes and the twist plane to uniquely define the required template. (b) Template-based refinement does not use the twist plane so uniquely defining the required template is impossible.

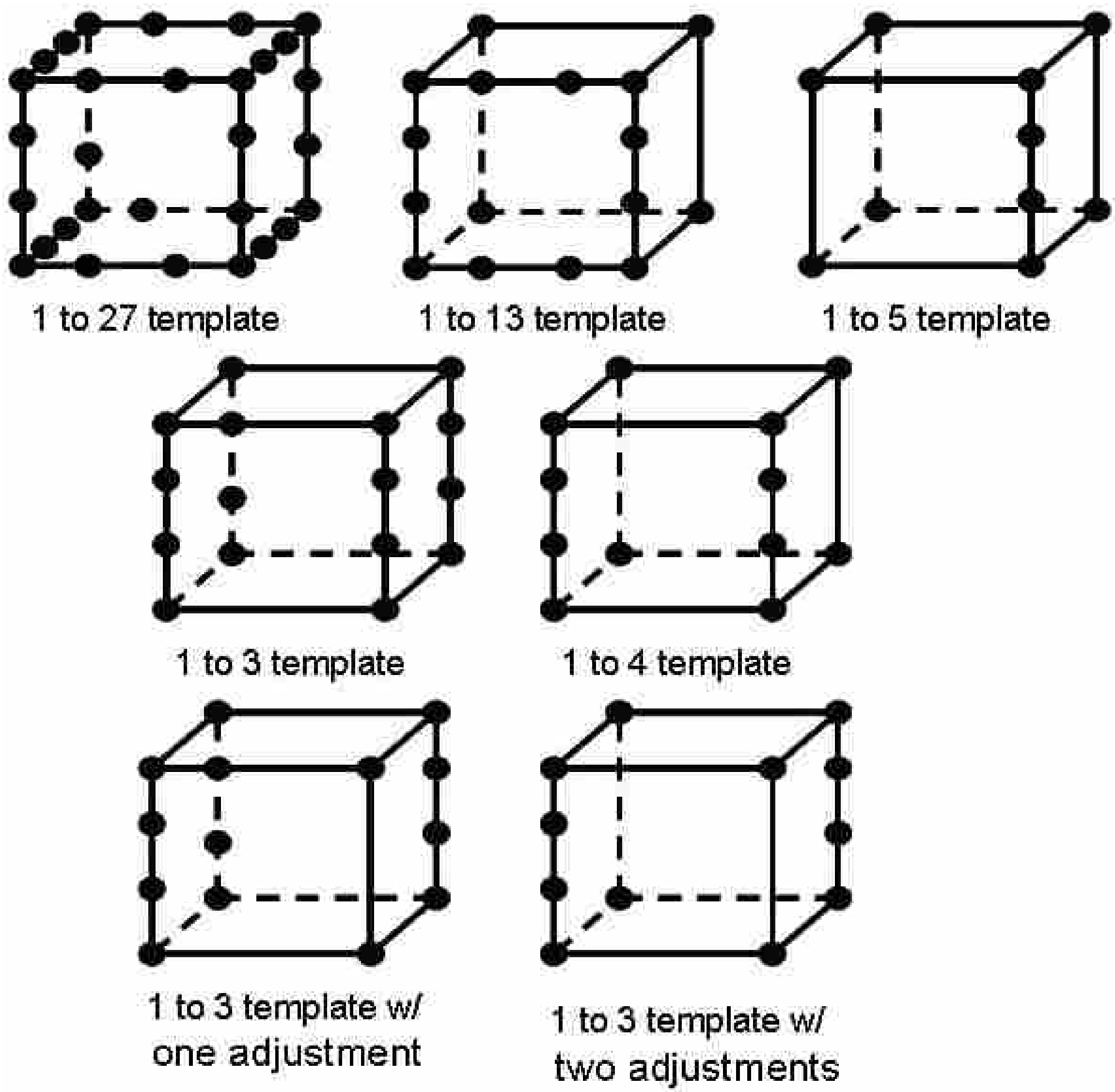


Figure E-4: Split edges that correspond to templates

1 to 27 Template

The 1 to 27 template as shown in Figure E-5 and in transparent view in Figure E-6 could be considered the standard template in the Selective Approach Algorithm. This template is applied to each target hexahedron within the refinement region. A target hexahedron is defined as any hexahedron selected by the user during the refinement process. The 1 to 27 template is only used in the element by element refinement scheme of the Selective Approach Algorithm.

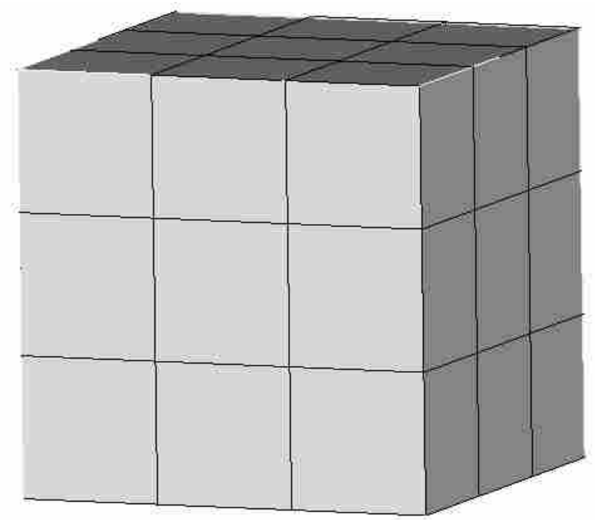


Figure E-5: 1 to 27 template

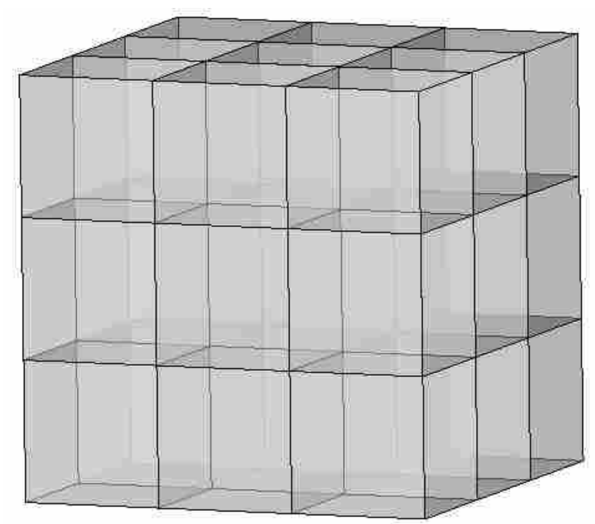
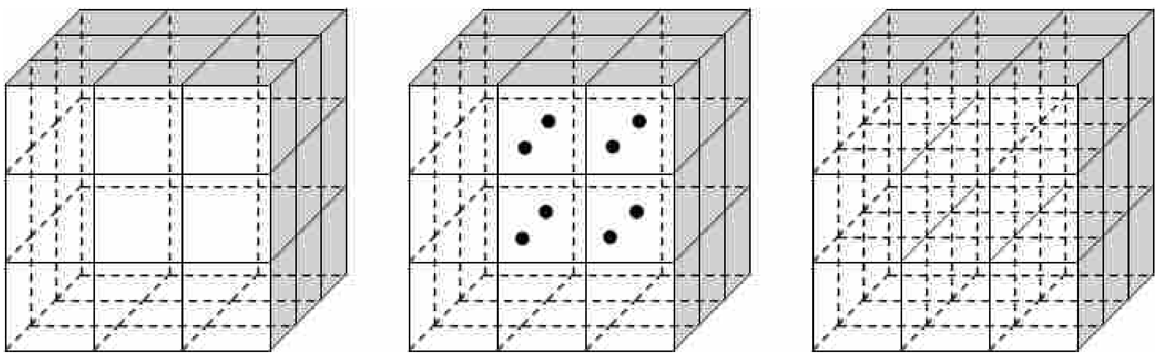


Figure E-6: 1 to 27 template (transparent view)

The 1 to 27 template is created by splitting all twelve edges of the original hexahedron. The 1 to 9 face template is applied to each of the six original faces as shown in Figure E-7(a) and eight nodes are placed within the interior of the original hexahedron as shown in Figure E-7(b). Twenty-seven hexahedra are then placed within the original hexahedron as shown in Figure E-7(c). This is by far the easiest template to visualize and understand.



(a) **(b)** **(c)**

Figure E-7: Creation of 1 to 27 template

1 to 13 Template

The standard view and transparent view of the 1 to 13 template are shown in Figure E-8 and

Figure E-9 respectively. This template was originally proposed by Schneiders in his octree-based mesh generator however slight modification has taken place since then to make this template conformal with the other templates used in this algorithm. As with the 1 to 27 template described above, the 1 to 13 template is only used in the element by element refinement scheme. This template can be applied to the transition region surrounding the target hexahedra where only four split edges exist and these four edges share a common face.

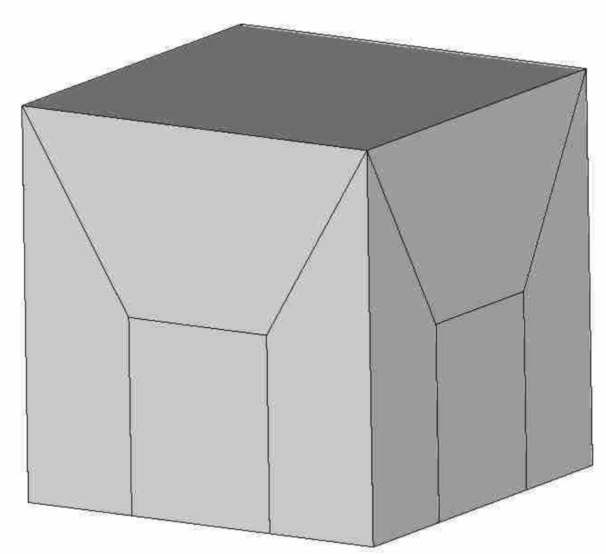


Figure E-8: 1 to 13 template

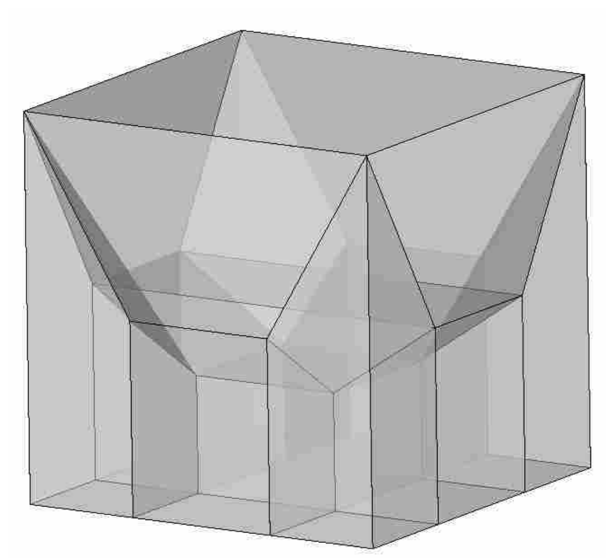
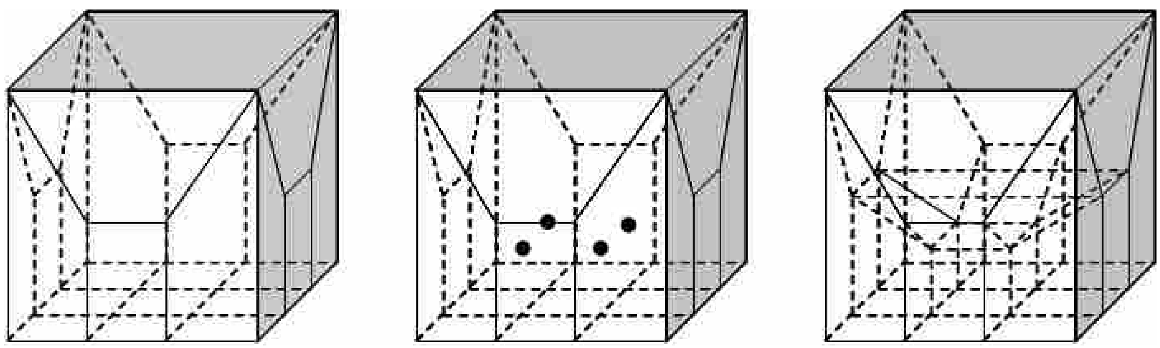


Figure E-9: 1 to 13 template (transparent view)

The 1 to 13 template can be constructed by applying the 1 to 9 face template to the bottom face, applying the 1 to 4 face template to the front, back, right, and left faces, and applying no face template to the top face (see Figure E-10(a)). Four nodes are placed within the interior of the original hexahedron closer to the bottom face (see Figure E-10(b)) and 13 hexahedra are then placed within the original hexahedron thus creating the conformal template used in the Selective Approach Algorithm (see Figure E-10(c)).



(a) **(b)** **(c)**

Figure E-10: Creation of the 1 to 13 template

1 to 5 Template

The 1 to 5 template, as shown in Figure E-11, is the first template discussed that is used in both the element by element refinement scheme and the directional refinement scheme within the Selective Approach Algorithm.

Figure E-12 depicts the 1 to 5 template in transparent view so that one may see how the template is constructed. This template is applied to the boundary layer as are all the templates except the 1 to 27 template discussed in this chapter.

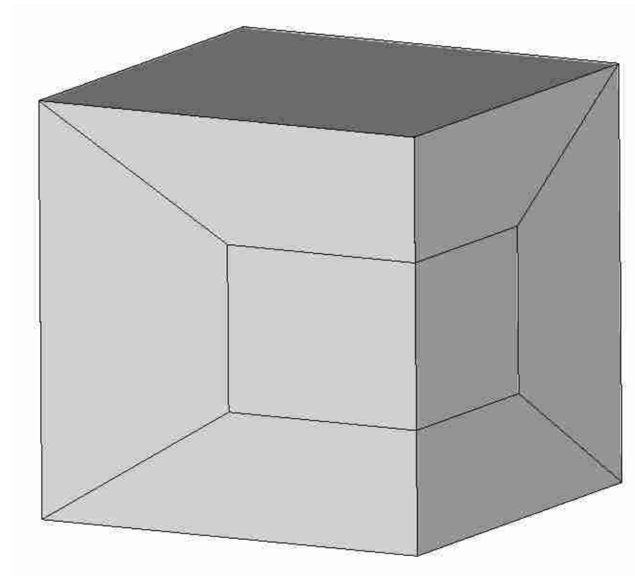


Figure E-11: 1 to 5 template

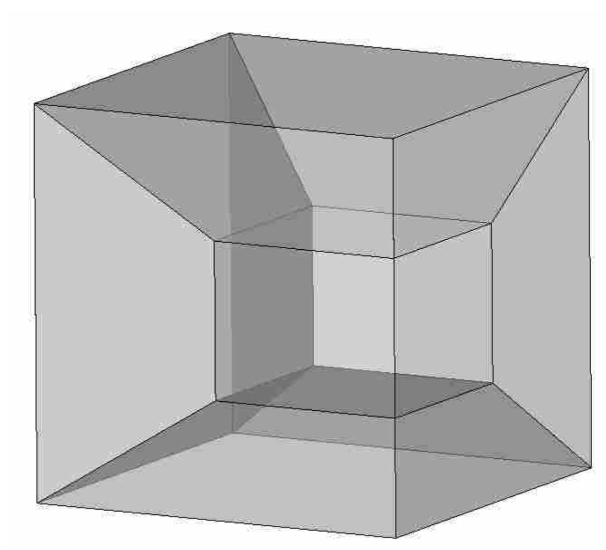
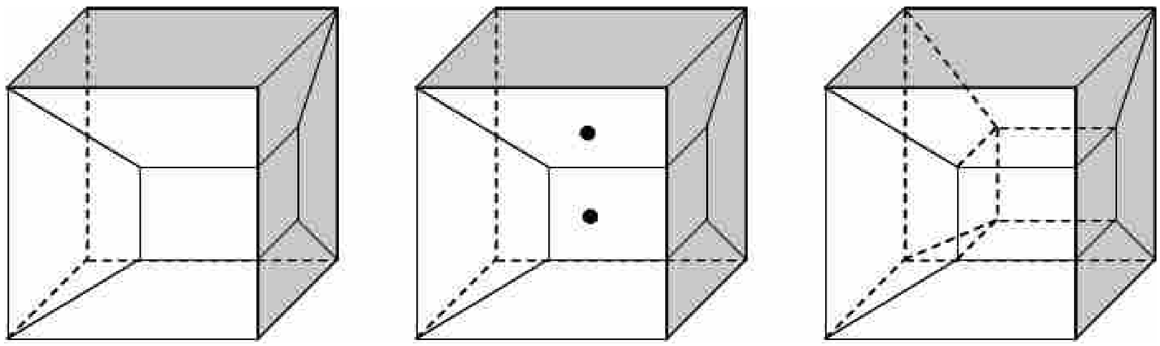


Figure E-12: 1 to 5 template (transparent view)

The 1 to 5 template can be constructed by applying the 1 to 4 face template to the front and right faces while no face template is needed for any of the other faces. This process is shown in Figure E-13(a). Two nodes are added in the interior of the original hexahedron as shown in Figure E-13(b). Finally five hexahedra are constructed creating a conformal template as shown in Figure E-13(c).



(a) (b) (c)

Figure E-13: Creation of the 1 to 5 template

1 to 4 Template

The 1 to 4 template can be used in element by element refinement; however, its primary purpose is to be a template used within directional refinement. Figure E-14 depicts the 1 to 4 template in standard view while Figure E-15 depicts the same template in transparent view. Again, this template is applied to the boundary hexahedra within the Selective Approach Algorithm.

Construction of the 1 to 4 template starts by applying the 1 to 3 face template to the front face. The 1 to 4 face template is applied to the right and the left face, and no face template is needed for the remaining faces. This step is shown in Figure E-16(a). No interior nodes are required to create this template differing from the templates discussed above. Four hexahedra are added to connect the face templates as shown in Figure E-16(b).

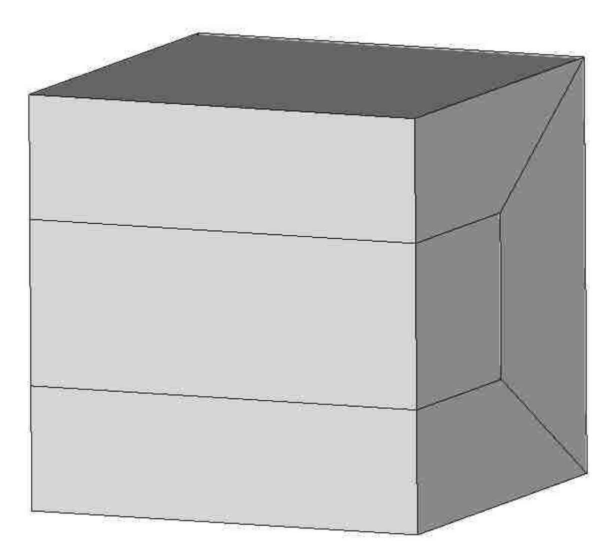


Figure E-14: 1 to 4 template

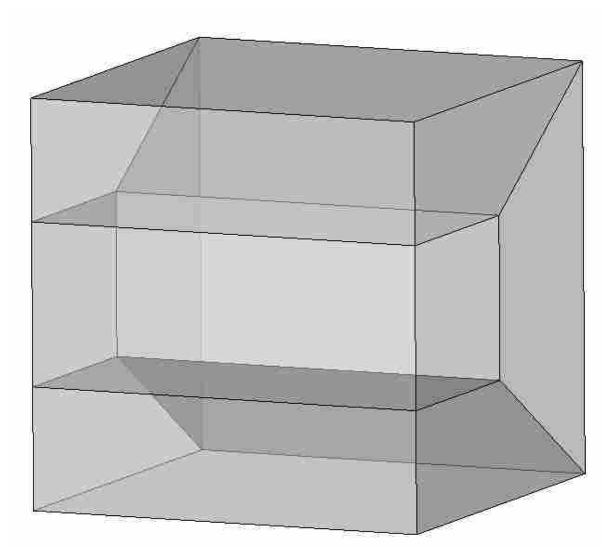
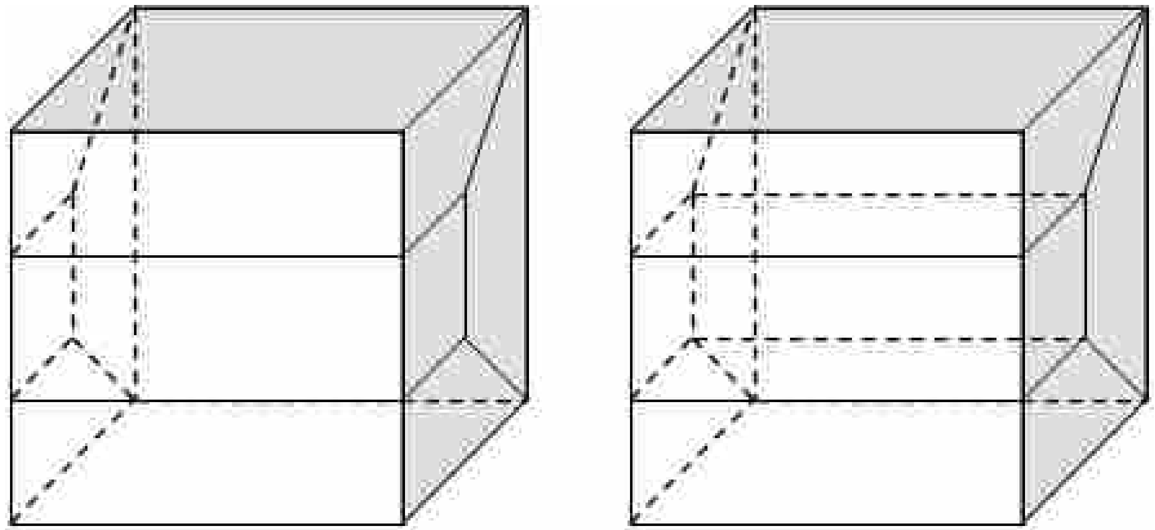


Figure E-15: 1 to 4 template (transparent view)



(a) (b)

Figure E-16: Creation of the 1 to 4 template

1 to 3 Template

The 1 to 3 template is another template that is easy to visualize. Figure E-17 shows the template in standard view and Figure E-18 shows the template in transparent view. Again, this template is used primarily in directional refinement however it will be used in the element by element refinement scheme on occasion.

The 1 to 3 template is constructed by placing the 1 to 3 face template on the front, back, right, and left faces of the template. No face template is required for the top and bottom faces. Also, as with the 1 to 4 template, no interior nodes are needed to create the template. The last step is to place three hexahedra within the original hexahedron as shown in the transparent view of the template.

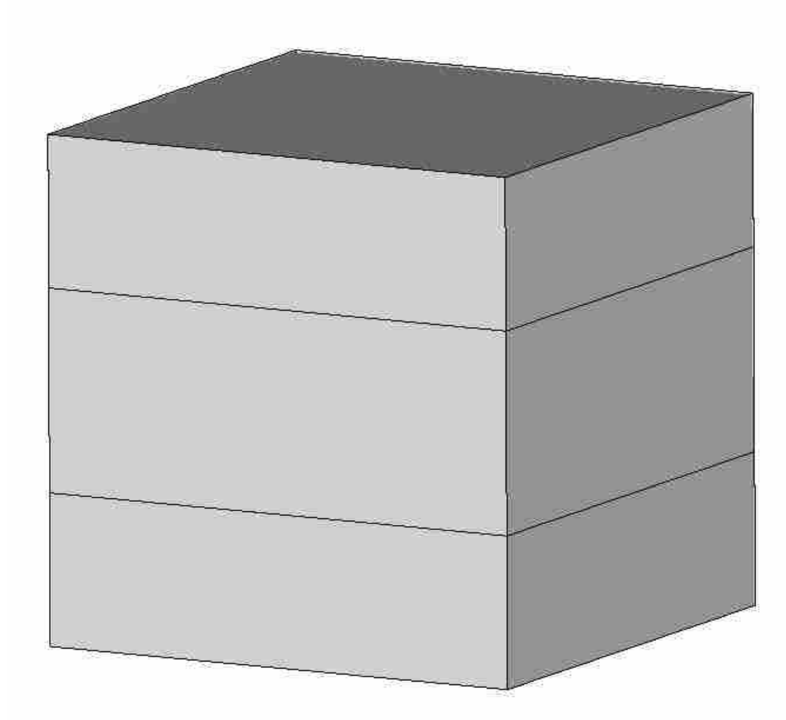


Figure E-17: 1 to 3 template

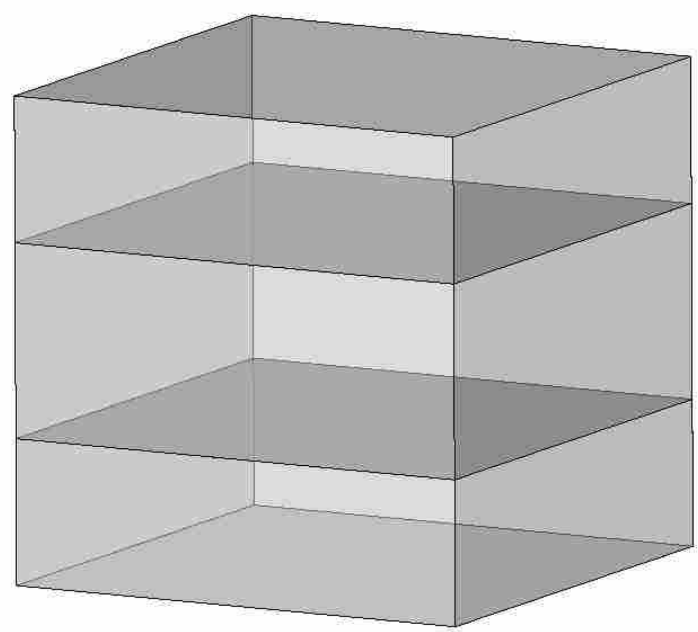


Figure E-18: 1 to 3 template (transparent view)

1 to 3 Template with One Adjustment

The remaining two templates are used in areas local to multiply-connected transition elements. The 1 to 3 template with one adjustment is primarily used in the directional refinement scheme; however, it is also used on occasion in element by element refinement. The 1 to 3 template with one adjustment is shown in standard view in Figure E-19 and transparent view in Figure E-20.

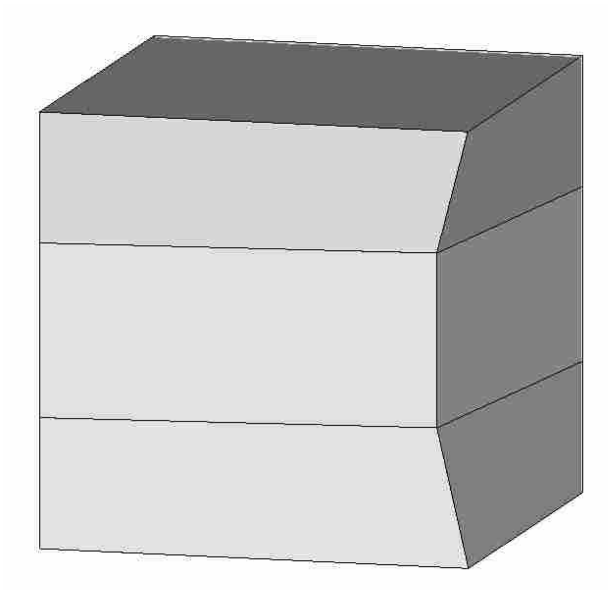


Figure E-19: 1 to 3 template with one adjustment

The 1 to 3 template with one adjustment is constructed by first applying the 1 to 4 face template to the front and right faces. The 1 to 3 face template is applied to the back and left faces. This face template configuration forces the inserted twist plane to self-intersect within the template. Such a template cannot be constructed with reasonable quality. To accommodate this situation, the template is adjusted as shown in Figure

E-21. This adjustment allows the multiply-connected transition element to be handled properly creating a conformal mesh while maintaining a reasonable element quality.

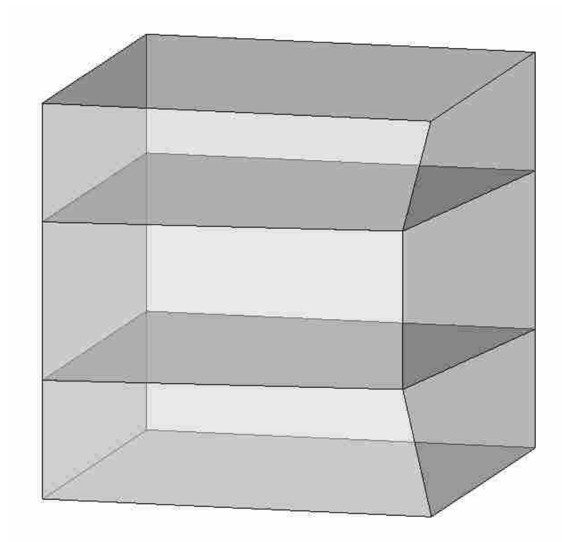


Figure E-20: 1 to 3 template with one adjustment (transparent view)

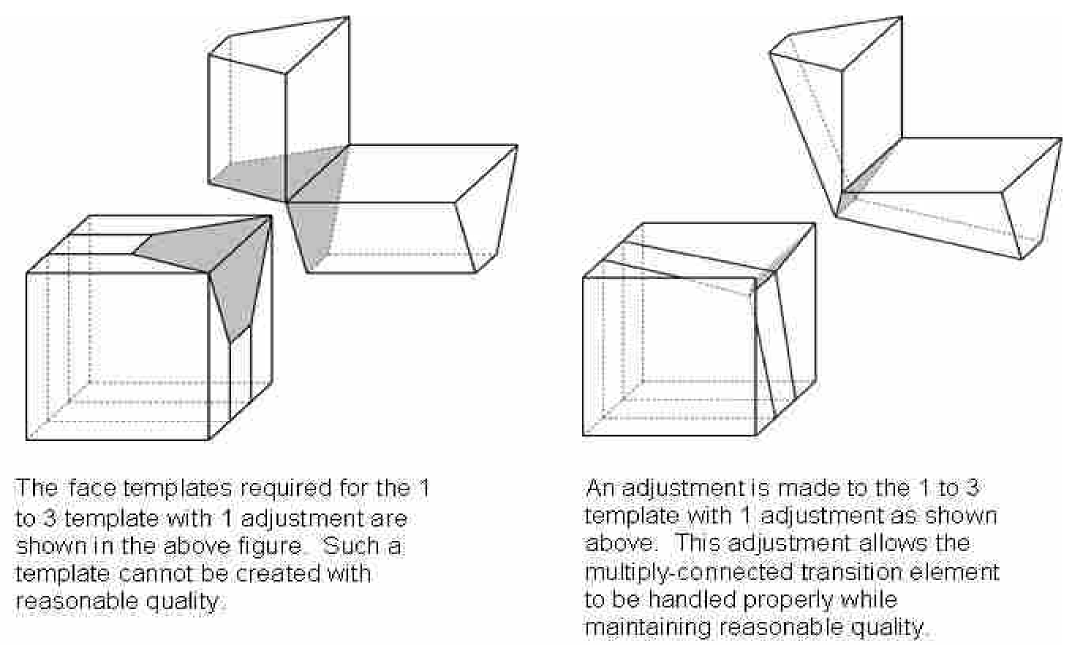


Figure E-21: Multiply-connected transition element adjustment

1 to 3 Template with Two Adjustments

The 1 to 3 template with two adjustments is very similar to the 1 to 3 template with one adjustment except an adjustment also exists between the left and back faces as well as between the front and right faces. Figure E-22 shows the standard view of the 1 to 3 template with two adjustment.

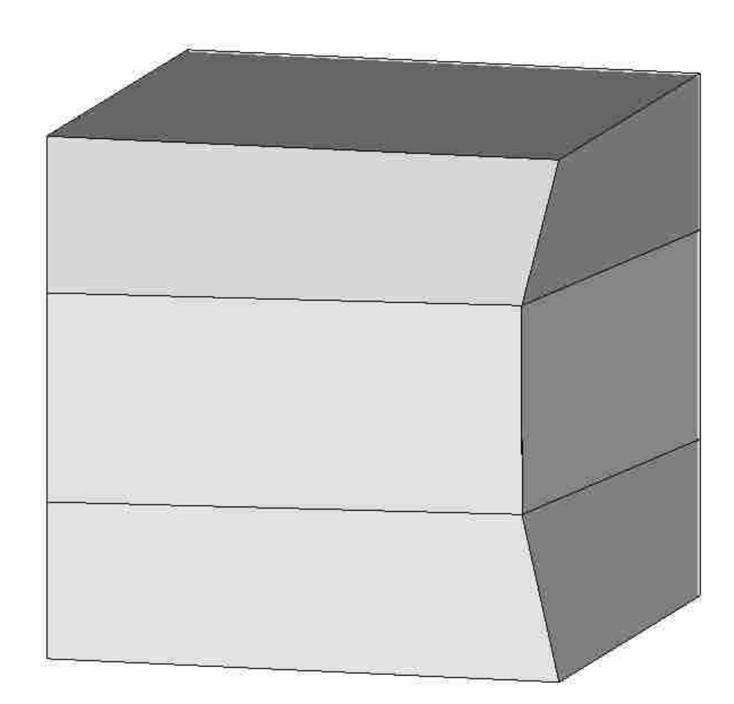


Figure E-22: 1 to 3 template with two adjustments

Construction of the 1 to 3 template with two adjustments begins by applying the 1 to 4 face template to the front, back, right, and left faces. No face templates are necessary for the top and bottom faces. The same adjustment is made for this template though this time the adjustment is made both in the front right corner and the back left corner. Three hexahedra are created within the template and the resulting template is shown in Figure E-23, this time in transparent view.

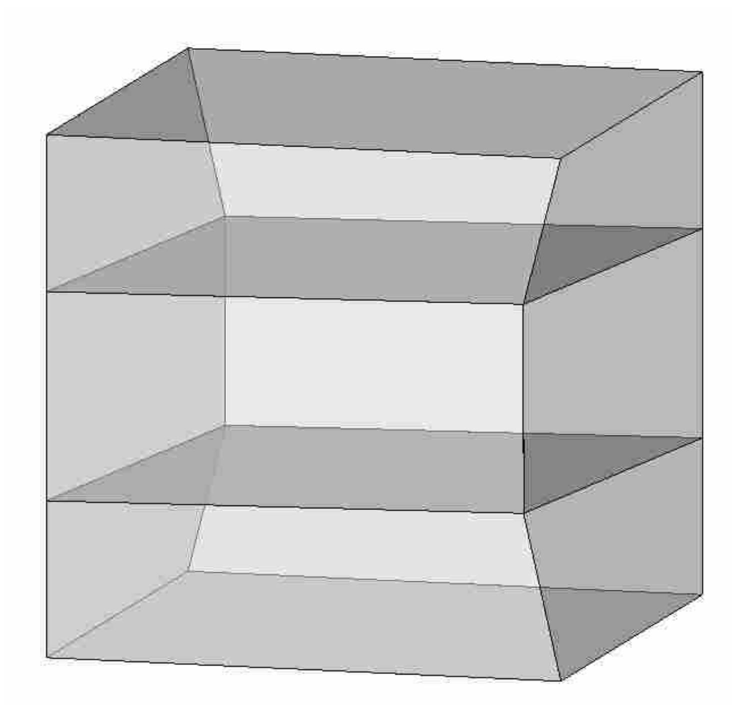


Figure E-23: 1 to 3 template with two adjustments (transparent view)

Appendix F. The Doublet Problem

The creation of doublets during the refinement process is a major concern for any hexahedral refinement scheme. This appendix will discuss some of the key issues involved in hexahedral refinement where doublets could potentially be created.

Definition of a Doublet

In two dimensions, a doublet is defined as two quadrilateral faces that share two edges [18]. In three dimensions, a doublet occurs where two hexahedra share two faces (see Figure F-1). A doublet in three dimensions can also be viewed as a pair of two-dimensional doublets.

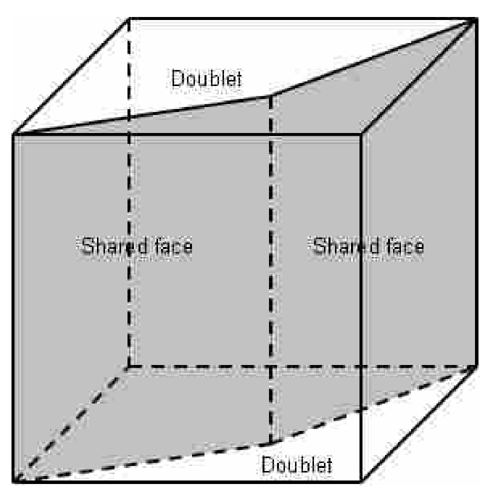


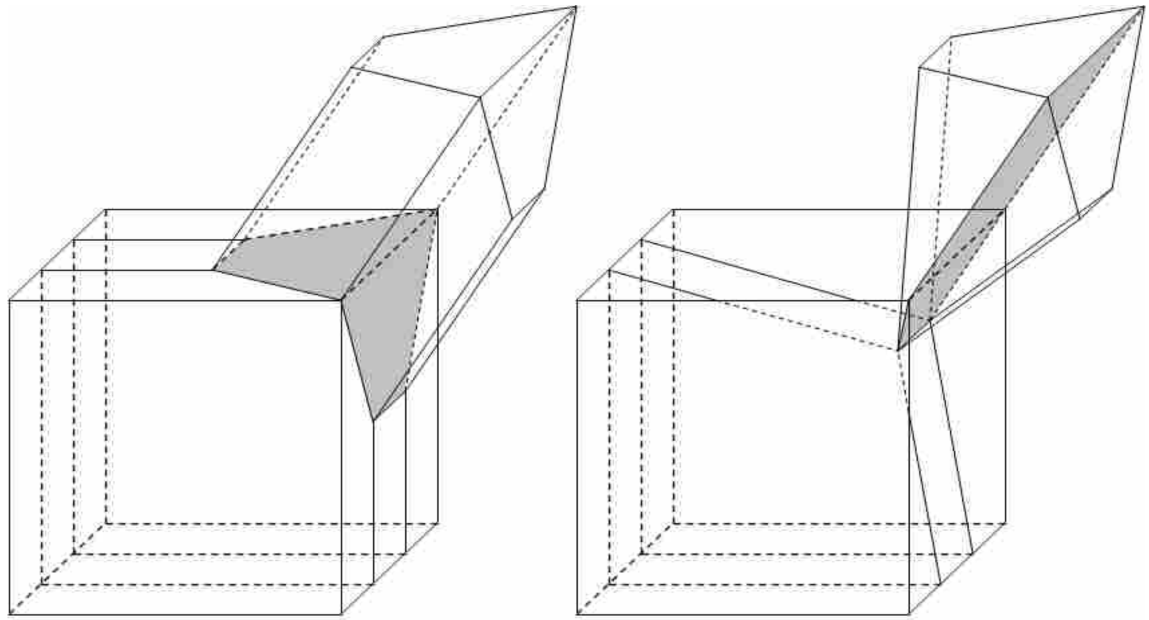
Figure F-1: Two hexahedra sharing two faces implies two doublets

While the connectivity of a doublet is valid, it requires that one of the hexahedra that form the doublet be inverted. An inverted element results in a poor quality mesh unsuitable for an analysis.

Doublets in Hexahedral Refinement

The adjustment templates discussed in Appendix E can potentially create doublets in the Selective Approach Algorithm. This may occur because these adjustment templates essentially collapse two faces into one. If the adjacent hexahedra already share a face, this collapsing of faces ensures that these hexahedra will share two faces which by definition is a doublet. Figure F-2 graphically illustrates how a doublet can form when using the adjustment templates.

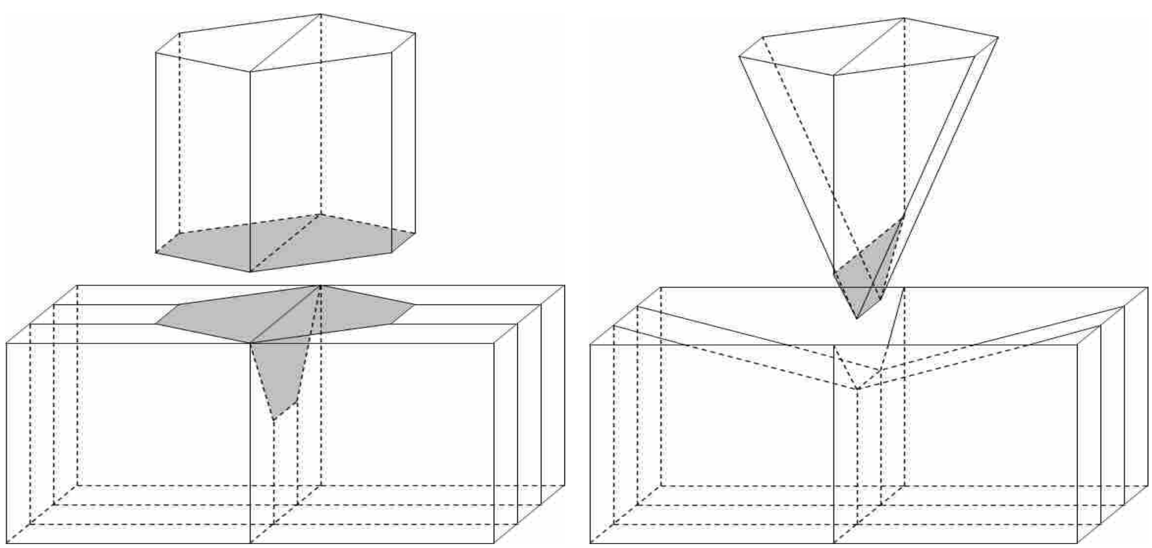
A variant of the doublet problem discussed above arises when two 1 to 3 templates with one adjustment share a common face. This configuration would collapse three faces into one. If the adjacent hexahedra already share a face, a doublet will be created. Figure F-3(a) depicts these two 1 to 3 templates with one adjustment. The adjacent hexahedra are also shown. Initially, only the face templates have been applied while the adjustment and resulting hexahedra have been excluded. Figure F-3(b) shows the same two templates after the adjustment has been applied. Notice that while the hexahedra inside each of these templates are fine, the adjacent hexahedra now share two common faces thus creating a doublet.



The 1 to 3 template with 1 adjustment is shown where only the face templates have been applied. The shaded region indicates the faces that will be collapse together. The adjacent hexahedra are also shown and already share a face

The adjustment is made and the faces are collapsed on themselves. The shaded region indicates the two faces shared by the adjacent hexahedra because they already shared a face before the adjustment was made.

Figure F-2: Doublet problem



(a) **(b)**

Figure F-3: 1 to 3 templates that share a common face

Doublet Resolution

Since doublets make a mesh unsuitable for an analysis, the Selective Approach Algorithm must be able to determine where doublets might occur and then resolve these issues by altering the mesh. It was shown above that doublets can occur whenever the hexahedra adjacent to an adjustment template share a face. This criterion can be generalized in that when an edge is only shared by three faces, no adjustment template can be applied to any of the surrounding hexahedra where the adjustment is done at the edge otherwise doublets will be created. The previous statement also indicates the general method for resolving these doublet issues once they have been detected. An adjustment template may not be used. Therefore, the edge must be split in order to remove the need for an adjustment template. Figure F-4 graphically shows how the doublet problem is resolved. Since the edge is split before templates are applied to the original mesh, the adjacent hexahedra no longer exist. Different templates must be applied to all the hexahedra that share the split edge.

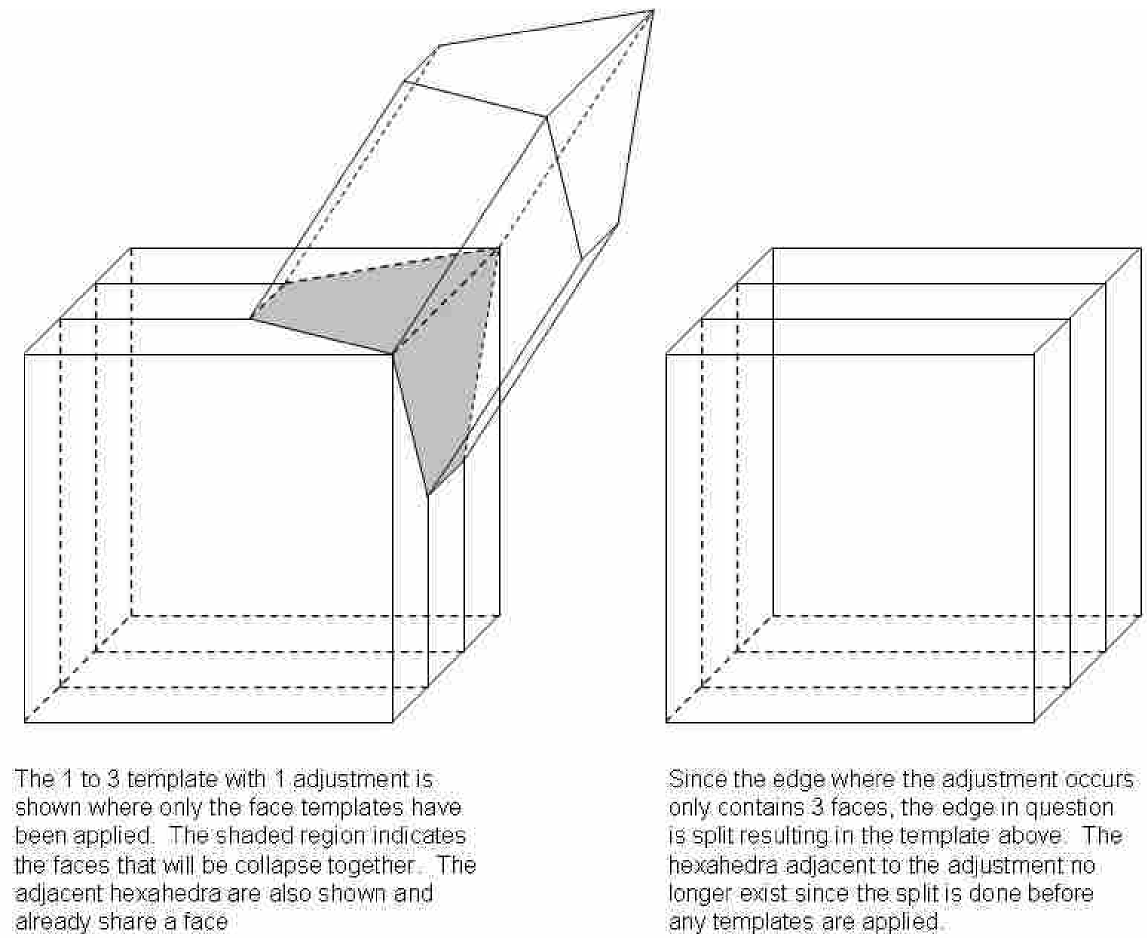


Figure F-4: Doublet resolution

Appendix G. Results

Chapter 6 gives a detailed discussion of the results of this work. This appendix contains supporting data and enlarged images to clarify the results of this work.

Multiply-Connected Transition Elements

While an example was given in the body of this paper, another example of the Selective Approach Algorithm's ability to effectively handle multiply-connected transition elements is given here. Figure G-1 is a simple meshed brick where the user desires to refine the left and bottom surfaces. Figure G-2 depicts the brick after refinement has occurred using the sheet refinement algorithm implemented by Harris. Notice that in order to remove the multiply-connected transition elements from the mesh, the entire brick is refined. Figure G-3 is the same meshed brick refined using the Selective Approach Algorithm. Because the Selective Approach Algorithm uses the 1 to 3 template with one adjustment and the 1 to 3 template with two adjustments, no hexahedra must be added to the refinement region. The mesh is still conformal and provides the result the user intended.

Table G-1 gives the numerical results for each refinement scheme. The Selective Approach Algorithm had a final element count of only 8,060 hexahedra while Harris' sheet refinement had 27,000 elements. The time required to perform the refinement was much lower for the Selective Approach Algorithm, only requiring 0.797 seconds. Without applying smoothing, the minimum quality of the sheet refinement method was better. However, this is because the entire brick was refined to remove the multiply-connected transition elements from the refinement region. The minimum quality of the Selective Approach Algorithm without smoothing is still adequate for an analysis and the benefits of the Selective Approach Algorithm far outweigh the reduction in quality.

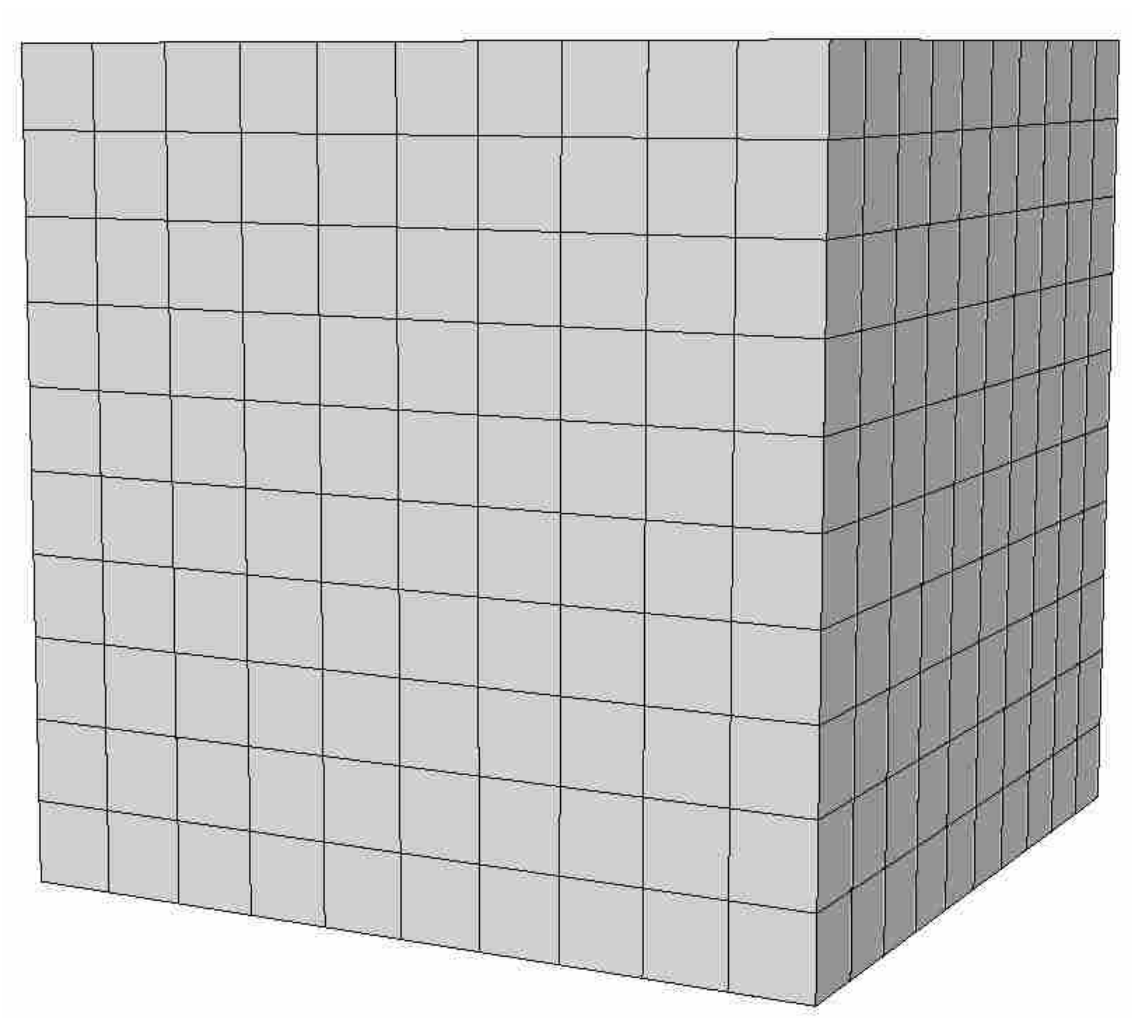


Figure G-1: Simple meshed brick where left and bottom faces must be refined

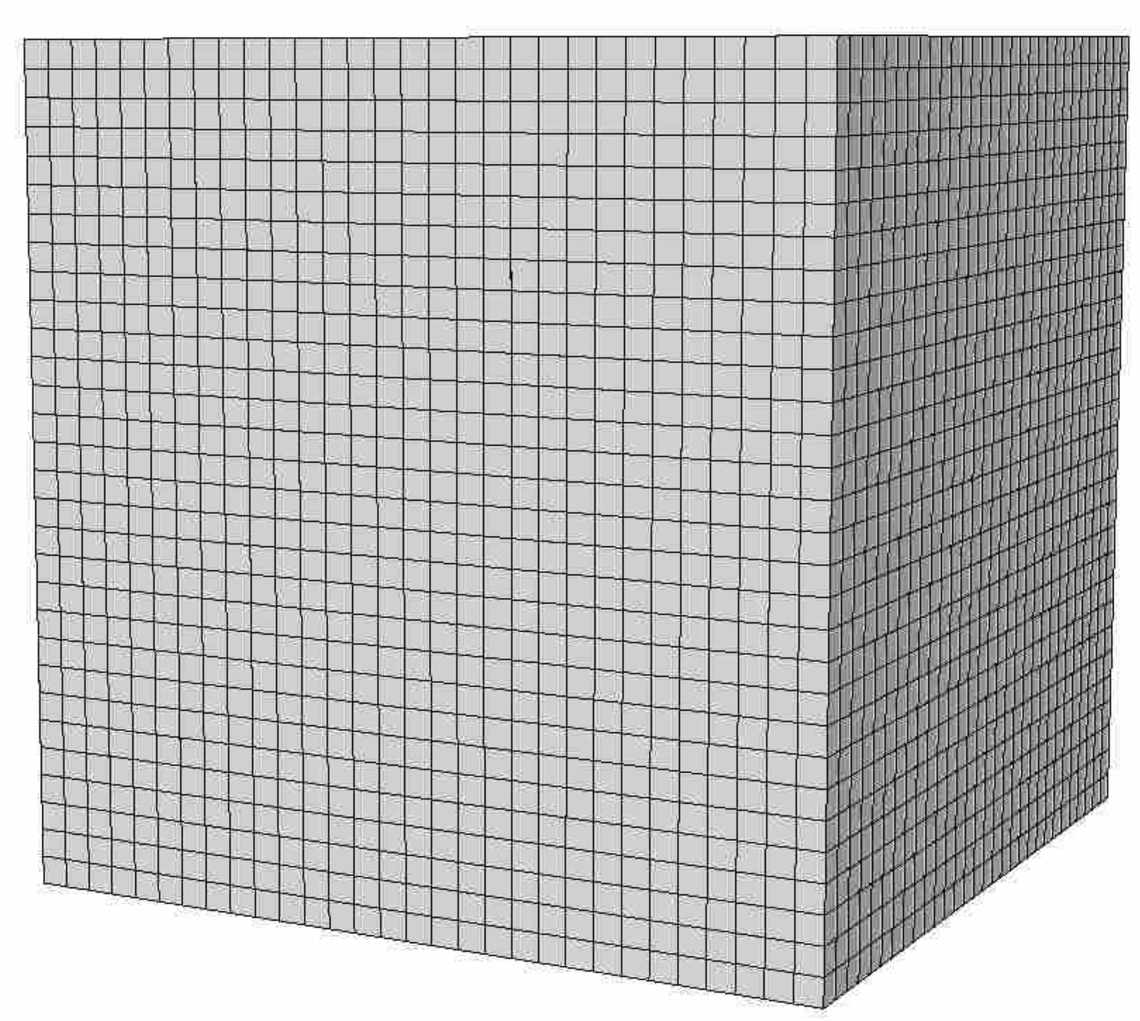


Figure G-2: Refined brick (sheet refinement)

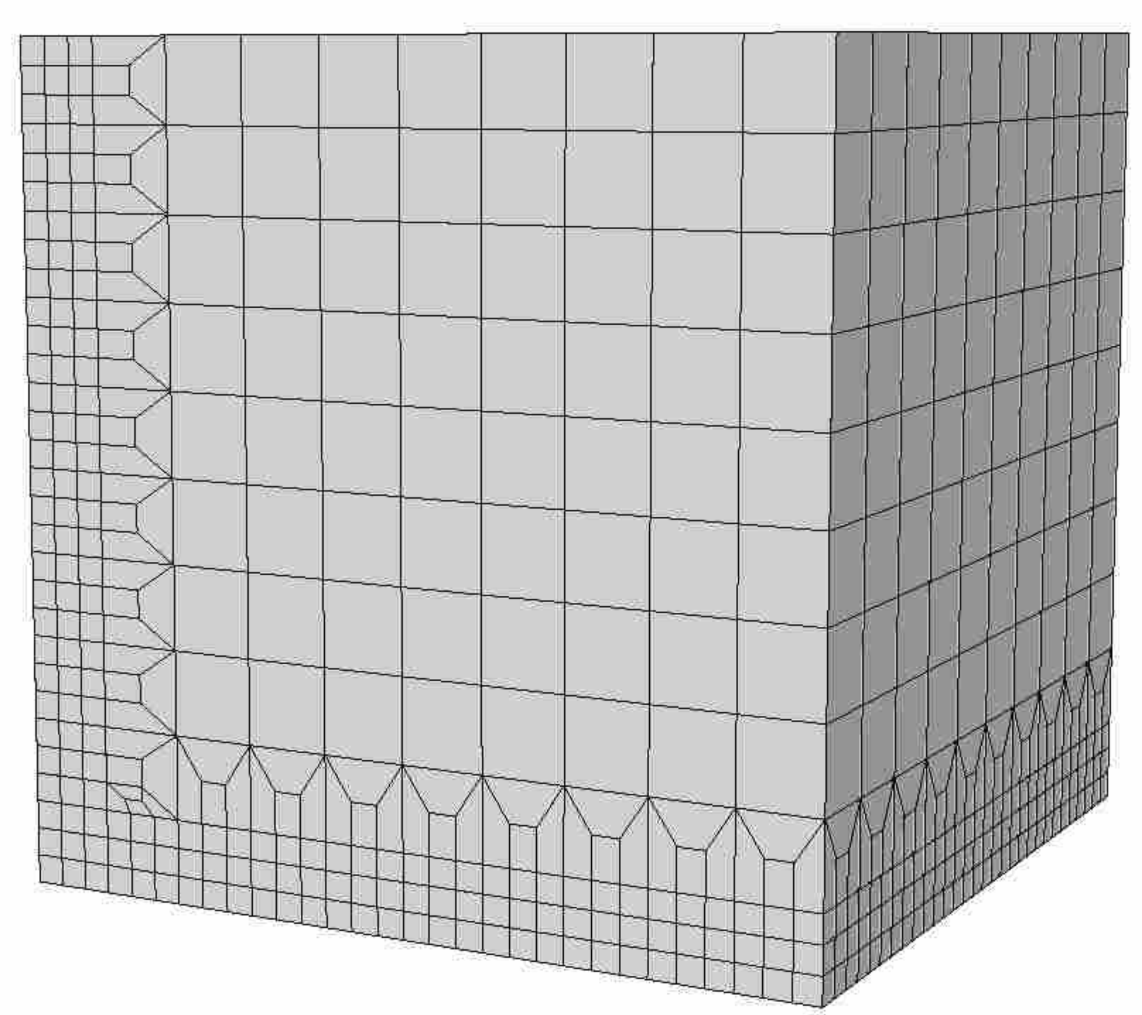


Figure G-3: Refined brick (Selective Approach Algorithm)

Table G-1: Measurements of sheet refinement and the Selective Approach Algorithm

Measurement	Sheet Refinement	Selective Approach
Initial Element Count	1000	1000
Final Element Count	27000	8060
Time (sec)	6.484	0.797
Initial Minimum Quality	1.0	1.0
Final Minimum Quality	1.0	0.3077

Scalability

The majority of the results concerning scalability were given in the body of this work. Table G-2 shows the actual time values for each run of both refinement schemes in the first scalability analysis described herein. This analysis involved increasing the interval count of a simple brick and measuring how long it took both refinement schemes to run. Figure G-4 and Figure G-5 are plots of those values comparing the Selective Approach Algorithm with Harris' sheet refinement scheme. Figure G-6, Figure G-7, and Table G-3 are the results from the second scalability analysis described in the body of this thesis. This analysis involved increasing the interval count of a simple brick. The refinement region was specified as all hexahedra within a constant radial distance from the top front right vertex of the brick. Refining this region required some directional refinement to occur within the Selective Approach Algorithm.

Table G-2: Recorded time (sec) for each refinement scheme as number of initial elements is increased

Interval	Element Count	Selective Approach	Sheet Refinement
0	0	0	0
5	125	0.04	0.12
10	1000	0.27	1
15	3375	1	5.2
20	8000	2.45	19.25
25	15625	4.94	60.88
30	27000	8.68	175.3
35	42875	17.27	549.59
40	64000	26.7	1521
45	91125	34.75	3526.25
50	125000	51.26	6967.52
55	166375	71.99	12621.43
60	216000	106.05	21640.71
65	274625	149.83	36485.14
70	343000	204.84	62529.3
75	421875	284.37	82962.49

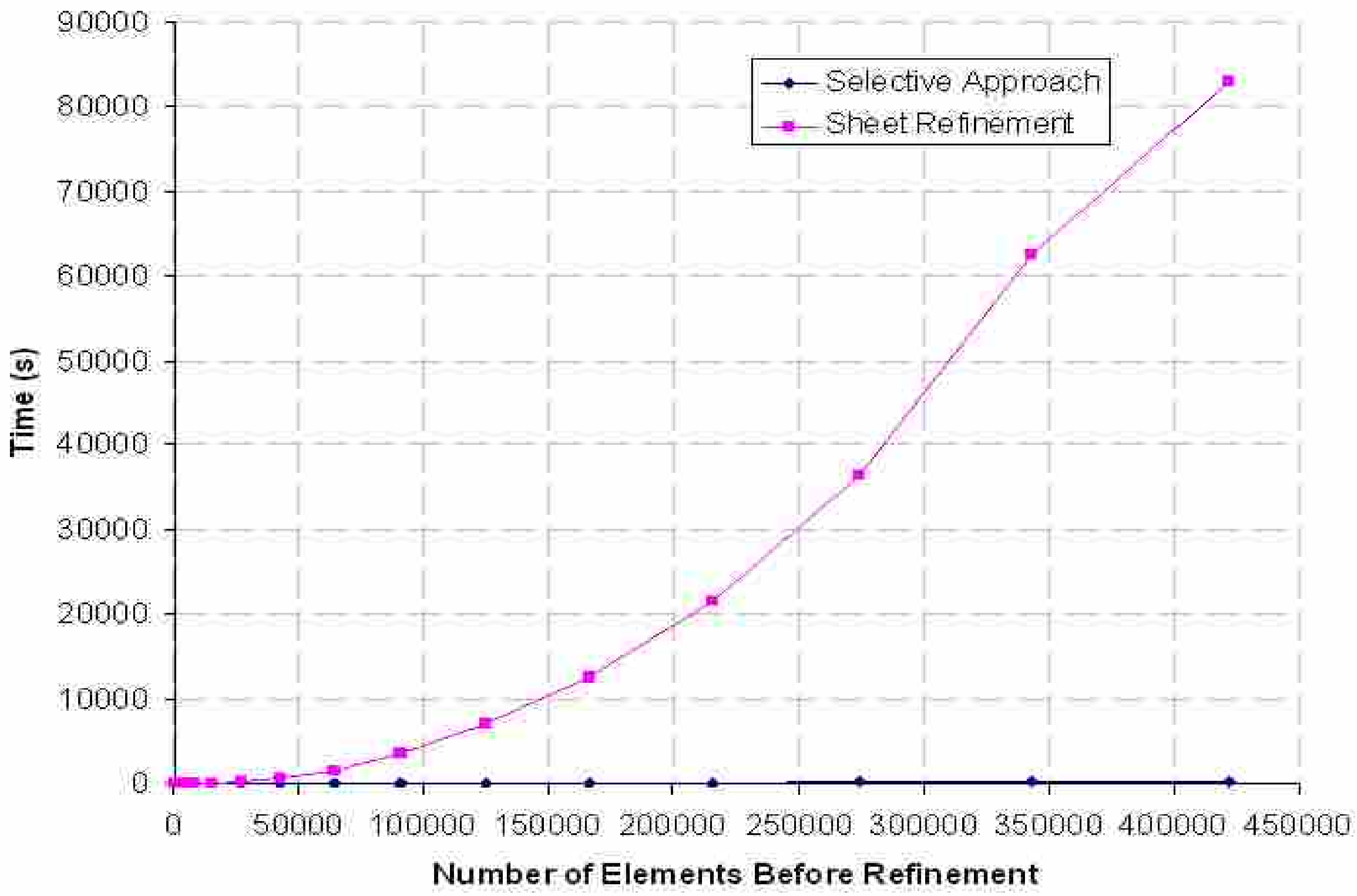


Figure G-4: Scalability comparison between Harris' sheet refinement and the Selective Approach Algorithm

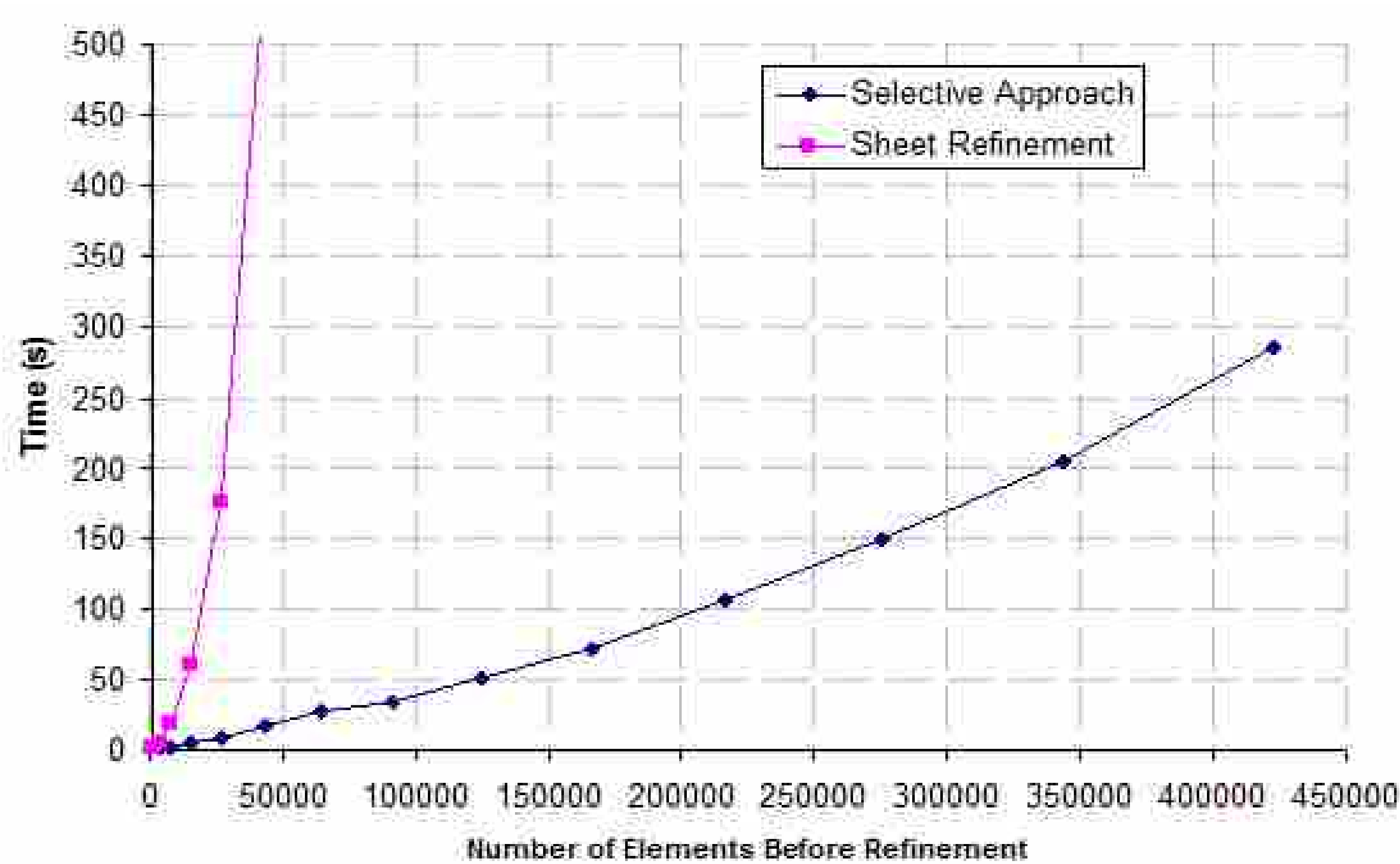


Figure G-5: Scalability comparison between Harris' sheet refinement and the Selective Approach Algorithm (y-axis reduced)

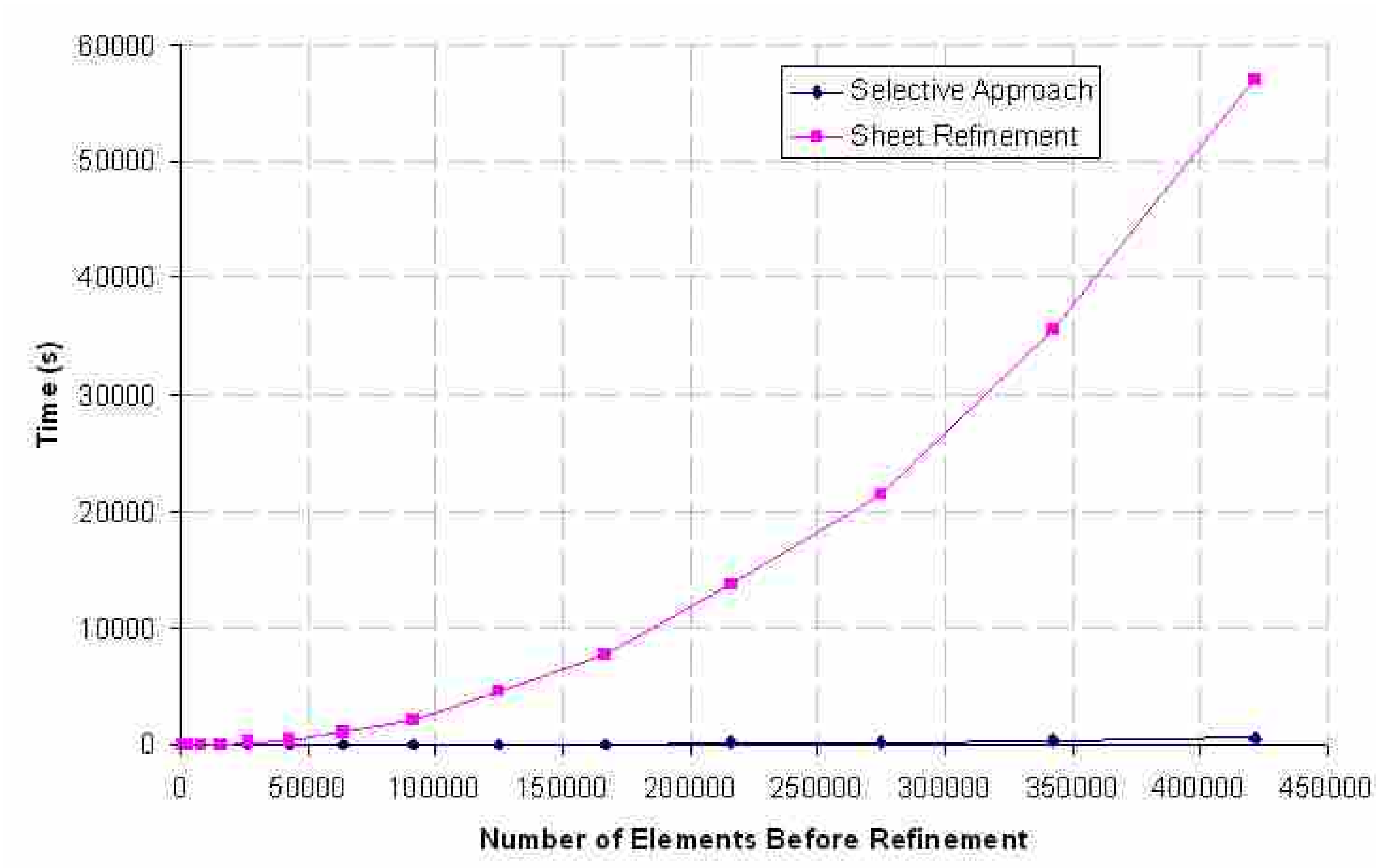


Figure G-6: Scalability comparison between Harris' sheet refinement and the Selective Approach Algorithm with directional refinement

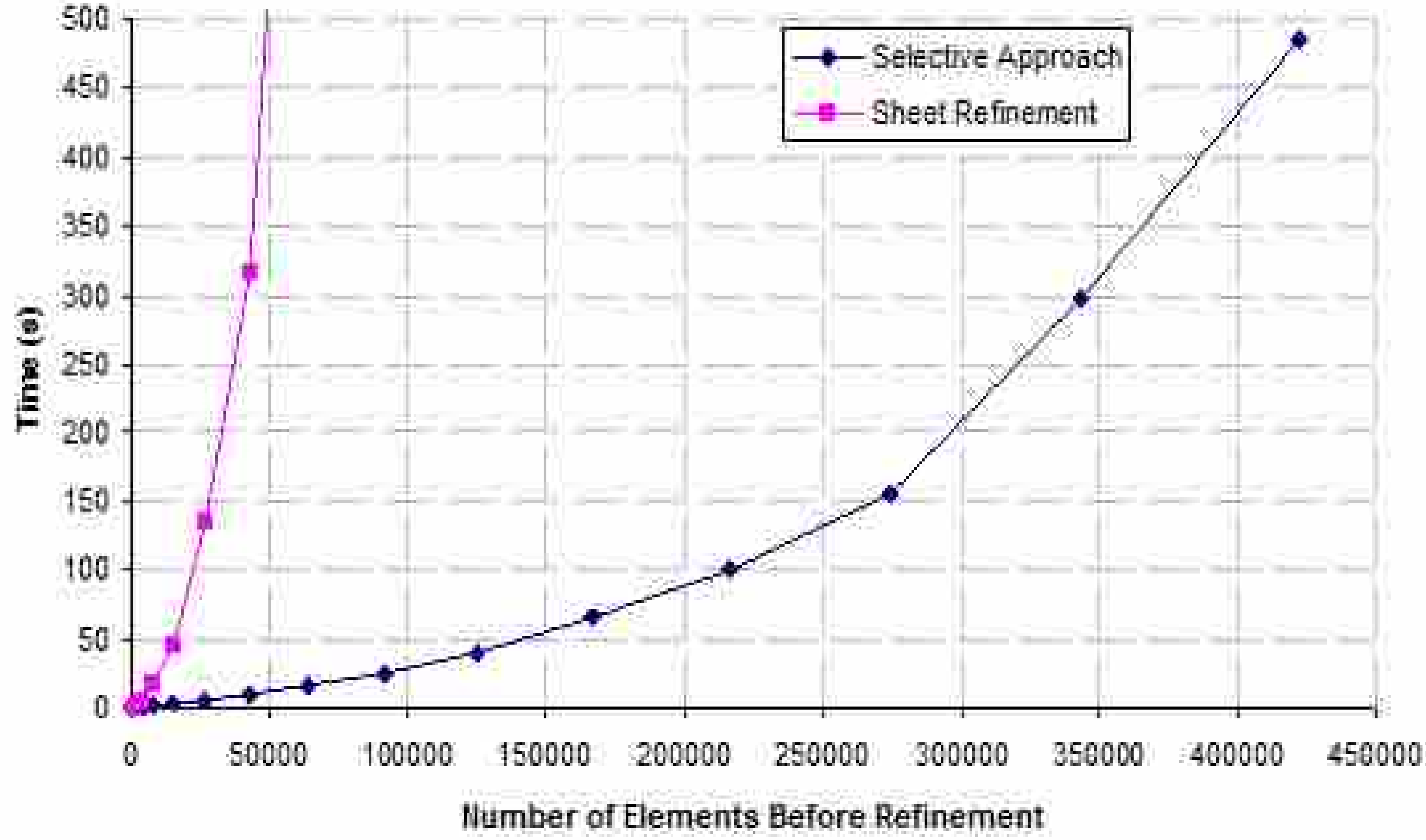


Figure G-7: Scalability comparison between Harris' sheet refinement and the Selective Approach Algorithm with directional refinement (y-axis reduced)

Table G-3: Recorded time (sec) for each refinement scheme as number of initial elements is increased (Selective Approach Algorithm includes some directional refinement)

Interval	Element Count	Selective Approach	Sheet Refinement
0	0	0	0
5	125	0.04	0.12
10	1000	0.25	1.03
15	3375	1	4.44
20	8000	1.68	17.72
25	15625	3.32	45.87
30	27000	6.15	135.08
35	42875	9.91	316.78
40	64000	17.01	960.94
45	91125	25.3	2056.72
50	125000	41.16	4607.22
55	166375	65.2	7547.38
60	216000	100.38	13664.87
65	274625	155.44	21381.45
70	343000	297.72	35530.61
75	421875	484.95	57019.17

Appendix H. Examples

A single example of the Selective Approach Algorithm was given in the body of this thesis. The Appendix contains four more examples each showing the robust capabilities of the Selective Approach Algorithm. For all examples, Harris' sheet refinement was used for comparison with the Selective Approach Algorithm.

Gear Example

The first example is the model of a gear as shown in Figure H-1. This model has been meshed with an all-hexahedral mesh and contains 8568 elements. Each of the individual teeth could be of interest in a stress analysis. Figure H-2 is a close up of a section of the gear.

Figure H-1: Gear model

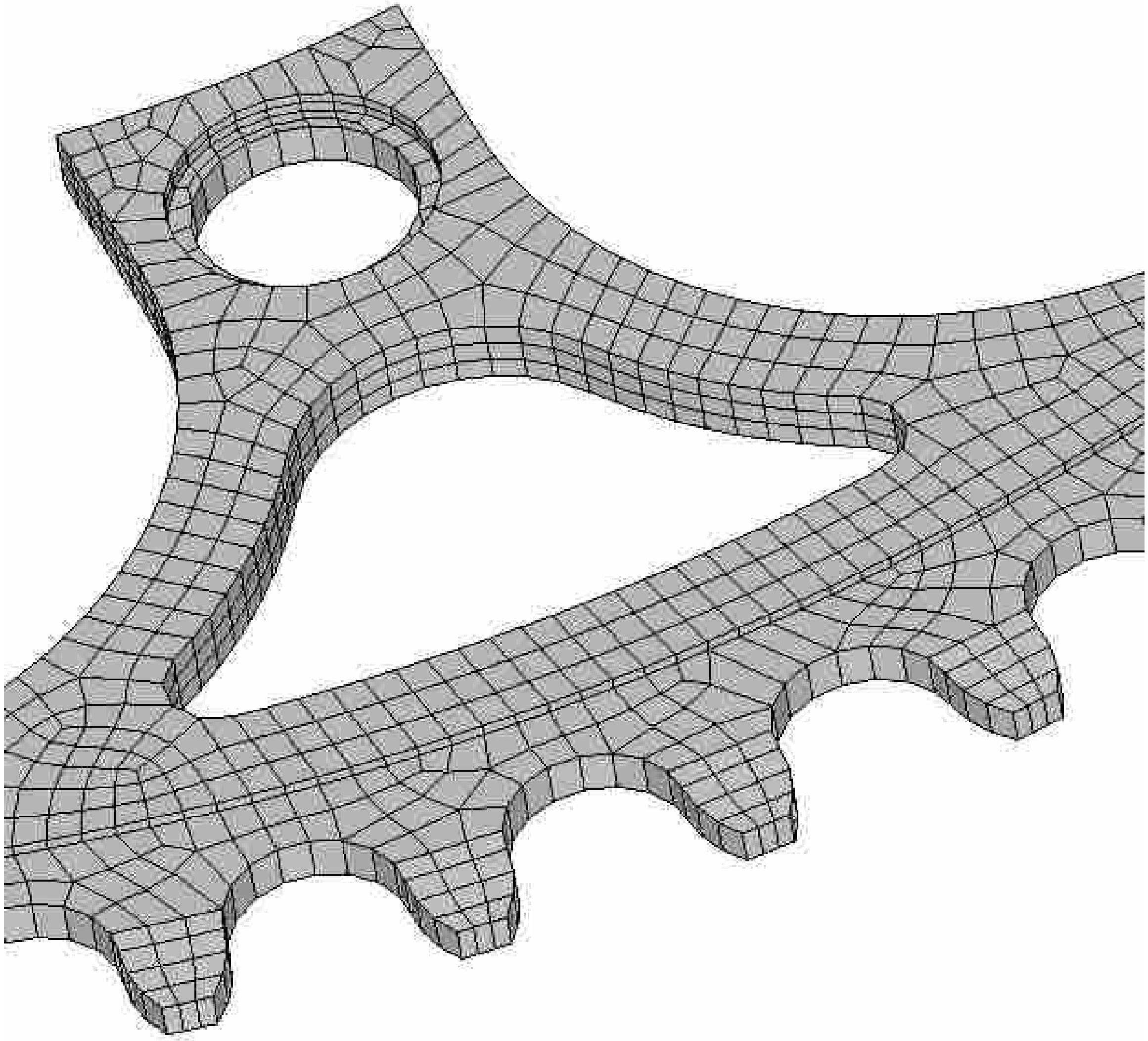


Figure H-2: Close up of gear

To improve the potential numerical accuracy of a stress analysis, the teeth of the gear are refined, thus increasing the density of the mesh. Both the sheet refinement scheme implemented by Harris and the Selective Approach Algorithm were used to refine the teeth of the gear. Since the refinement region did not contain any multiply-connected transition elements, the resulting mesh is the same for both methods (see Figure H-3).

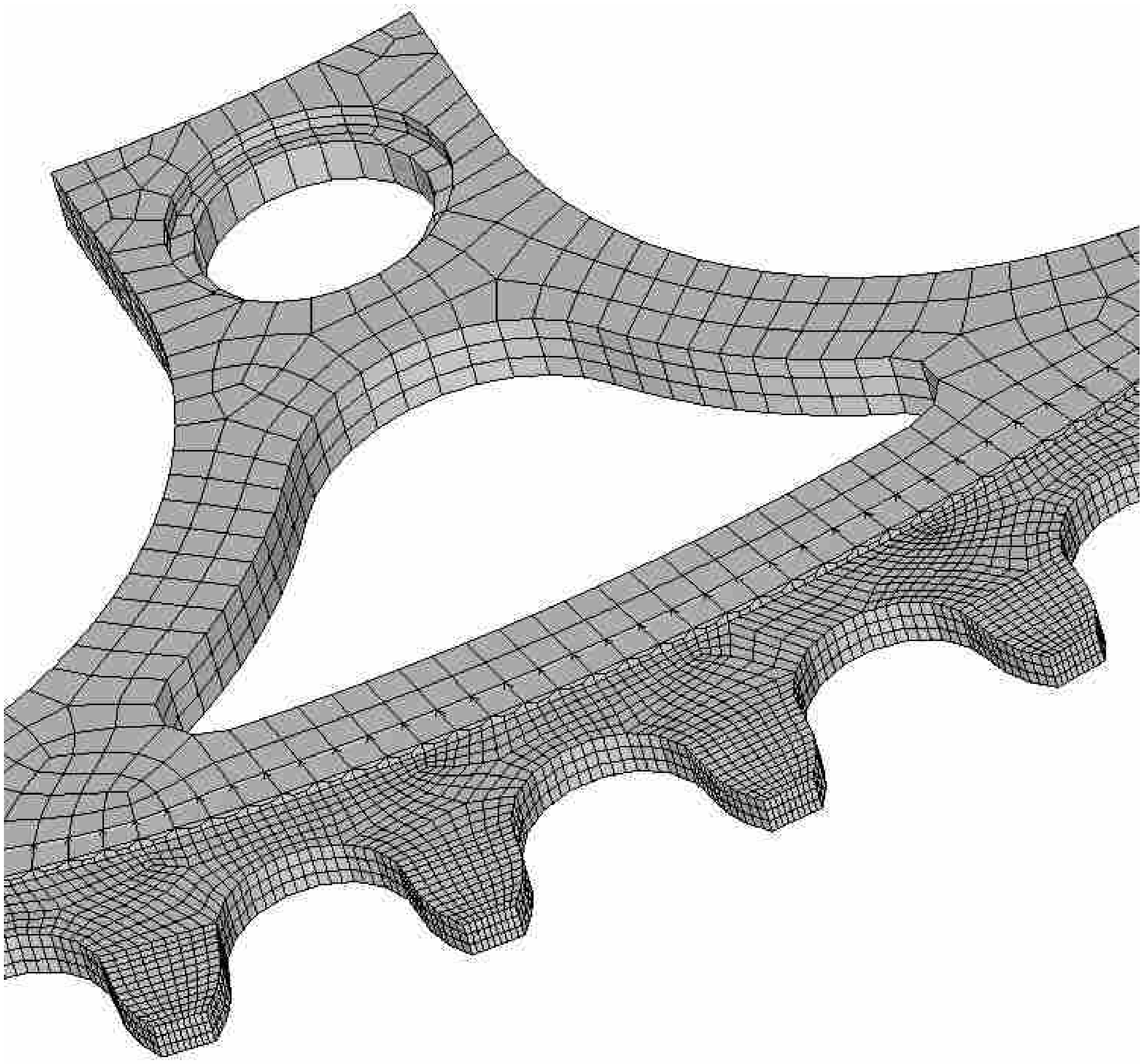


Figure H-3: Close up of the gear with refined teeth

The numerical results of both refinement schemes are given in Table H-1. Since this refinement region had no multiply-connected transition elements, the final element count is the same for both methods. The Selective Approach Algorithm took about half as long to complete the refinement process as is expected. The most peculiar result, however, is that the final minimum quality of the Selective Approach Algorithm was higher than that of the sheet refinement method. While this augments the attractiveness of the Selective Approach Algorithm, no definitive findings can be concluded from the result.

Table H-1: Numerical results for the gear example

Measurement	Sheet Refinement	Selective Appraoch
Initial Element Count	8569	8569
Final Element Count	63093	63093
Time (sec)	37.344	21.687
Initial Minimum Quality	0.4294	0.4294
Final Minimum Quality	0.1579	0.1580
Final Minimum Quality (Smoothed)	0.1896	0.2287

Multiple Refinements Example

Sometimes one level of refinement may not be enough. In this example, a simple brick's left surface is refined three times. Figure H-4 is a simple brick that contains 64 hexahedra.

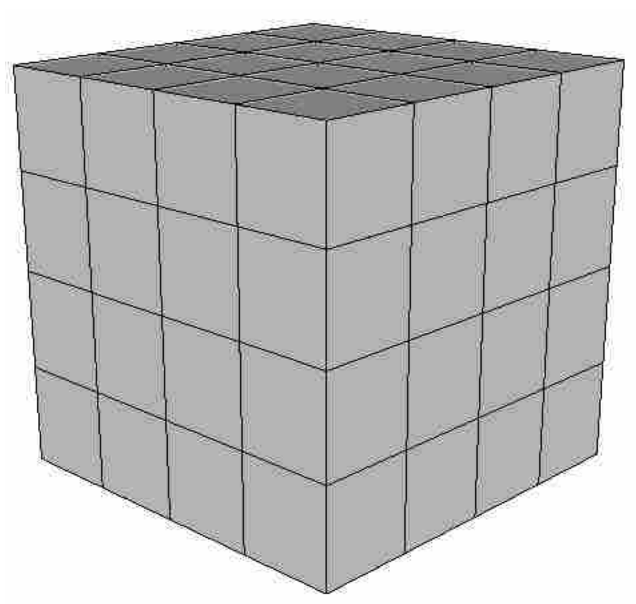


Figure H-4: Simple 4x4x4 brick

Figure H-5 shows the simple brick refined multiple times with the sheet refinement scheme implemented by Harris. Figure H-6 shows the simple brick refined multiple times with the Selective Approach Algorithm. The numerical results for this example are given in Table H-2.

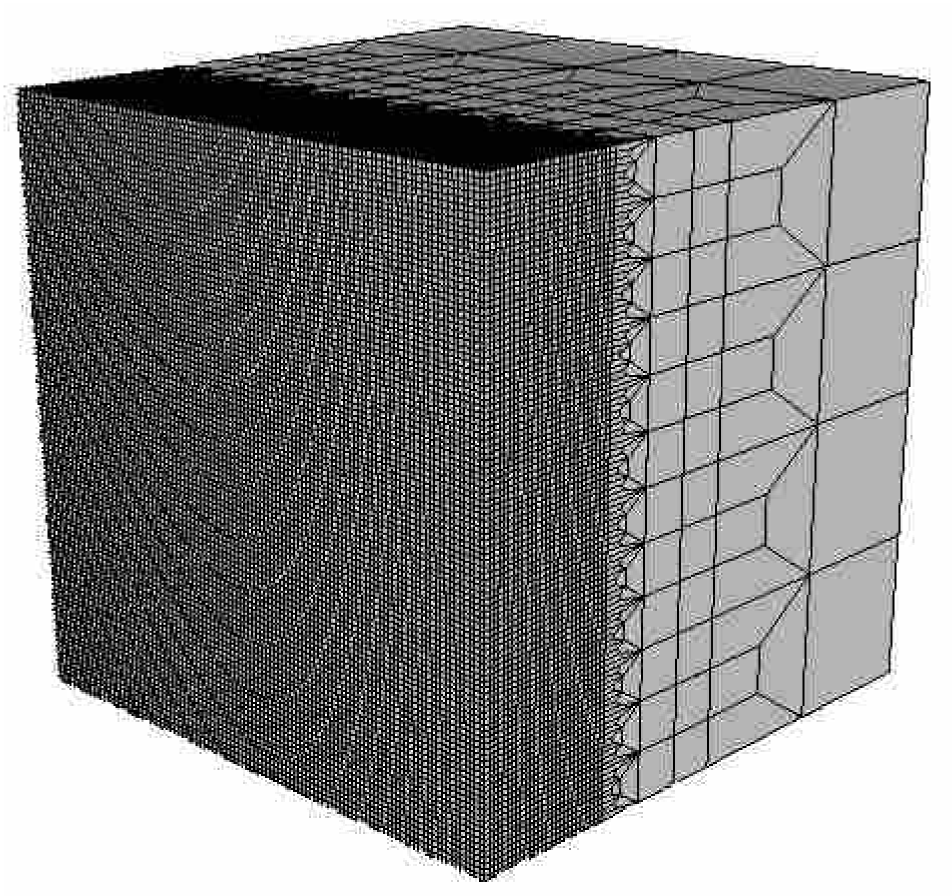


Figure H-5: Multiple sheet refinements of left face

The time required to complete the refinement with the Selective Approach Algorithm was much less than the time required to complete the sheet refinement scheme implemented by Harris. The final element count for the Selective Approach Algorithm was greater than the sheet refinement method. This is not usually the case since the

Selective Approach Algorithm refines multiply-connected transition elements more efficiently than the sheet refinement method. However, when the Selective Approach Algorithm refines multiple times, a buffer layer is added with each pass. This moves the transition elements away from other transition elements. Performing multiple refinements in this manner greatly increases the minimum quality of the mesh while the increase in the final element count is minimal. When comparing the final minimum element qualities, the Selective Approach Algorithm has a much higher value. This minimum quality is also sufficient for an accurate analysis while the minimum quality of the sheet refinement scheme is not.

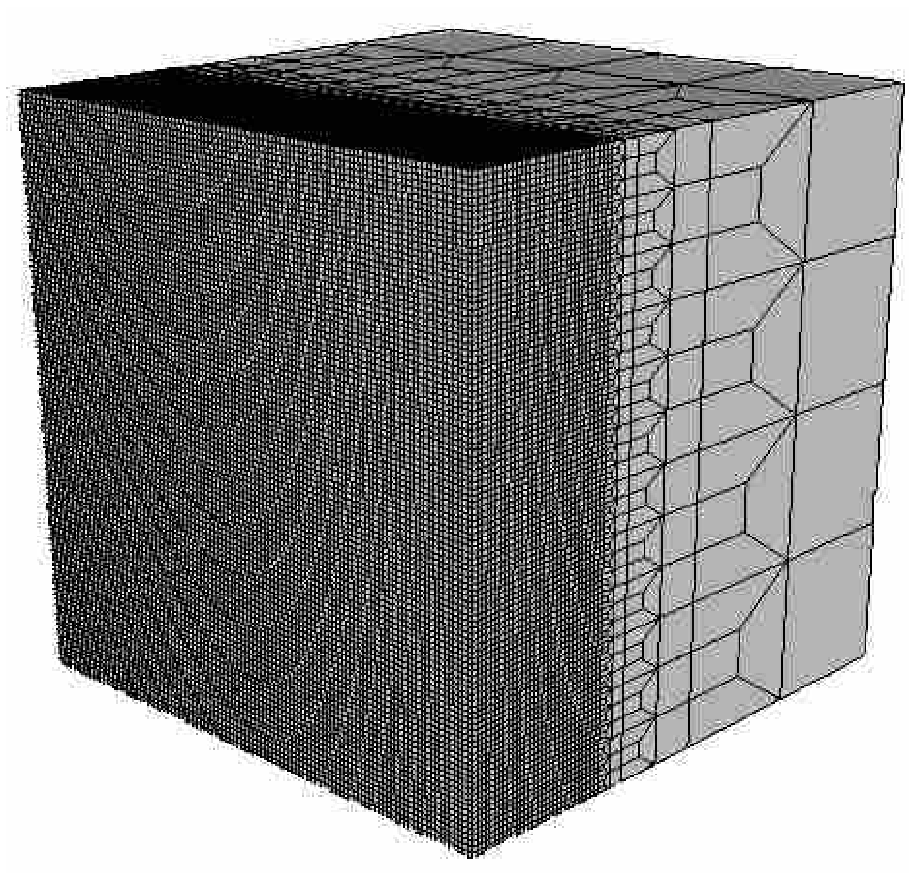


Figure H-6: Multiple refinements with Selective Approach Algorithm

Table H-2: Numerical results for the multiple refinements example

Measurement	Sheet Refinement	Selective Appraoch
Initial Element Count	64	64
Final Element Count	332864	370304
Time (sec)	126.4	14.66
Initial Minimum Quality	1.0	1.0
Final Minimum Quality	0.03501	0.3077

Mechanical Plate Example

Figure H-7 is part of a mechanical plate that has been meshed with an all-hexahedral mesh. It is likely that in a stress analysis, large concentrations of stress will occur in the neck of this plate. It is therefore desirable to refine the neck region of this mechanical plate to increase the numerical accuracy in this region.

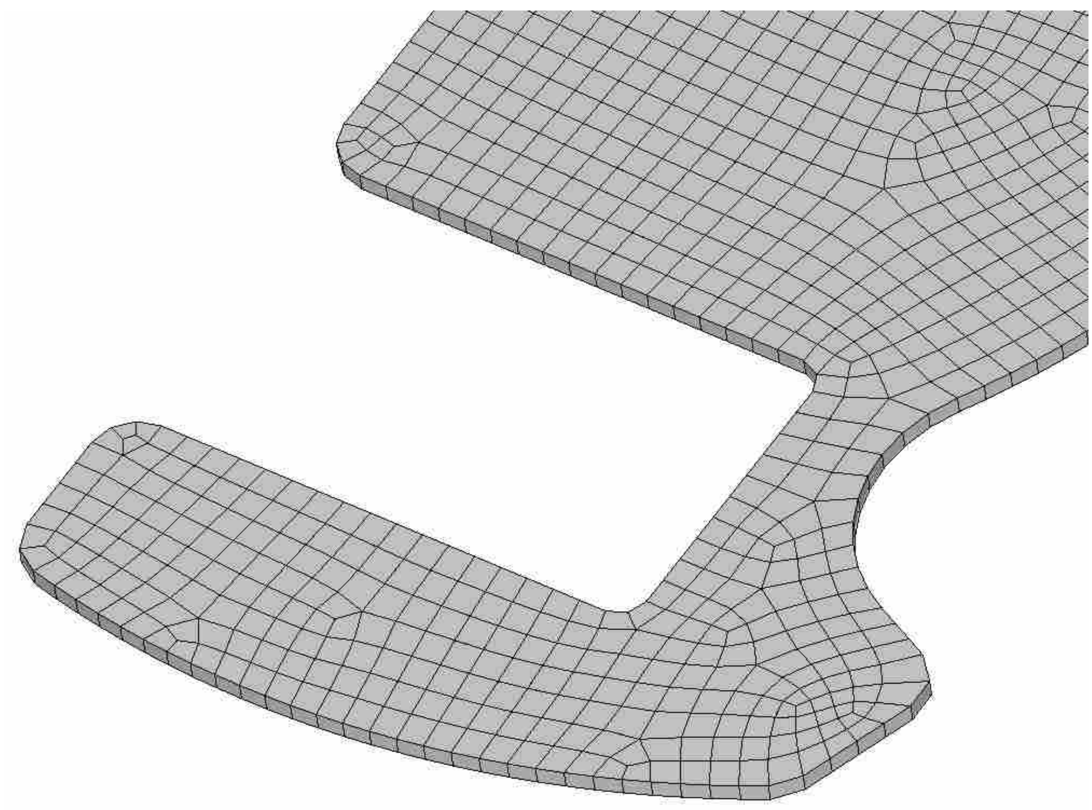


Figure H-7: Meshed mechanical plate

Figure H-8 shows the mechanical plate refined using the sheet refinement method implemented by Harris. Notice that more hexahedra were refined in order to remove the multiply-connected transition elements from the refinement region. Figure H-9 depicts the mechanical part after it has been refined using the Selective Approach Algorithm. Table H-3 gives the numerical results from both refinement schemes.

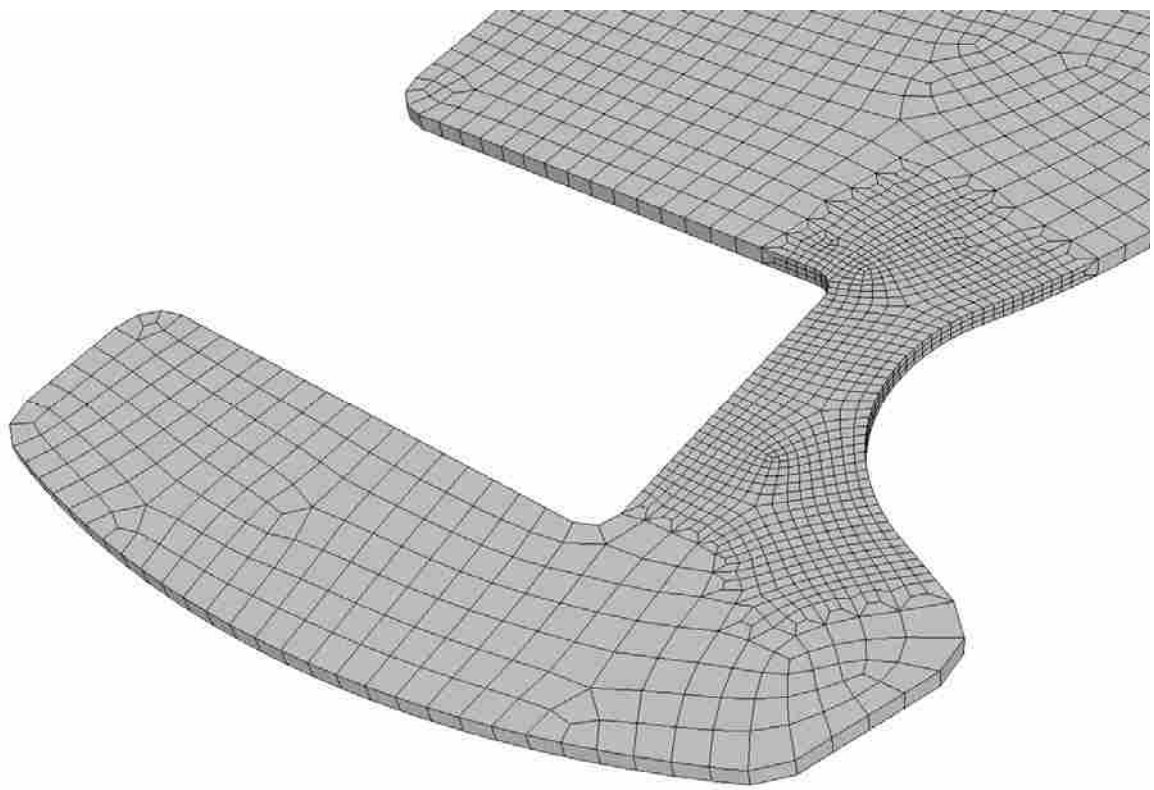


Figure H-8: Mechanical plate refined using the sheet refinement scheme

The Selective Approach Algorithm produced fewer hexahedra and completed the refinement nearly five times as fast as the sheet refinement scheme implemented by Harris. The sheet refinement method had a better minimum final quality; however, both refinement schemes had a quality that is suitable for an accurate analysis.

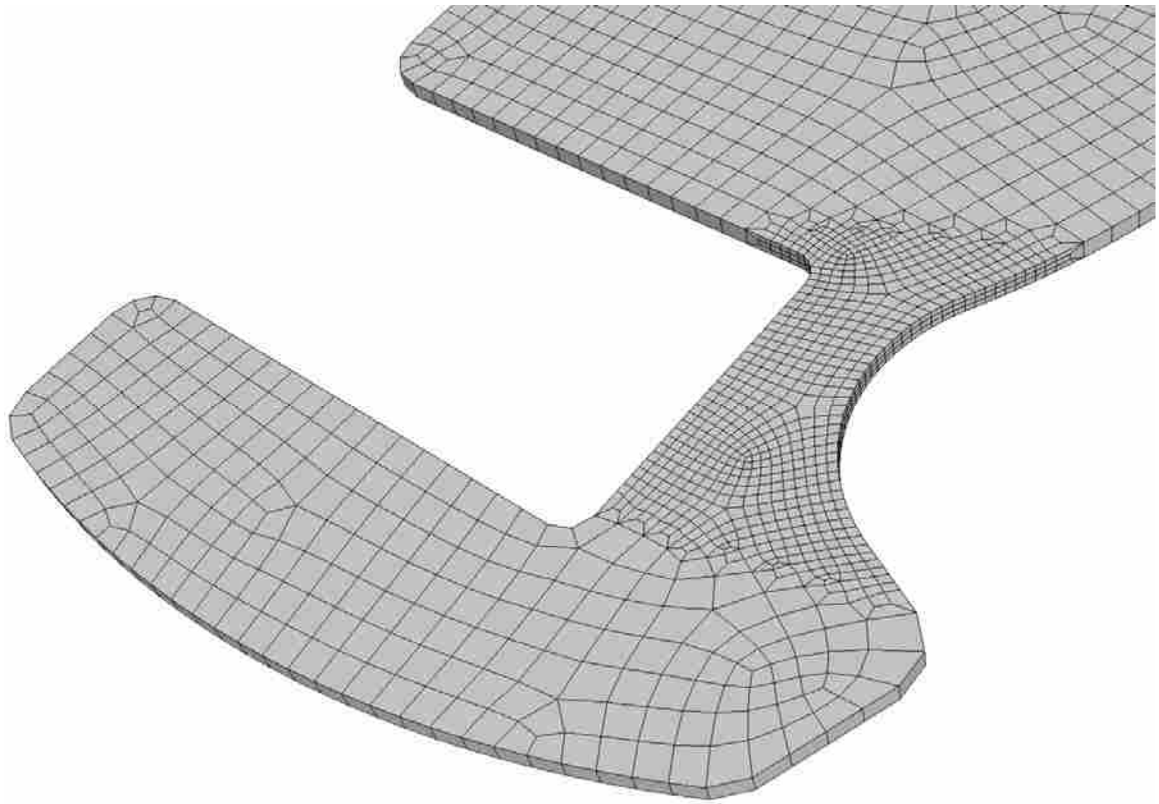


Figure H-9: Mechanical part refined using the Selective Approach Algorithm

Table H-3: Numerical results for the mechanical plate

Measurement	Sheet Refinement	Selective Appraoch
Initial Element Count	1643	1643
Final Element Count	3987	3539
Time (sec)	4.953	0.938
Initial Minimum Quality	0.5963	0.5963
Final Minimum Quality	0.2509	0.2028

Hook Example

The final example given in this appendix is the complete refinement of a mechanical hook. In general, a finite element analysis will converge to the correct answer as the number of elements approaches infinity. For this reason, many times an analyst may want to increase the total number of elements throughout the entire mesh to obtain a more accurate solution. Figure H-10 is the model of a mechanical hook that has been meshed with hexahedra. It contains 2032 hexahedra and has a minimum element quality of 0.5667.

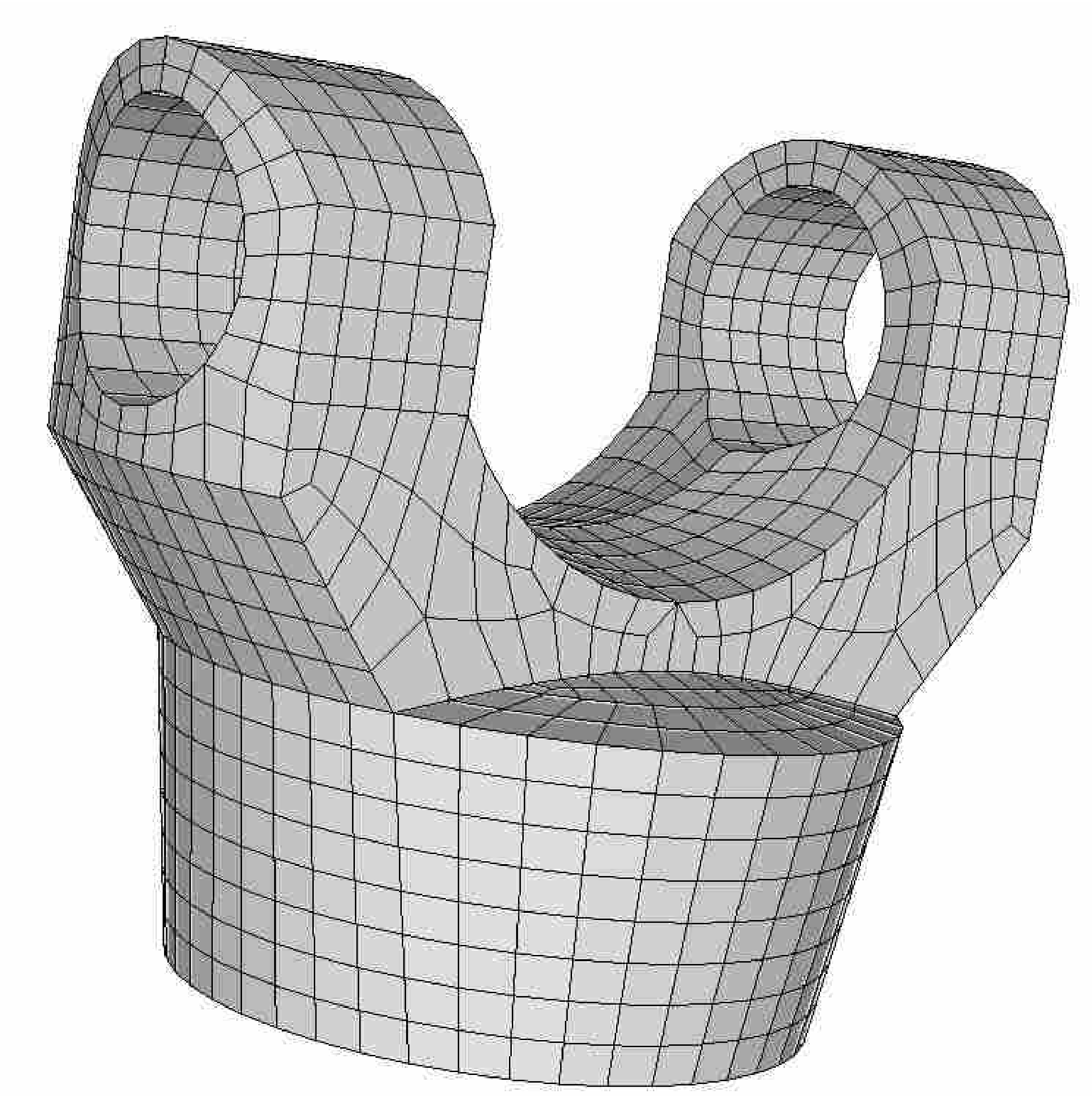


Figure H-10: Meshed mechanical hook

Figure H-11: Refined mechanical hook

Figure H-11 depicts the same mechanical hook after refinement. Both refinement schemes produced the same mesh with the same number of elements and the same final minimum quality. In fact, as expected the final minimum quality was the same as the initial element quality. The only difference between the two refinement schemes was the time required. Sheet refinement required 15 seconds to complete while the Selective

Approach Algorithm required only seven seconds to complete. Table H-4 gives the numerical results for this example. As shown in the body of this thesis, the scalability of the Selective Approach Algorithm is much better than the scalability of the sheet refinement scheme implemented by Harris. It would be expected that as the number of initial elements increases, the difference in times to complete the refinement would also increase.

Table H-4: Numerical results of the mechanical hook

Measurement	Sheet Refinement	Selective Appraoch
Initial Element Count	2032	2032
Final Element Count	54864	54864
Time (sec)	15.094	7.094
Initial Minimum Quality	0.5567	0.5567
Final Minimum Quality	0.5567	0.5567