3A.3

A Selective Approach to Conformal Refinement of Unstructured Hexahedral Finite Element Meshes

Michael Parrish[1], Michael Borden[2], Matthew Staten[3], Steven Benzley[4]

[1] Brigham Young University, Provo, UT U.S.A. mhp7@et.byu.edu
[2] Sandia National Laboratories, Albuquerque, NM, U.S.A. mborden@sandia.gov
[3] Sandia National Laboratories, Albuquerque, NM, U.S.A. mlstate@sandia.gov
[4] Brigham Young University, Provo, UT U.S.A. seb@byu.edu

Summary. 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 concave refinement regions and self-intersecting hex 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 Hex are refined in one step using one of seven templates. Because of its computational superiority over directional refinement, but its inability to handle concavities, element by element refinement is used in all areas of the specified region except regions local to concavities. The directional refinement scheme refines the three inherent directions of a hexahedron separately on a hex by hex basis. This differs from sheet refinement which refines hexahedra using hex sheets. Directional refinement is able to correctly handle concave refinement regions. A ranking system and propagation scheme allow directional refinement to work within the confines of the Selective Approach Algorithm.

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 (Tet) or hexahedral (Hex) elements are commonly used to model three dimensional problems. Tet elements have extremely robust modeling capabilities for any general shape while Hex 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 nonconformal elements which cannot be accommodated by some solvers. Other techniques insert non-hex elements that result in hybrid meshes or require uniform dicing to maintain a consistant 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 concavities (see Section 2.2). 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 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 hex sheets (see Section 2.2), concavities, 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]. Each sheet represents a unique and inherent direction within a hexahedron. Figure 1 shows a hexahedron with its three dual twist planes. Each plane represents a unique direction of refinement.

2.1 Element by Element 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 nonconformal 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 shows how a mesh is refined using element by element refinement.

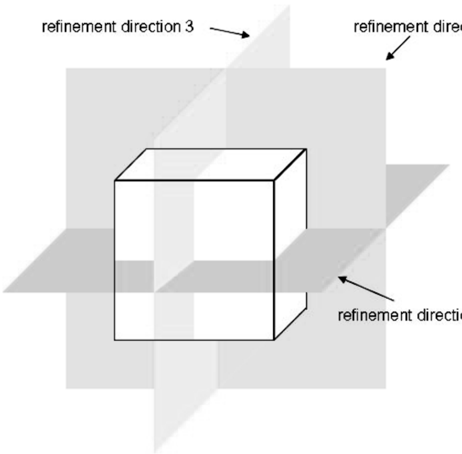


Fig. 1. A Hex with its twist planes representing directions of refinement

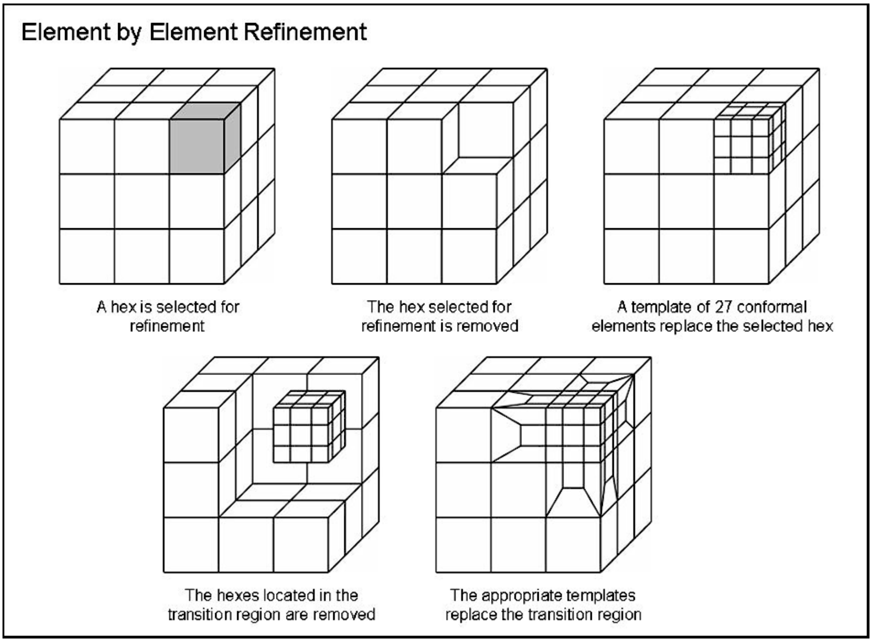
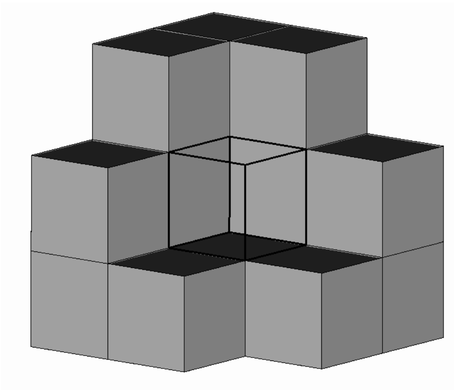
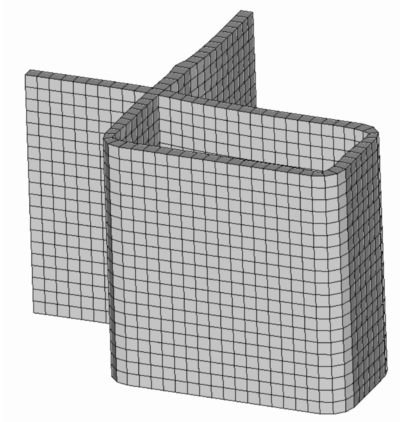


Fig. 2. Element by element refinement

Element by element refinement is limited by its inability to produce a conformal mesh in a concave region. In hexahedral refinement, a concave region 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 fig. 3(a)). 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.



(a) Example of concave region - hex outlined in black is a transition element in a concave region and shaded elements are selected for refinement



(b) Example of self-intersecting hex sheet

Fig. 3. Limitations of existing refinement methods

2.2 Sheet Refinement

The sheet refinement method refines a hex one direction at a time. The refinement region is processed in hex sheets allowing unstructured meshes to remain conformal throughout the entire process. Since conformity is maintained, sheet refinement inherently produces a conformal mesh. Figure 4 shows how a mesh is refined using 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 hex sheets, 2) the inefficiency in refining concave regions, and 3) scalability.

Self-Intersecting Hex Sheets

For conformal, all Hex meshes, a hex sheet must either initiate at a boundary and terminate at a boundary or form a closed surface. Sometimes meshing algorithms will create self-intersecting hex sheets as shown in Figure 3(b).

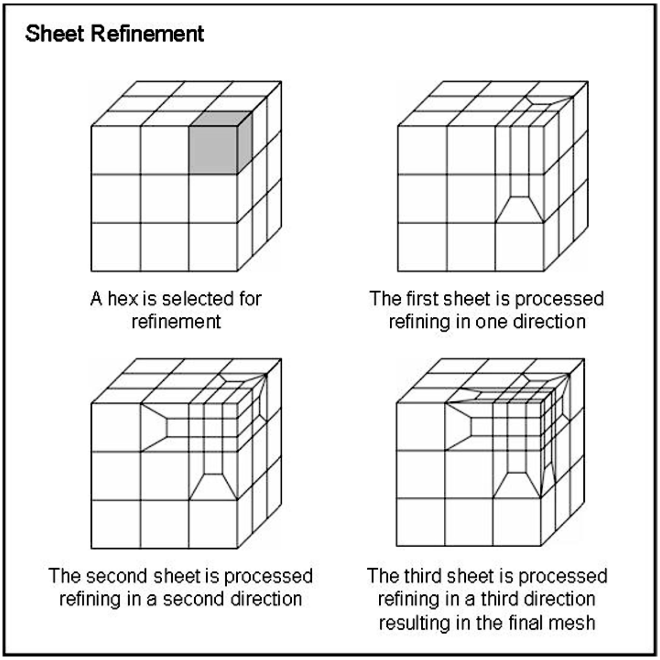


Fig. 4. Sheet refinement

A self-intersecting hex sheet is defined as any hex sheet that passes through the same stack of elements multiple times (i.e. any dual twist plane that intersects itself). Hexes at the intersection of a self-intersecting hex 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.

Concavities

Sheet refinement is able to produce a conformal mesh in concave regions however early implementations dealt with these concavities inefficiently. Initially, hexes were added to the concave region until all concavities were removed. While this produces a conformal mesh in a concave region, it leads to excessive refinement. Excessive refinement increases the computational load for both mesh generation and analysis. Templates were later proposed to handle concavities[11] but these templates were never implemented into any sheet refinement scheme.

Scalability

Empirical studies show that the time requirement of sheet refinement grows exponentially as the number of initial elements increases. A major contributor to this problem is the process of creating and deleting intermediate hexes.

The process occurs in the following manner(see fig. 4). The first sheet is processed, deleting the original hex and creating three intermediate hexes. The second sheet is then processed, deleting the three intermediate hexes created by the first sheet and creating nine new intermediate hexes. Finally, the third sheet is processed, deleting the nine intermediate hexes created by the second sheet and creating the final 27 hexes. In total, 13 hexes are deleted and 39 hexes are created to obtain the desired refinement. Also, each creation and deletion requires a data base query further increasing the computational time.

3 A Selective Approach

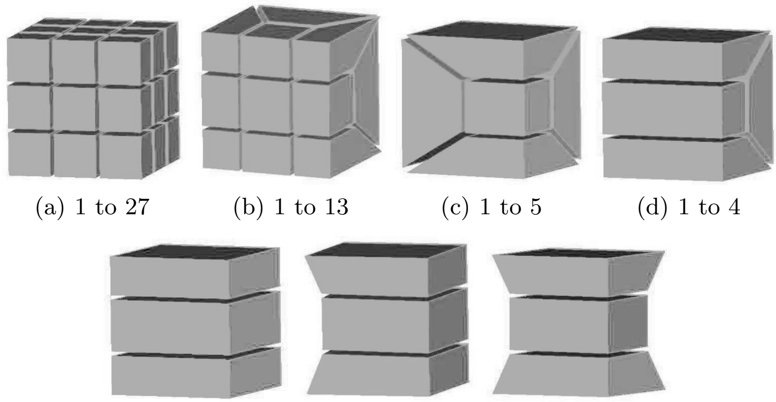
The Selective Approach Algorithm is a new robust refinement scheme. This procedure (as its name suggests) automatically selects the more appropriate of two different refinement schemes for each hex within a target region. A target region is defined as the elements selected for refinement and the transition 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 3.2) and directional (see Section 3.3) 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.

3.1 Templates

Seven templates[4][11][12] are used within the Selective Approach Algorithm (see fig. 5). 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. Figures 5(f) and 5(g) are the templates required to handle any concavity given in a target region. Figure 6 explains how the 1 to 3 template with 1 concavity is constructed. The 1 to 3 template with 2 concavities is constructed in a similar fashion.

3.2 Element by Element Refinement

The general process of performing element by element refinement was discussed in Section 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 hex in



(a) 1 to 27 (b) 1 to 13 (c) 1 to 5 (d) 1 to 4

(e) 1 to 3 (f) 1 to 3 with 1 concavity (g) 1 to 3 with 2 concavities

Fig. 5. Templates used in The Selective Approach Algorithm

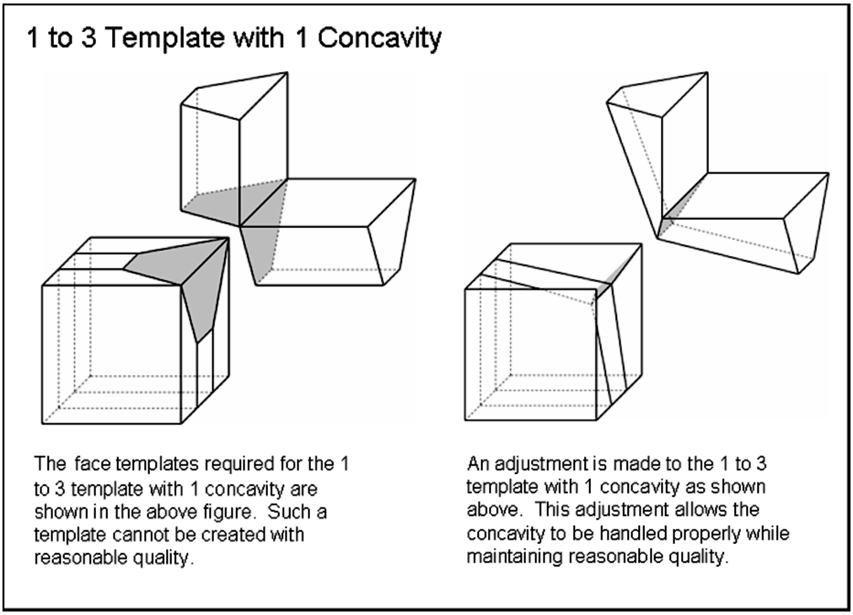


Fig. 6. Concavity template construction

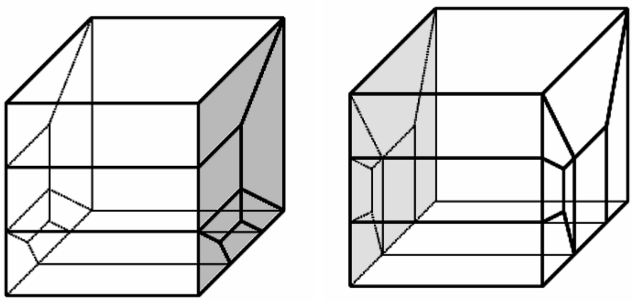
one step. A single hex is deleted and the final group of elements is created using one of the seven templates described previously. Since no intermediate hexes are created or deleted, the computational efficiency of element by element refinement is far superior to that of sheet refinement. The limiting factor then, of the element by element refinement method is its inability to handle concavities. Therefore, the Selective Approach Algorithm uses element by element refinement in all areas of the target region except areas local to concavities.

3.3 Directional Refinement

Like sheet refinement, the directional refinement scheme refines each inherent direction of a hex 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 concave regions. Directional refinement is therefore used in areas of the target region that contain concavities.

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 7 with two hexes that share a single face. The common face for both hexes is shaded in the figure. These two hexes share two common "directions" or "sheets." These directions must be refined in the same order in both hexes, otherwise a nonconformal mesh will be created. In Figure 7, both hexes contain valid refinement schemes yet the shared face is not conformable. This problem could potentially occur often since each hex 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) Refine order 1 (b) Refine order 2

Fig. 7. Conformity issues

To solve the conformity problem, the functionality of dual twist planes is used. Twist planes 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 by each concave region. Since each directional refinement group is confined to a single concavity, the possibility of containing a self-intersecting hex sheet is extremely unlikely. Each group is 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 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 8. Refinement then occurs on a hex by hex basis starting in the direction with the lowest rank and continuing in ranked order until the hex is completely refined and the algorithm moves onto the next hex.

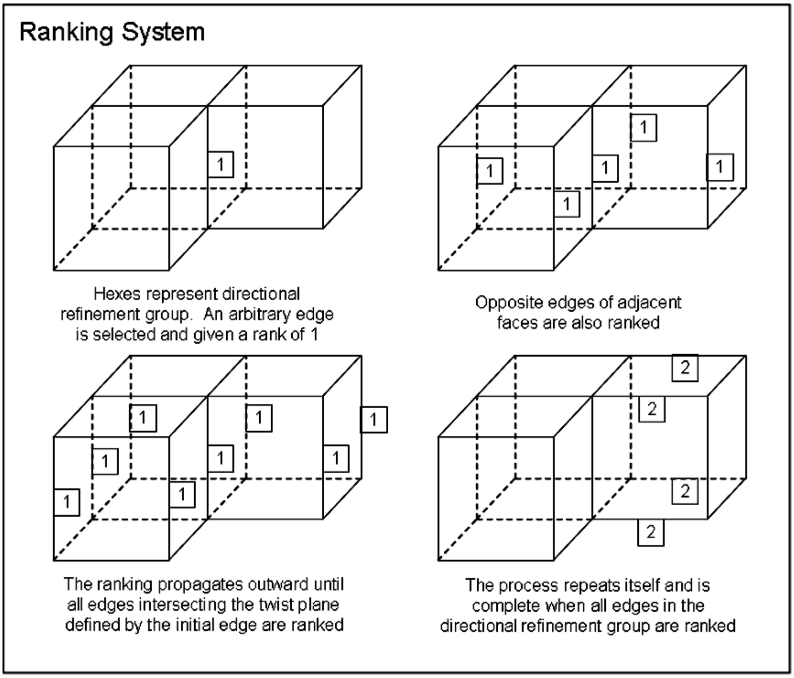


Fig. 8. Ranking system

Propagation Scheme

After a hex 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 parallel to the direction of refinement are considered in the propagation scheme. Figure 9 graphically shows how the propagation scheme works with a specific example.

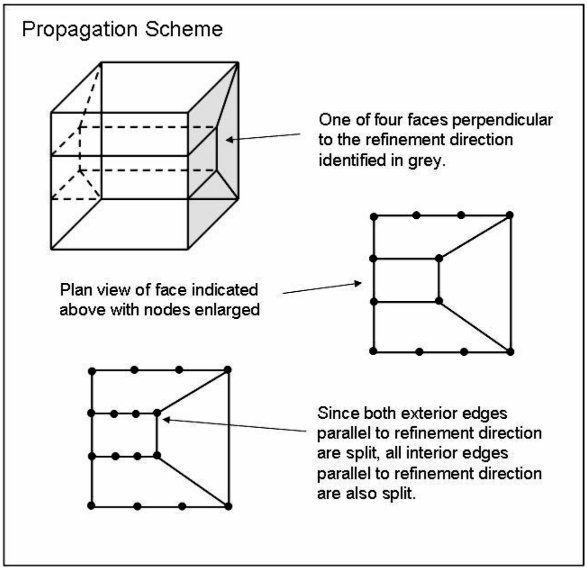


Fig. 9. Propagation scheme

3.4 Algorithm

An outline of the Selective Approach Algorithm is given in Algorithm 1. The Selective Approach Algorithm starts by applying the 1 to 27 template to the elements selected for refinement as specified in step 1.2. The transition hexes are all that remain after this step. Because element by element refinement is more efficient, it is applied first in step 1.4. The remaining hexes are then ranked as shown in algorithm step 1.11. Finally, the remaining hexes are refined directionally in order of increasing rank. The propagation scheme is applied to each hex during the directional refinement process. Figure 10 demonstrates the logic of the algorithm with a simple two-dimensional example.

Algorithm 1 The Selective Approach Algorithm

```
loop target hexes                                    ▷ element by element refinement
    apply 1 to 27 template to elements selected for refinement
end loop
loop transition hexes
    if template applies then
        refine hex using template
    else
        add to directional hex list
    end if
end loop
loop directional hex list
    apply ranking system
end loop
loop directional hex list                            ▷ directional refinement
    loop refinement directions in order of increasing rank
        apply template
        apply propagation scheme
    end loop
end loop
```

4 Results and an Example

The Selective Approach Algorithm solves the sheet refinement limitations of self-intersecting hex sheets, inefficiently handled concavities, 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.

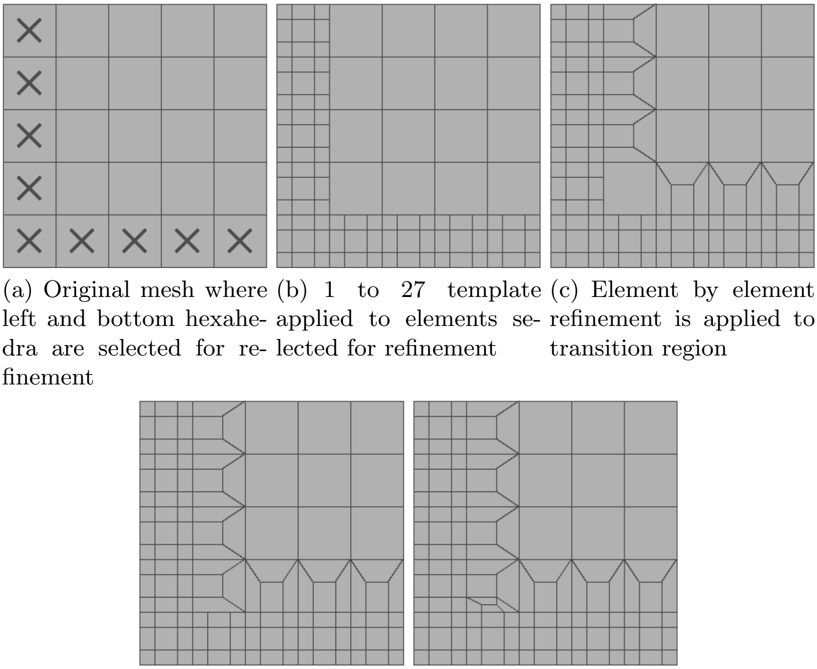
4.1 Self-Intersecting Hex Sheets

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

4.2 Concavities

To illustrate the new capabilities of the Selective Approach Algorithm when considering concavities, 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 11(a) shows the model refined using the sheet refinement scheme implemented by Harris and Figure 11(b) shows the brick refined using



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

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

(c) Element by element refinement is applied to transition region

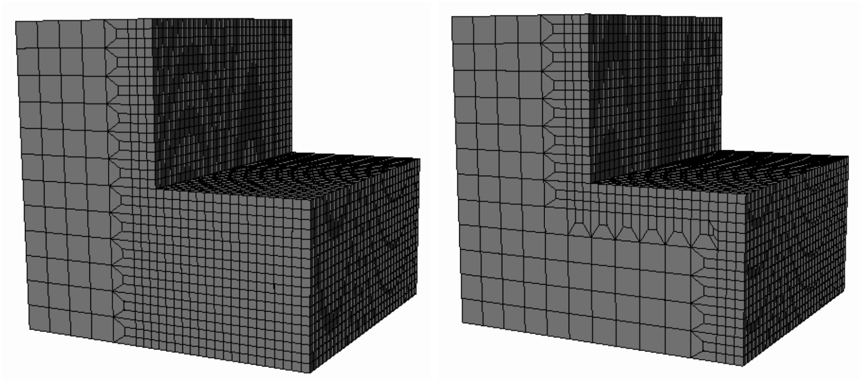
(d) Element is refined in one direction followed by propagation scheme

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

Fig. 10. Example of algorithm

the Selective Approach Algorithm. While sheet refinement could perform the refinement in a similar fashion to the Selective Approach Algorithm, the concave templates were never implemented. The sheet refinement scheme refined the entire bottom right section of the model in an attempt to remove the concavity. Excessive refinement is not a problem with the Selective Approach method. The newly implemented concave templates eliminate the need to add hexes to the target region.

Values for the number of elements, time for both methods, and element quality using a scaled Jacobian metric are given in Table 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 Appraoch method would also require less time thus lowering the overall time required for a full analysis. The final minimum scaled Jacobian produced by both refinement schemes is the same and adaquate for an accurate analysis.



(a) Sheet refinement (b) A Selective approach

Fig. 11. Simple model where surfaces composing right boundary are refined

Table 1. Results of refining the left and bottom faces of a brick

Measurement	Sheet Refinement	Selective Approach
Initial Elements	1188	1188
Final Elements	16500	8712
Time (sec)	5.359	0.859
Initial Min. Scaled Jacobian	1.0	1.0
Final Min. Scaled Jacobian	0.3143	0.3143

4.3 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. Each meshed brick was completely refined and the required time recorded. The results are shown in Figure 12.

Arguably the greatest advantage of the Selective Approach method over sheet refinement is scalability. Figure 12 decisively shows the exponential increase in time for sheet refinement as the nember of elements before refinement is increased. The scalability of the Selective Approach Algorithm is nearly linear in comparison. 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

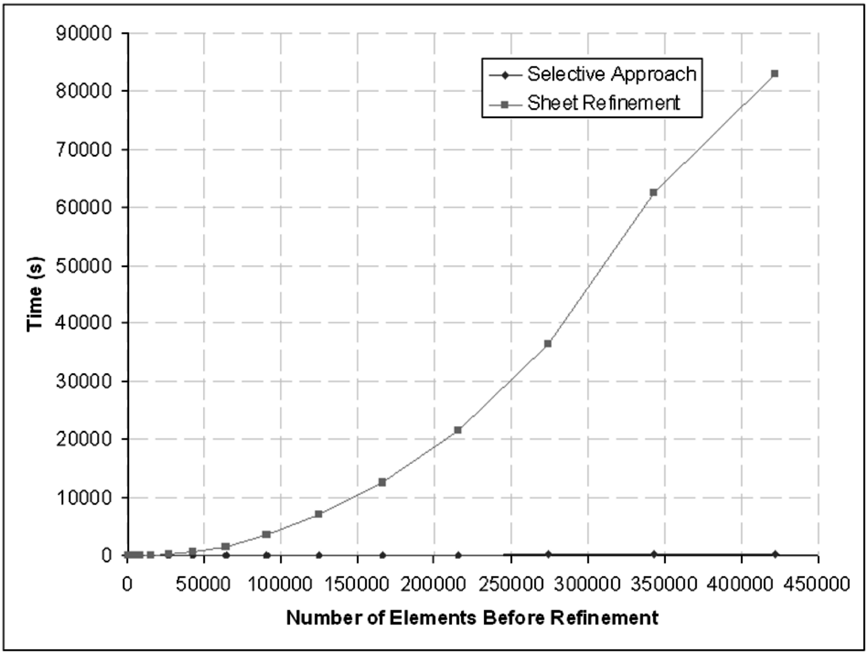


Fig. 12. Comparison of scalability between sheet refinement and the Selective Approach Algorithm

the brick were refined instead of the entire brick as shown in Figure 13. 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 14 shows the results of the second scalability test where directional refinement is used. This graph illustrates that while directional refinement may increase the computation time, the Selective Approach Algorithm is still far superior to traditional sheet refinement methods in terms of scalability.

4.4 Example

The example considered is a model of a gear (see Figure 15(a)). All of the teeth of the gear were refined using the Selective Approach Algorithm. Number of elements, speed, and quality using a scaled Jacobian metric were considered in the analysis and the model was smoothed before calculating the final element quality. Figure 15(b) is a closeup of a gear section before refinement. Figure 15(c) shows the same section after refinement. The results are given in Table 2.

In this example, the Selective Approach Algorithm refined the teeth of the gear, adding over 50,000 elements in approximately 20 seconds. The final

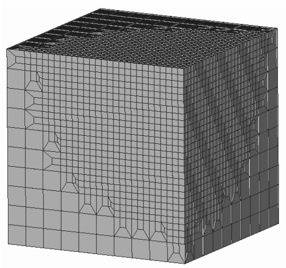


Fig. 13. Brick with constant radius away from top front vertex refined

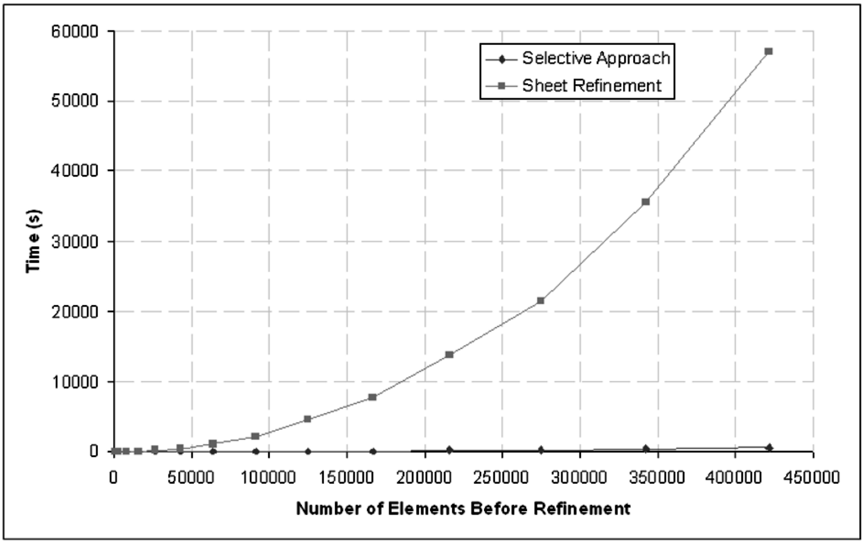


Fig. 14. Comparison of scalability between sheet refinement and the Selective Approach Algorithm with some elements refined using directional refinement

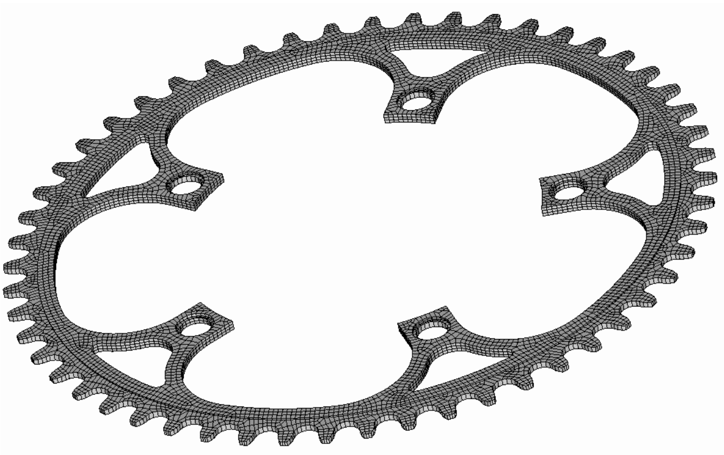
mesh is conformal and the smoothed minimum scaled Jacobian is adaquate for an analysis.

5 Conclusion

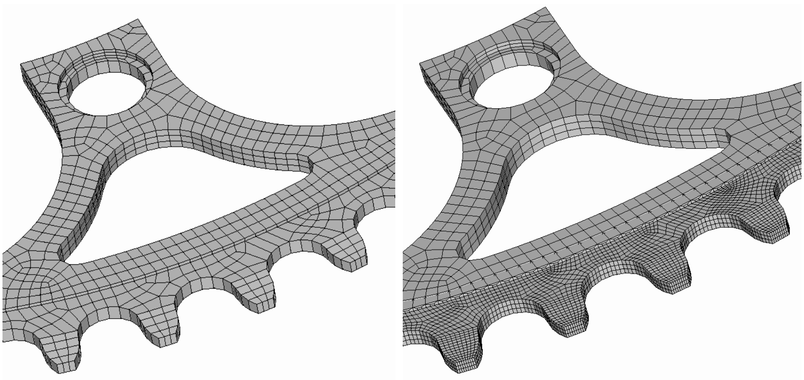
The refinement scheme presented in this work is a powerful mesh modification tool. The Selective Approach Algorithm is able to handle self-intersecting hex sheets, concavities, and scalability issues by leveraging the advantages of both element by element and sheet refinement schemes. Directional refinement is a new refinement technique that refines the three inherent directions of a hex sequencially while the target region is processed on a hex by hex basis. A

Table 2. Results of refining the teeth of a gear using the Selective Approach Algorithm

Measurement	Value
Initial Elements	8569
Final Elements	63093
Time (sec)	21.687
Initial Min. Scaled Jacobian	0.4294
Final Min. Scaled Jacobian	0.1580
Final Min. Scaled Jacobian (smoothed)	0.2287



(a) Gear model



(b) Close up of gear (c) Close up of gear with refined teeth

Fig. 15. Gear Example

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. 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. 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

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