2009-06-05

Automated All-Quadrilateral Mesh Adaptation through Refinement and Coarsening

Bret D. Anderson
Brigham Young University - Provo

AUTOMATED ALL-QUADRILATERAL MESH ADAPTATION THROUGH REFINEMENT AND COARSENING

by

Bret Dallas Anderson

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 2009

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Bret Dallas Anderson

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 Bret Dallas Anderson 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

AUTOMATED ALL-QUADRILATERAL MESH ADAPTATION THROUGH REFINEMENT AND COARSENING

Bret Dallas Anderson

Department of Civil and Environmental Engineering

Master of Science

This thesis presents a new approach to conformal all-quadrilateral mesh adaptation. In finite element modeling applications, it is often desirable to modify the node density of the mesh; increasing the density in some parts of the mesh to provide more accurate results, while decreasing the density in other parts to reduce computation time. The desired node density is typically determined by a sizing function based on either the geometry of the model or the results of a finite element solution. Although there are numerous mesh adaptation methods currently in use, including initial adaptive mesh generation, node redistribution, and adaptive mesh refinement, there are relatively few methods that modify the mesh density by adding and removing mesh elements, and none of these guarantee a conformal, all-quadrilateral mesh while allowing general coarsening. This work introduces a new method that incorporates both conformal refinement and coarsening strategies on an existing mesh of any density or configuration.

Given a sizing function, this method modifies the mesh by combining existing template based quadrilateral refinement methods with recent developments in localized quadrilateral coarsening and quality improvement into an automated mesh adaptation routine.

ACKNOWLEDGMENTS

I would like to thank my graduate advisor, Dr. Steven E. Benzley for his support in this research. I would also like to thank Sandia National Laboratories for providing funding as well as the software and expertise necessary to develop this new adaptation technique. Specifically, I would like to thank Jason Shepherd and Steven J. Owen for their constant willingness to provide assistance. I would also like to recognize the support of the other graduate students at BYU that I collaborated with on this work; in particular, Mark Dewey as a mentor when I began this research and Jared Edgel for his assistance in providing examples for this thesis. Lastly, I would like to thank my family for their constant support and encouragement, especially my wonderful wife, Carrie, and our son, Brian.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

The accuracy of the solution to a finite element analysis problem is dependent on both the shape and size of the elements in the mesh. For most two-dimensional finite element calculations, quadrilateral elements that are perfect squares provide the best accuracy. Also, properly sized elements must be small enough to minimize error in the solution, accurately model the geometry, and/or increase resolution in areas of high gradients. In addition to accuracy, analysis efficiency must be considered. Since computation time is directly related to the number of nodes in the mesh, a lower mesh density that maintains acceptable accuracy can significantly decrease the amount of time required for an analysis.

Mesh adaptation is the process of altering the characteristics of the elements in a mesh by changing element geometry, topology, and/or degree of interpolation functions. The goal of adaptive meshing is primarily to provide a finite element mesh that will ensure accurate results in areas of interest. Additionally, it is desirable to simplify the mesh as much as possible in less critical areas to reduce computation time. One of the principal methods used to adapt a finite element mesh is to vary the element density across the domain; providing a high enough element density to ensure accurate results in areas of interest while providing as few elements as possible in less critical areas to minimize computation time.

The desired element sizes throughout an adaptive mesh can be determined by use of a sizing function which specifies the target size of an element at a particular location and can be based on analysis-specific areas of interest, a posteriori error estimates, or geometric characteristics of the model. Some sizing functions may be available before the initial mesh is created and an appropriate mesh generation scheme can be used to create a mesh with the desired element density. However, since the areas of interest and desired element sizes in a mesh are sometimes unknown before an initial analysis is performed, mesh adaptation is often required after an appropriate sizing function is determined from the results of an analysis of a relatively coarse mesh. Additionally, some time dependent finite element applications require the mesh density to be updated in response to an evolving geometry [1] or shifting areas of interest [2, 3] and the mesh must be adapted several times throughout the analysis process.

The ability to automatically adapt a finite element mesh based on a sizing function is an important component of an automatic modeling and simulation process. Although it is not a new concept, its principal application has been to triangle and tetrahedral-based methods. Quadrilateral meshes are often preferred by analysts for improved accuracy over triangle-based methods. In spite of this, adaptive quadrilateral techniques are not as prevalent in the literature.

A truly general mesh adaptation scheme must have the ability to both enhance (refine) and simplify (coarsen) a mesh to provide sufficient accuracy and efficiency in the analysis. While there are numerous methods currently used, relatively few provide for both refinement and coarsening. Additionally, no current algorithm has the ability to

adapt an all-quadrilateral mesh with refinement and a coarsening technique not constrained to de-refining.

This thesis presents a unique all-quadrilateral mesh adaptation algorithm that modifies a given mesh by adding and removing elements and employs a coarsening process that is not limited to undoing previous steps of refinement. This algorithm combines existing quadrilateral refinement techniques [4] with recent developments in coarsening [5] and quadrilateral improvement [6] to adapt an existing mesh. Additionally, to provide an algorithm that will meet conformity and element type requirements of finite element solvers, this method guarantees a fully conformal, all-quadrilateral mesh.

The remainder of this thesis is organized as follows. Chapter 2 provides background of current methods in mesh adaptation. Chapter 3 presents a new adaptation method, outlines the algorithm, and describes heuristics used; showing each step performed in an example. Chapter 4 provides several examples of mesh adaptation performed by this method, including one example with a sizing function based from error estimates determined from an analysis. Chapter 5 provides a brief summary of this research and suggests areas of future research.

2 Background

To meet the ever increasing computational demands of complex finite element models, mesh adaptation has become a valuable area of study. There are three basic classes of adaptation, commonly referred to as *r*-, *h*-, and *p*-adaptation [7, 8]. *r*-adaptation, sometimes known as smoothing, refers to methods that alter element geometry by repositioning the nodes, but do not change the topology of the mesh. *h*-adaptation, refers to methods which change both the geometry of the elements and the topology of the mesh by adding and/or removing elements. *p*-adaptation, involves methods that do not alter the geometry or topology of individual elements; instead, these techniques change the degree of the elements in the mesh. While recognizing the importance of both *r*- and *p*-adaptive methods, this thesis focuses on the *h*-adaptive technique, concentrating specifically on the conformal refinement and coarsening of all-quadrilateral meshes.

2.1 Current Methods

Initial adaptive mesh generation is perhaps the easiest way to build an adapted finite element mesh because it simply employs the given sizing function in the original creation of the mesh. This is a widely used method and is available in many mesh generation schemes, including paving [9]. The major drawback of initial adaptive mesh

generation is that it requires significant foresight into the probable results of the analysis which are used to determine element sizes and an appropriate distribution of element density across the mesh. Because of this required foresight, initial mesh generation techniques that incorporate sizing are particularly useful when based on geometric characteristics of the model [10], rather than a posteriori error estimates.

Closely related to initial adaptive mesh generation is adaptive mesh re-generation; a mesh adaptation scheme in which the mesh is analyzed, a sizing function is determined, and the entire mesh or the region of the mesh requiring modification is removed and reconstructed according to the new sizing function. A significant amount of research has been done in this area and numerous algorithms have been presented employing adaptive mesh regeneration [11, 12, 13]. Although regenerating the mesh does not have the same drawbacks with respect to required foresight as does initial adaptive mesh generation, deleting and re-creating the mesh can be inefficient compared with other methods that require less modification.

Node redistribution, commonly referred to as *r*-adaptation, is an adaptation scheme based solely on relocating nodes to alter the mesh density. Several effective algorithms have been developed using this method [14, 15, 16, 17]. Although node redistribution is very simple in that it does not alter the topology of the mesh, the allowable change in element density can be limited by the shape quality of the elements, constraining the ability to both refine and coarsen the mesh.

Adaptive mesh refinement is an *h*-adaptation method based solely on refining the region of interest. Since the primary goal of adaptation is to ensure accurate results, this is a popular adaptation tool [18, 19, 20, 21] that can be used to provide sufficient

resolution in all areas of the mesh that require it and is often used for applications requiring adaptation based on a posteriori error estimates. Because there are no provisions for coarsening in adaptive mesh refinement, there must be enough foresight to produce a coarse base mesh with enough resolution to provide good error estimates. At the same time, this coarse mesh should not require a significant amount of computation time outside the areas of interest, which may not be known before the analysis.

Possibly the most effective methods to adapt a mesh use a combination of two or more types of adaptation. Branets and Carey [22] suggest combining adaptive refinement with node redistribution, but present no formal algorithm to accomplish the goal. Zhang and Bajaj [23] propose an adaptation method combining initial adaptive mesh generation with adaptive mesh refinement, but provide no consideration of coarsening. Jiao, et al. [24] propose combining topological modifications to the mesh with node redistribution, specifically for evolving surfaces; however, their presented algorithm is limited to meshes with triangle elements only.

2.2 Concurrent Refinement and Coarsening

While the methods discussed in the previous section can be effective ways to adapt a mesh, they all lack the ability to provide effectively coupled coarsening and refinement for quadrilateral meshes. Utilization of both coarsening and refinement in mesh adaptation greatly increases the ability to modify a mesh to provide an appropriate element density without the need to know the coarsest state of the model. While several algorithms employing coarsening and refinement have been developed for both surface and volume meshes, most do not have provisions for all-quadrilateral adaptation.

In some applications, large deformations in the model geometry are realized and the mesh must be adapted to provide high quality elements. Wan, et al. [1] present a tetrahedral adaptation method that uses topological operators such edge collapsing and edge splitting to provide coarsening and refinement. In other applications the area of interest changes with time and the mesh must be adapted to meet the element density requirements of an evolving solution. Li, et al. [3] present a surface adaptation algorithm that applies refinement ahead of a moving boundary and coarsening behind. Similarly, Kallinderis and Vijayan [25] present an algorithm that provides evolutionary adaptation by adding and removing elements in volume meshes. Although these two methods provide both coarsening and refinement, they are limited to triangle and tetrahedral elements, respectively. De Cougny and Shephard [26] also present a volume adaptation method that provides for coarsening and refinement; however, it too is limited to only tetrahedral elements.

In addition to providing adaptation for evolving geometries or solutions, mesh adaptation procedures can be used to increase element quality. Chalasani, et al. [27] use adaptation to improve element quality of extruded volume meshes near concave or convex features. Their algorithm employs both coarsening and refinement through edge collapse and face refinement, but does not maintain conformity and allows a hybrid mesh.

Connectivity requirements of quadrilateral elements make the creation of a conformal adaptation algorithm of all-quadrilateral meshes using concurrent coarsening and refinement a much more difficult problem than adapting a mesh with only triangular elements or even a quad-dominant mesh. Hierarchical adaptation methods [28] have

been developed that are able to adapt all-quadrilateral meshes while maintaining conformity by using quadtree refinement with transition templates. Coarsening in hierarchical adaptation is accomplished by simply removing quadtrees from parent elements; however, a major limitation of this procedure is that there is no way for the mesh to be coarsened further than the initial base mesh. By taking advantage of new coarsening techniques the algorithm presented in this thesis provides adaptation that includes coarsening not limited to undoing previous refinement.

3 Automated Mesh Adaptation

3.1 Sizing Functions

The first step in creating an adaptive mesh is to provide an appropriate sizing function across the mesh domain. Sizing functions are typically based on error estimates derived from the solution of a finite element analysis, geometric characteristics of the model, or other user defined constraints. A solution based sizing function might specify an increased element density in regions of high stress or strain gradients, such as the edge of a hole, the location of an applied load, or at a change in material. Geometry based sizing functions, such as a skeleton sizing function [10, 29], consider feature size as well as surface or boundary curvature and specify an appropriate element density throughout the mesh.

In addition to specifying the desired size of elements throughout the mesh, sizing functions must also take into account mesh gradation, the rate at which the element sizes change across the mesh [30, 31]. Gradation control is an important part of ensuring high shape quality of elements in a conformal mesh by not allowing a large change in size between adjacent elements. Although it is an important area of study, the development of sizing functions is not part of this research and it is assumed that an appropriate sizing function is provided as input to each adaptive meshing problem.

3.2 Tools and Requirements

The adaptation technique presented in this thesis employs a combination of quadrilateral refinement, coarsening, and quality improvement operations to modify a given mesh. Only adaptation operations that preserve a conformal all-quadrilateral mesh are used. Additionally, these operations can be applied locally to allow for concurrent coarsening and refinement.

Since the primary goal of adaptation is to ensure accurate results, refinement is usually required. This adaptation algorithm employs a refinement method that subdivides faces in the refinement region using a four element quadtree (referred to as 2-refinement) with templates inserted into the transition zone to maintain a conformal all-quadrilateral mesh [32]. While nine element quadtrees (3-refinement) are sometimes used to refine quadrilateral faces, 2-refinement was chosen because it offers more control over the number of elements added to the mesh. Further discussion of refinement techniques can be found in Appendix A.

This adaptation algorithm uses the Automated Quadrilateral Coarsening by Ring Collapse (AQCRC) algorithm recently developed by Dewey [5]. This coarsening method provides completely localized coarsening by selecting and removing rings of adjacent quadrilaterals from within a specified coarsening region. One consequence of the removal of these coarsening rings is the creation of poor quality faces and quadrilateral improvement (clean-up) is a necessary step in this process. Although the AQCRC algorithm is a very effective local coarsening technique, it assumes that the coarsening rings are closed rings and does not have any provisions for coarsening of the mesh boundaries. Because of this limitation, the removal of dual chords [33] is also employed

in the adaptation procedure to coarsen the boundaries. While almost any chord can be removed from a quadrilateral mesh, it is not guaranteed to be a local operation and will usually coarsen outside of the coarsening region. Despite the non-local nature of chord removal, it is necessary tool used for boundary coarsening in this algorithm. Further discussion of these and other coarsening techniques can be found in Appendix B.

In addition to the need for clean-up within the AQCRC algorithm, poor quality faces may form as a result of the refinement of irregular regions, making quadrilateral improvement a necessary step in this adaptation procedure. The clean-up procedure used in this algorithm works to improve nodal valence, providing a higher quality mesh [6]. Further discussion of this quadrilateral improvement method is found in Appendix C.

3.3 Algorithm

The goal of this automated mesh adaptation algorithm is to modify an existing mesh so that all of the faces are as close to the size specified by a sizing function but no larger. This ensures that solution accuracy and resolution are not sacrificed for decreased computation time. The algorithm flowchart is shown in Figure 3-1 and described by the following steps:

1. A quadrilateral mesh to be adapted and an accompanying sizing function are provided as input.
2. Each curve defining the boundary of the mesh is checked to see if coarsening is required. If a bounding curve must be coarsened, chords intersecting the boundary are removed until an appropriate size is reached.

3. If coarsening is needed anywhere in the mesh, those regions are coarsened. The clean-up algorithm is included as a step within the coarsening algorithm. If at any point it is determined that coarsening is not needed, this step is skipped in all future iterations.
4. If refinement is needed anywhere in the mesh, those regions are refined.
5. Following the refinement of elements in the mesh, the entire mesh surface is cleaned-up.
6. Steps 3 through 5 are repeated until sufficient refinement has occurred.

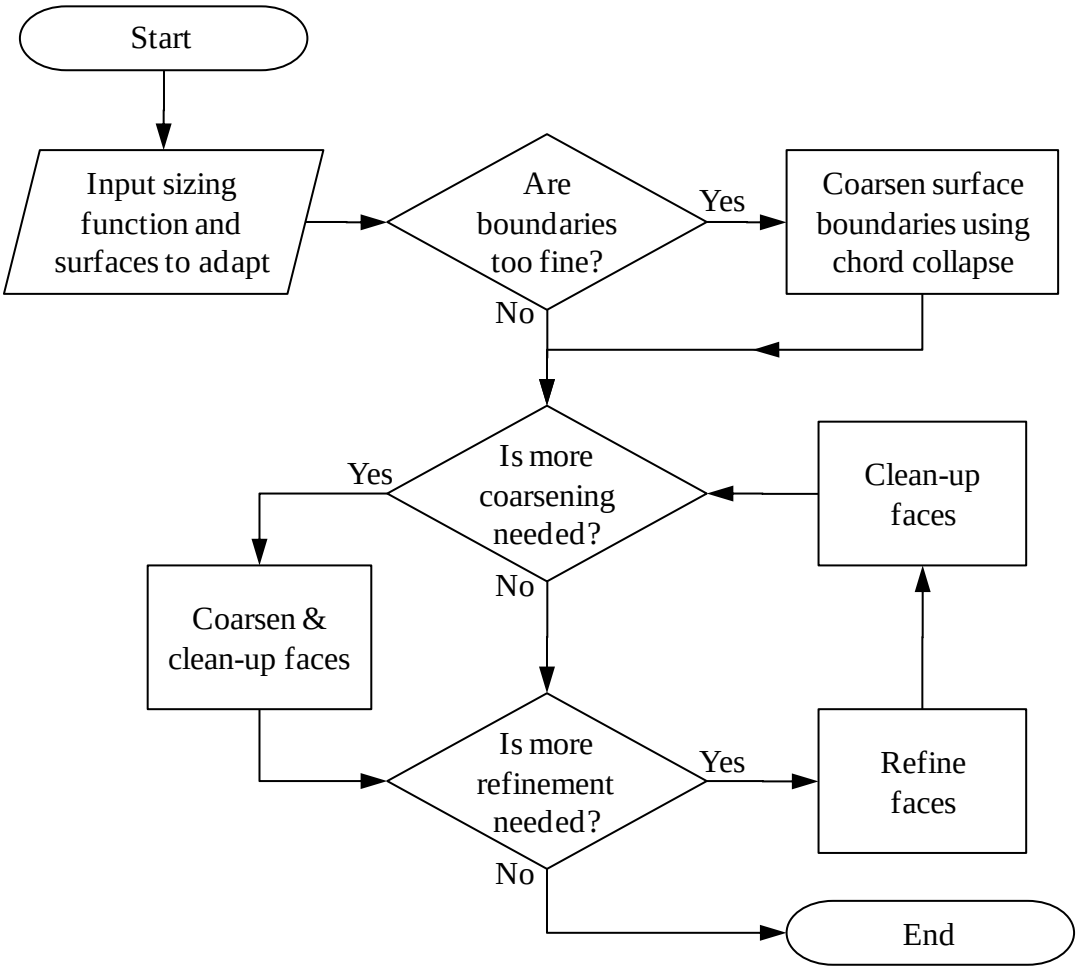

Figure 3-1: Algorithm flowchart.

3.4 Algorithm Example

To completely explain the steps of this algorithm, an example of adapting a mesh to a circular line load on a planar surface, as shown in Figure 3-2, is provided. In this case, the area of interest is at the location of the load. For simplicity, the algorithm can be divided into three distinct parts; input, boundary coarsening, and iterative coarsening/refinement.

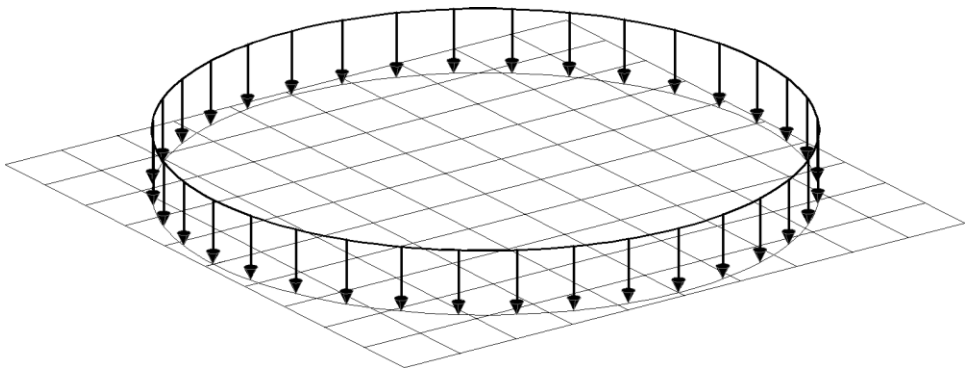

Figure 3-2: Circular load on plane surface.

3.4.1 Input

This algorithm requires an already meshed surface and an appropriate sizing function to be provided as input. In this example, the surface is a flat 10 x 10 plate, meshed with a perfectly structured 10 x 10 quadrilateral mesh shown in the left panel of Figure 3-3 and a contour plot of the sizing function is shown in the right panel. The dark colors represent the need for high element density while the light colors recommend a low element density.

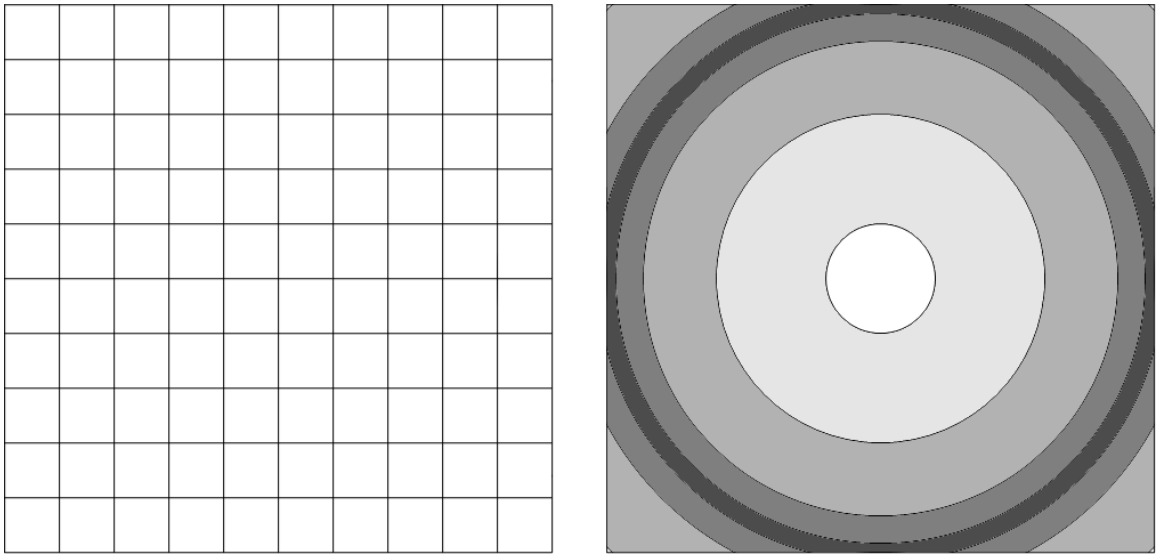

Figure 3-3: Input for circular load on plane surface. Initial base mesh (left) and contour plot of sizing function (right).

As defined by Equation 3-1, the size of each quadrilateral face is the average length of its four edges; therefore, for this example the size of each face in the initial mesh is 1.0. The sizing function provided in this example specifies a high element density at the location of the applied load with an element size of 0.1. This specified element density gets progressively lower, varying linearly, as we move further away from the load, eventually reaching a recommended element size of nearly 5 at the center of the plate.

$$h_a = \frac{1}{4}\sum_{i=1}^{4} l_i \qquad \textbf{(3-1)}$$

where h_a = actual face size

l_i = length of i^{th} edge of quadrilateral face

3.4.2 Boundary Coarsening

Since the AQCRC algorithm developed by Dewey does not allow boundary coarsening, it is achieved with simple chord removals in areas of the boundary that require larger element sizes. The edge length ratio is defined in Equation 3-2 as the ratio of actual edge size to desired edge size and is calculated for each edge along boundary curves. The algorithm uses these edge length ratios to select chords for removal.

$$f_l = \frac{l_a}{l_d} \tag{3-2}$$

where f_l = edge length ratio

l_a = actual edge length

l_d = desired edge length as specified by the sizing function

In this example, boundary coarsening is necessary near the corners of the mesh. Four chords, shown in the left panel of Figure 3-4 are chosen for removal. The right panel of Figure 3-4 shows the mesh following the removal of the four chords. Each bounding curve that was coarsened is then smoothed to maintain a more isotropic mesh structure.

3.4.3 Iterative Coarsening, Refining, and Quality Improvement

The remainder of the algorithm modifies the interior of the mesh by iteratively coarsening and refining elements until the goal has been reached. To provide a balanced approach to the adaptation problem, each iteration of this algorithm alternates between

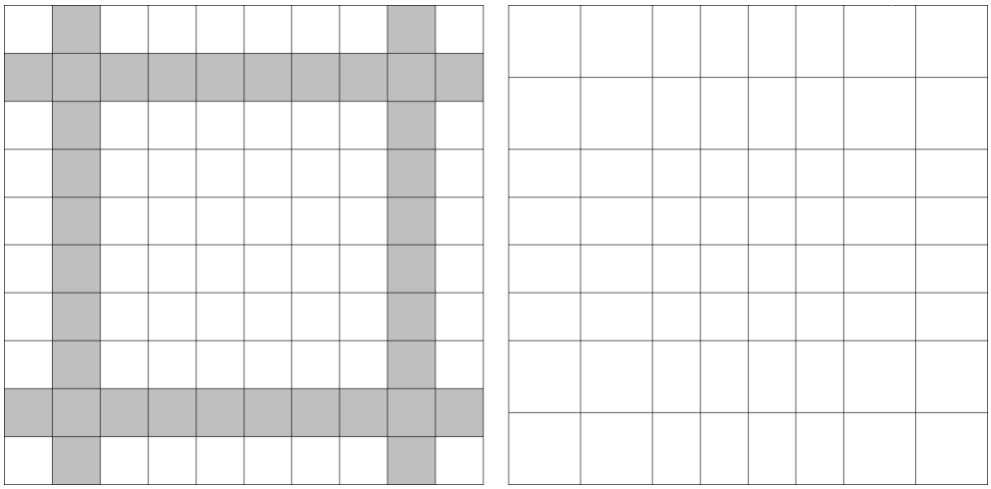

Figure 3-4: Chords selected for removal and removed.

coarsening and refinement. Since coarsening tends to make the mesh less structured and refinement tends to make the mesh more structured, the algorithm always begins with coarsening which is followed by refinement. Refining after coarsening also helps to achieve the goal of ensuring that the elements in the mesh are smaller than specified by the sizing function.

Each iteration begins by calculating the size ratio of each face as defined in Equation 3-3. A size ratio greater than 1.0 indicates that the face is too large and should be refined; a size ratio less than 1.0 indicates that the face is too small and can be coarsened.

$$f_s = \frac{h_a}{h_d} \tag{3-3}$$

where f_s = face size ratio

h_a = actual face size as defined in Equation 3-1

h_d = desired face size as specified by the sizing function

To provide more control over the amount of coarsening that takes place and to ensure that the coarsening operation does not overshadow refinement requirements, a dynamic threshold, given in Equation 3-4, is used to govern which faces should be coarsened and which should not be.

$$t_c = 1.0 - 0.2n_c \quad \textbf{(3-4)}$$

where t_c = coarsening threshold

n_c = coarsening iteration number; 0,1,2…

Any face with a size ratio less than the coarsening threshold is considered too small and should be coarsened. For example, in the first iteration, any face with a size ratio less than 1.0 is considered for coarsening. In later iterations, the coarsening threshold is gradually relaxed until the sixth iteration when it becomes zero and disappears. Also, if at any point in the adaptation process, fewer than 10% of the faces in the mesh require coarsening, the coarsening is considered complete and is skipped in all future iterations. This requirement helps to ensure that the same mesh regions are not being repeatedly coarsened and refined. Figure 3-5(a) shows the faces selected for coarsening in the example. This shaded region is coarsened and the resulting mesh topology can be seen in Figure 3-5(b).

Since the coarsening algorithm requires a contiguous region of quadrilateral faces to create coarsening rings, a lone face requiring coarsening surrounded by faces that do not require coarsening will automatically be neglected by the AQCRC algorithm, making sure that coarsening does not extend outside of the desired region.

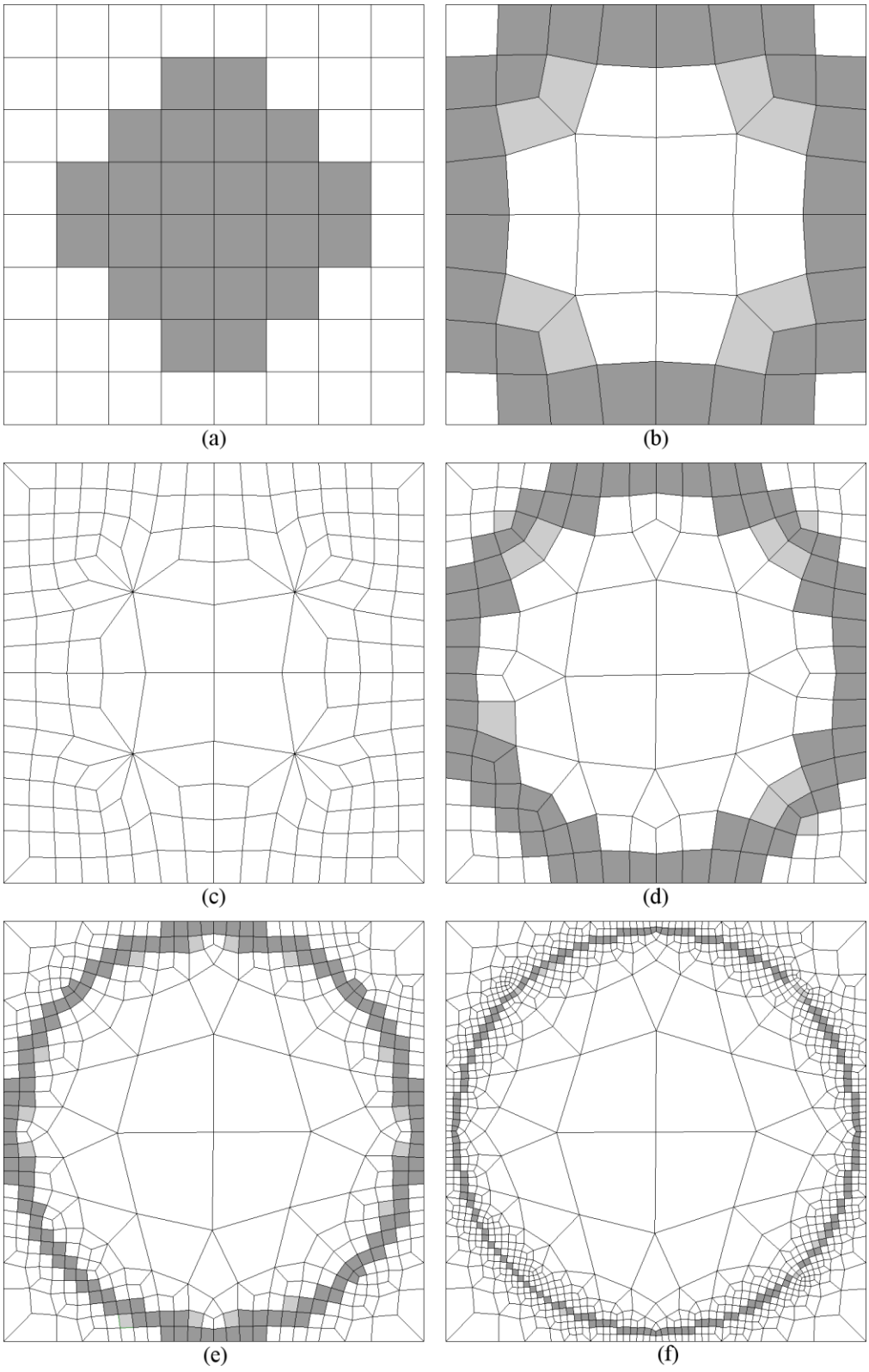

Figure 3-5: Coarsening, refinement, and quality improvement steps. (a) Iteration 1, coarsening region. (b) Iteration 2, refinement regions. (c) Iteration 2, before quality improvement. (d) Iteration 3, refinement regions. (e) Iteration 4, refinement regions. (f) Iteration 5, refinement regions.

After the first iteration of coarsening, the size ratio is re-calculated for all of the faces in the mesh in preparation for refinement. Any face with a size ratio greater than 1.0 is considered too large and requires refinement. This limit of 1.0 is not relaxed at any time throughout the algorithm; however, a dynamic refinement threshold, defined in Equation 3-5, is used to separate the elements requiring refinement into two categories.

$$t_r = 1.25 - 0.05 n_r \quad \textbf{(3-5)}$$

where t_r = refinement threshold

n_r = iteration number; 0,1,2…

Any face with a size ratio greater than 1.0 and the refinement threshold is considered a high-refine face, while any face with a size ratio greater than 1.0 but less than the refinement threshold is considered a low-refine face. In the refinement step, only the high-refine faces are refined, unless there are none, in which case all of the low-refine faces are refined. The purpose of the separation between faces is that the low-refine faces are often located very close to the high-refine faces and fall within their transition zones. Since the transition zone experiences some refinement through the insertion of templates, low-refine faces in the transition region are also refined. Similar to the coarsening threshold, the refinement threshold gradually shrinks the allowable range of low-refine face until the refinement threshold equals 1.0 and disappears. At this point, all faces with a size ratio greater than 1.0 are high-refine elements.

If at any time, less than 3% of the faces are considered low-refine faces and less than 0.5% of the faces are considered high-refine faces, refinement is deemed sufficient.

If these criteria are met, however, future iterations of refinement are not precluded as they are in the coarsening iterations.

In Figure 3-5(b) the high-refine faces are shaded dark gray and the low-refine faces are shaded light gray. As expected, the low-refine faces are in close proximity to the high-refine faces and, in fact, fall in the transition zone. The mesh topology resulting from the refinement of the high-refine faces is shown in Figure 3-5(c). Note the creation of four 9-valence nodes as a result of the refinement of the irregular high-refine region. Because of the formation of topologies like this, the quadrilateral improvement algorithm is applied after each instance of refinement. In the clean-up procedure, nodes with a valence greater than 6 are considered unacceptable and the mesh topology is changed to remove the high valence. The new mesh topology following the clean-up algorithm is shown in Figure 3-5(d) where the unacceptable 9-valence nodes have been reduced to acceptable 5-valence nodes. The remainder of the algorithm iteratively coarsens and refines the mesh until sufficient coarsening and refinement have both taken place.

At this point in the example, after the first refinement step, it was determined that fewer than 10% of the faces had a size ratio that warranted coarsening; therefore, coarsening is now considered complete for all future iterations. Following this completion of coarsening, only refinement steps occur. Figures 3-5(d-f) show the successive refinement iterations for the remainder of this example. The final adapted mesh is shown in Figure 3-6.

Table 3-1 provides the distribution of size ratios of the faces in the final mesh. Note that nearly all of the faces have a size ratio less than 1.0 and are therefore smaller than desired. Since the goal of this adaptation is to provide elements close to the desired

size, but not larger, this is a good result. It is not surprising, however, that some of the elements are too large since this method uses an isotropic smoothing scheme which ignores the desired size specified by the sizing function.

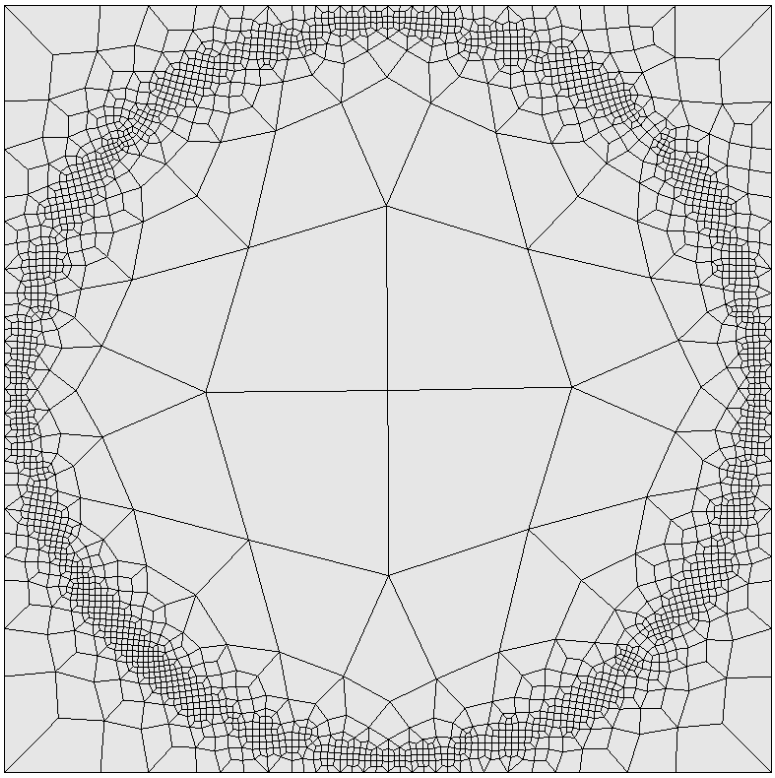

Figure 3-6: Adapted mesh of circular load on plane surface.

Table 3-1: Adaptation results of circular load on plane surface.

Size Ratio, f_s	Number of Faces	Percent of Total	Size Ratio, f_s	Number of Faces	Percent of Total
< 0.5	1485	61.0%	1.0 - 1.1	36	1.5%
0.5 - 0.8	708	29.1%	1.1 - 1.2	14	0.6%
0.8 - 0.9	114	4.7%	1.2 - 1.5	0	0.0%
0.9 - 1.0	77	3.2%	> 1.5	1	0.0%
	2384	97.9%		51	2.1%

4 Examples

The examples in this chapter are provided to show the results of this new quadrilateral adaptation scheme on a range of finite element models. Each of the four examples shows the initial mesh and a contour plot of the sizing function, as well as the mesh after adaptation and a table with data showing the results of the adaptation. The fourth example, a plate with a hole, also provides results from a finite element analysis of the plate under a tensile load. In each example the original mesh was created with the paving algorithm in the mesh generation software package, CUBIT [34].

In these examples, the size ratio data resulting from the adaptation technique are very similar. In each case, 2% or fewer of the faces are larger than their desired size and nearly all of the larger faces are within 10% of the target. This is a promising result considering the goal is to make sure most, if not all, of the elements are smaller than the desired size. The primary reason that there are some faces that are too large is that the smoothing algorithm used as part of the clean-up operations does not take into account the sizing function and may work against the desired size.

Even though there are a few faces with a size ratio greater than 1.0, more than 80% of the elements have a size ratio of less than 0.8, suggesting that this adaptation method over-refined the meshes by adding more elements than were necessary. This over-refinement, however, is to be expected since the quadtree refinement scheme

divides all faces in the refinement region into four, resulting in a reduction of interval size by half. Additionally, the transition zone around the region is refined by adding templates to ensure a conformal mesh.

4.1 Plate with Multiple Holes

The model of a plate with multiple holes is shown in the left panel of Figure 4-1 and a contour plot of the sizing function used to adapt this mesh is shown in the right panel. This sizing function specifies an increased element density immediately around the holes with an even higher density at each end of the oblong hole. The sizing function also specifies a lower element density in the corners of the mesh away from the holes and the top center between the holes.

The adapted mesh is shown in Figure 4-2 and it can easily be seen that the sizing function was successful in providing increased element density near the holes and decreasing element density in the specified regions. Table 4-1 shows the distribution of size ratios through the mesh. As mentioned previously, the vast majority of faces are too small, and most of those are between 50% and 80% of the desired size.

Table 4-1: Adaptation results of plate with holes example.

Size Ratio, f_s	Number of Faces	Percent of Total	Size Ratio, f_s	Number of Faces	Percent of Total
< 0.5	627	32.5%	1.0 - 1.1	19	1.0%
0.5 - 0.8	1060	55.0%	1.1 - 1.2	1	0.1%
0.8 - 0.9	160	8.3%	1.2 - 1.5	0	0.0%
0.9 - 1.0	60	3.1%	> 1.5	0	0.0%
	1907	99.0%		20	1.0%

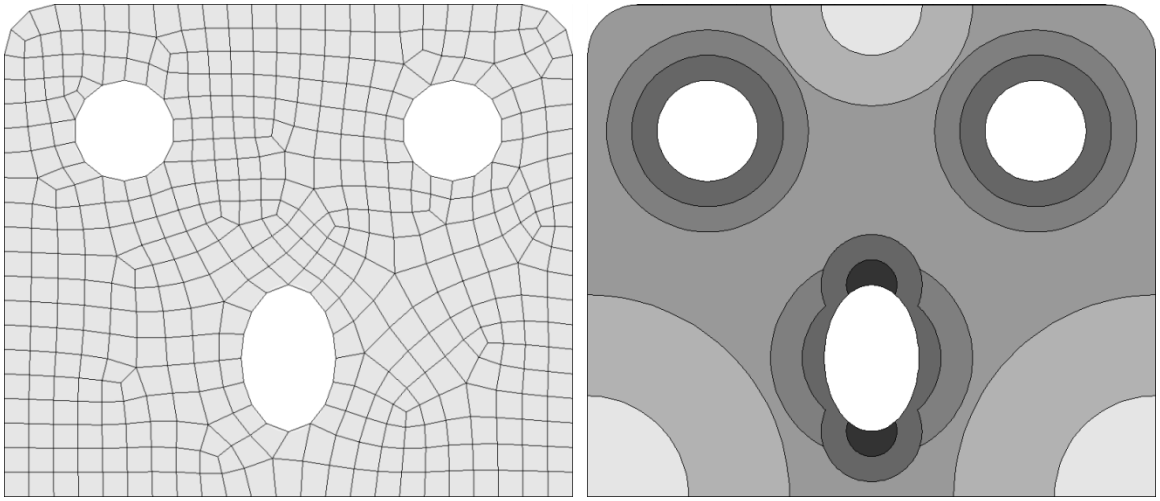

Figure 4-1: Original mesh of plate with holes (left) and contour plot of sizing function (right).

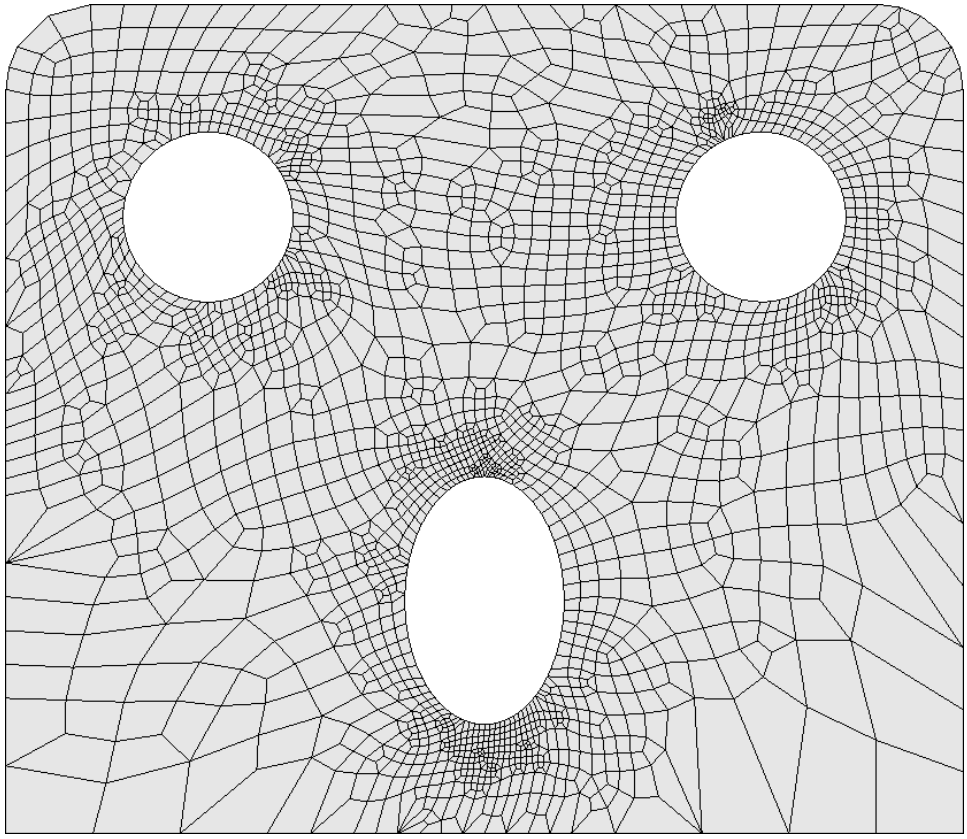

Figure 4-2: Adapted mesh of plate with holes.

4.2 Nosecone

The nosecone in this example is a non-planar surface with a paved quadrilateral mesh. Figure 4-3 shows an isotropic view the original mesh and a side view of the sizing function. The original element size in this example was 1.0 and the desired size ranged from 0.1 to 3. The sizing function used for this adaptation specified a high element density at the tip of the nosecone where the curvature is high and low element density away from the tip at the base. This example also illustrates how a sizing function might be used to adapt a mesh based on geometric characteristics of the model. In addition to the refinement near the tip of the object, note the difference in element size along the curve at the base of the nosecone.

Table 4-2 shows the distribution of size ratios through the mesh. In this example, fewer than 3% of the faces are too large and fewer than 7% of the faces are less than half of the desired size. This example also provides very good results with respect to not over-refining the mesh.

Table 4-2: Adaptation results of nosecone example.

Size Ratio, f_s	Number of Faces	Percent of Total	Size Ratio, f_s	Number of Faces	Percent of Total
< 0.5	132	6.5%	1.0 - 1.1	50	2.4%
0.5 - 0.8	1494	73.0%	1.1 - 1.2	1	0.0%
0.8 - 0.9	233	11.4%	1.2 - 1.5	0	0.0%
0.9 - 1.0	136	6.6%	> 1.5	0	0.0%
	1995	97.5%		51	2.5%

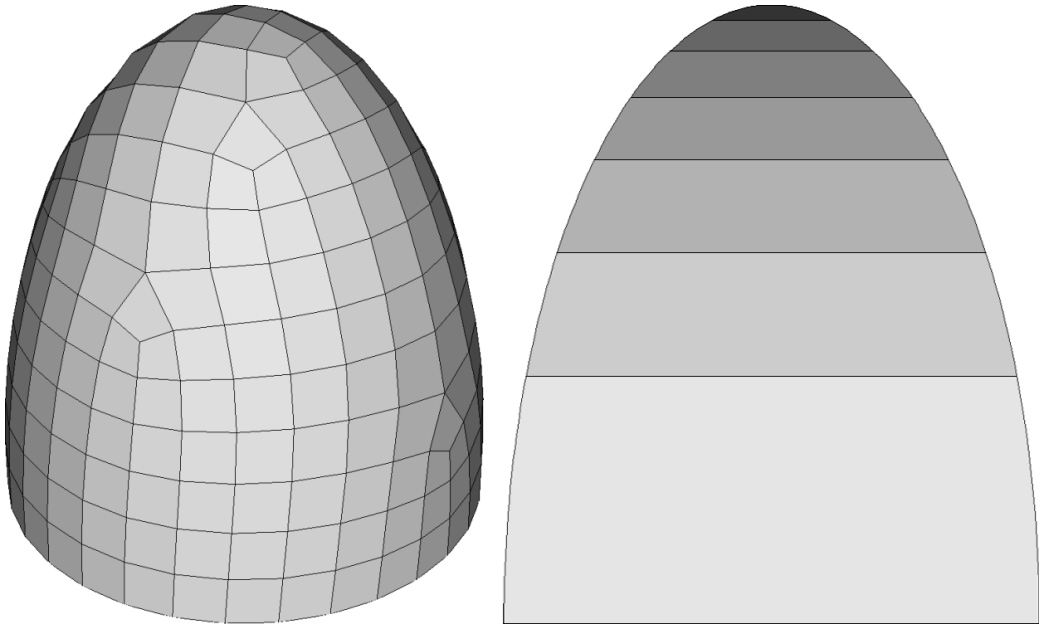

Figure 4-3: Original mesh of nosecone (left) and contour plot of sizing function (right).

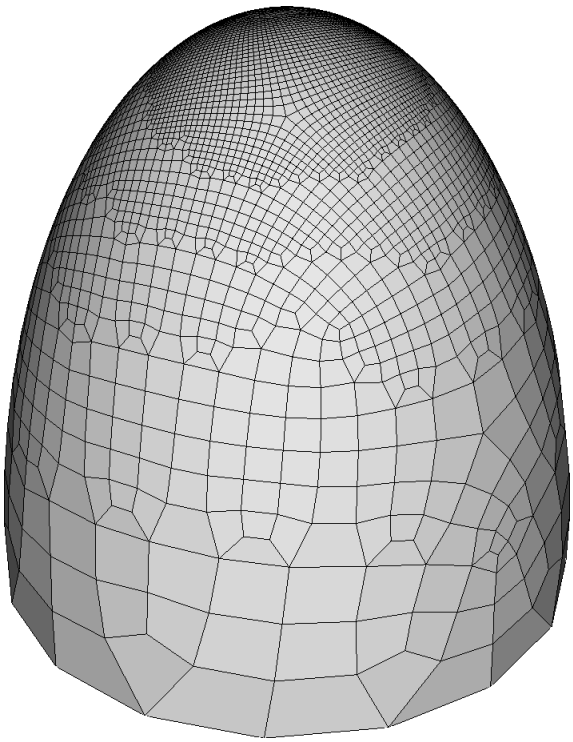

Figure 4-4: Adapted mesh of nosecone.

4.3 Gear

The gear in this example is a planar surface with a base mesh shown in the left panel of Figure 4-5. In a finite element model of a gear, it is likely that there will be stress concentrations between the teeth, where they connect with the main body of the gear. The sizing function is shown in the right panel of Figure 4-5 and specifies a reduction in mesh density near the center of the gear and a high mesh density where the teeth connect to the body of the gear.

The adapted mesh can be seen in Figure 4-6. Note the reduction in intervals along the inside radius of the gear as a result of chord removals. Table 4-3 shows the distribution of size ratios through the mesh. In this example, only 2% of the faces are too large. This example also provided very good results with respect to not over-refining the mesh and only about 10% of the quadrilateral faces had a size ratio less than 0.5 while more than 70% of the faces had a size ratio between 0.5 and 0.8.

Table 4-3: Adaptation results of gear example.

Size Ratio, f_s	Number of Faces	Percent of Total	Size Ratio, f_s	Number of Faces	Percent of Total
< 0.5	274	10.9%	1.0 - 1.1	43	1.7%
0.5 - 0.8	1788	71.1%	1.1 - 1.2	5	0.2%
0.8 - 0.9	281	11.2%	1.2 - 1.5	3	0.1%
0.9 - 1.0	120	4.8%	> 1.5	0	0.0%
	2463	98.0%		51	2.0%

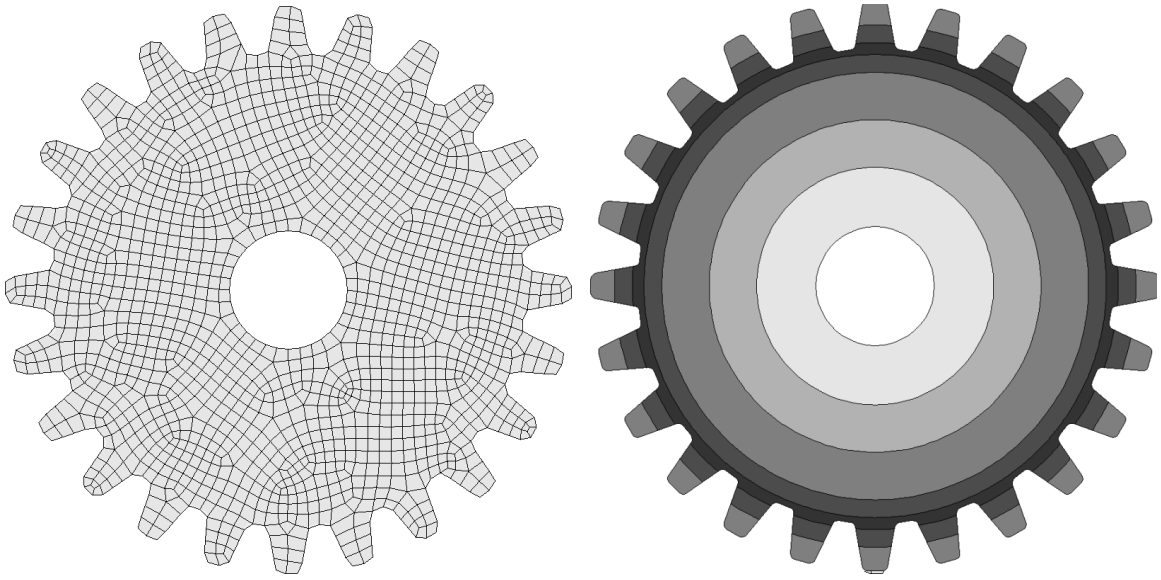

Figure 4-5: Original mesh of gear (left) and contour plot of sizing function (right).

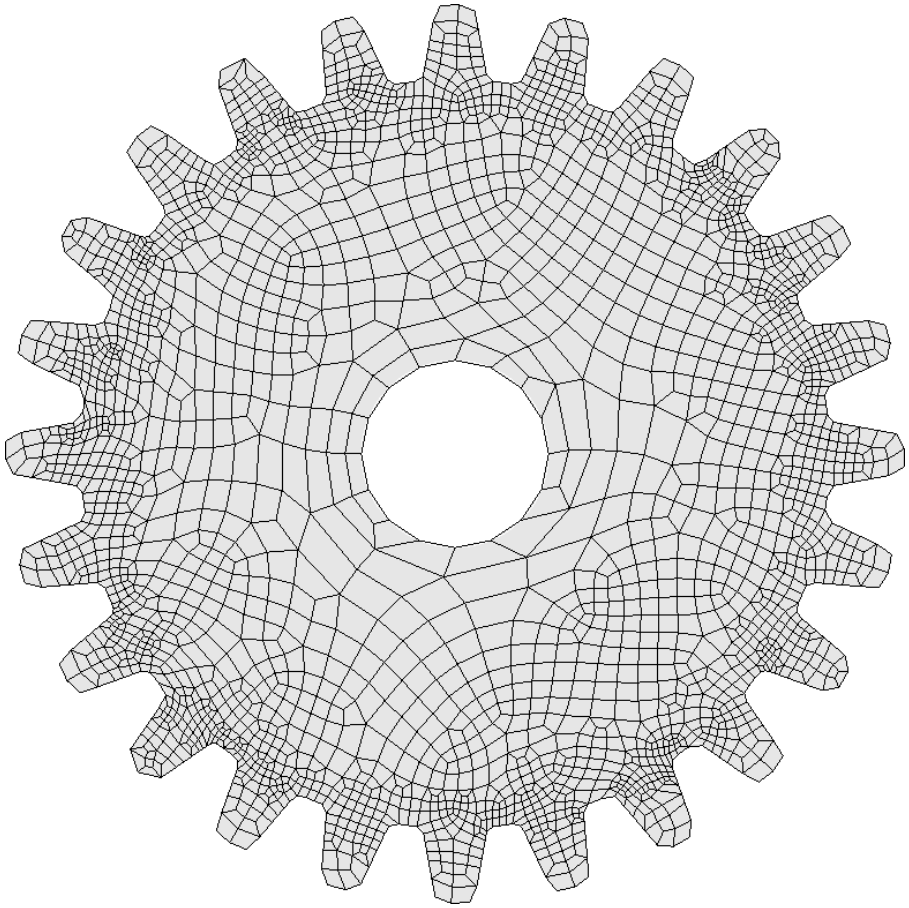

Figure 4-6: Adapted mesh of gear.

4.4 Plate with Hole in Tension

This example models a plate with a hole under a tensile loading, as shown in Figure 4-7. Due to symmetry of both the geometry and loads, this problem can be reduced to an analysis of a quarter of the plate, denoted by the shaded region. The three locations, A, B, and C, have been marked on the diagram where displacement results have been recorded after an analysis using the finite element program, ADINA [35].

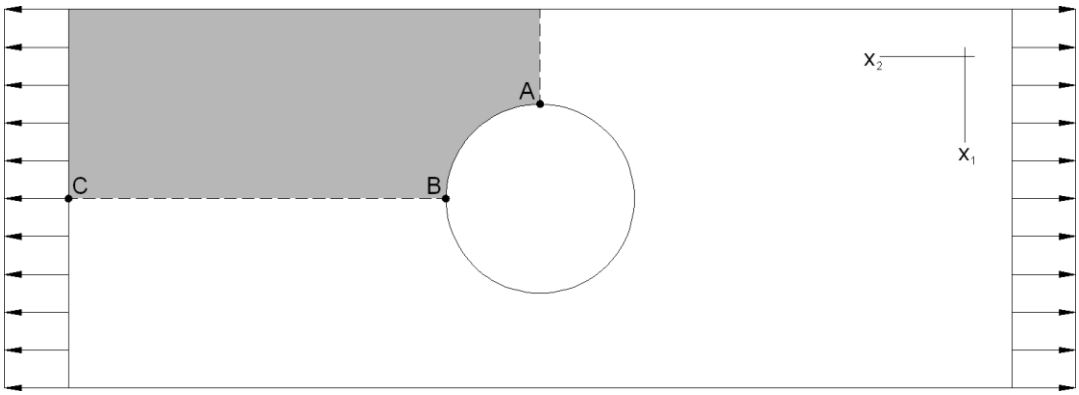

Figure 4-7: Model of plate with hole in tension.

The initial mesh of this example is shown in Figure 4-8 with an average element size of about 0.6. The sizing function used to adapt this mesh was based off of a posteriori stress error estimates [36] provided by the analysis of the original mesh in ADINA as shown in Figure 4-9. In this figure, the red indicates high error and the blue indicates little error. The sizing function determined from this analysis specified an element size of 0.04 in the areas of highest error ranging to an element size of 1.5 in areas with little error. The mesh resulting from the adaptation procedure is shown in Figure 4-10 and size ratio results of the adaptation are provided in Table 4-4.

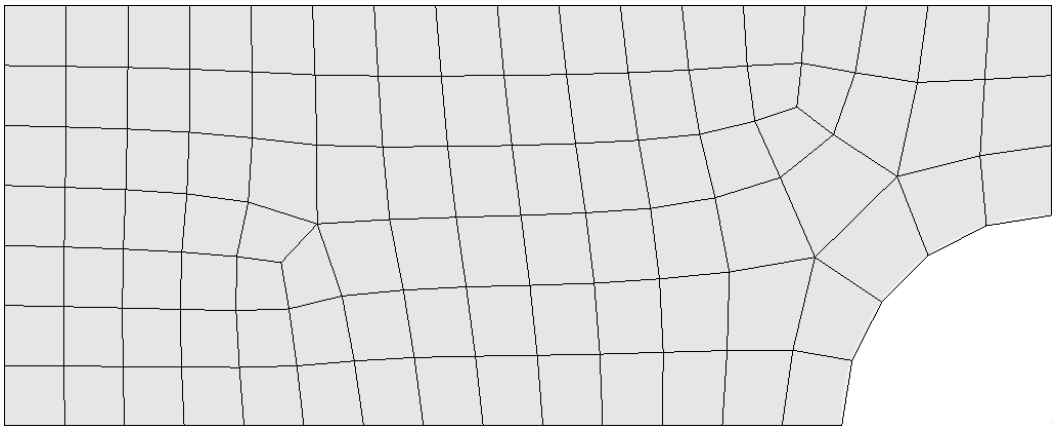

Figure 4-8: Original mesh of plate with hole in tension.

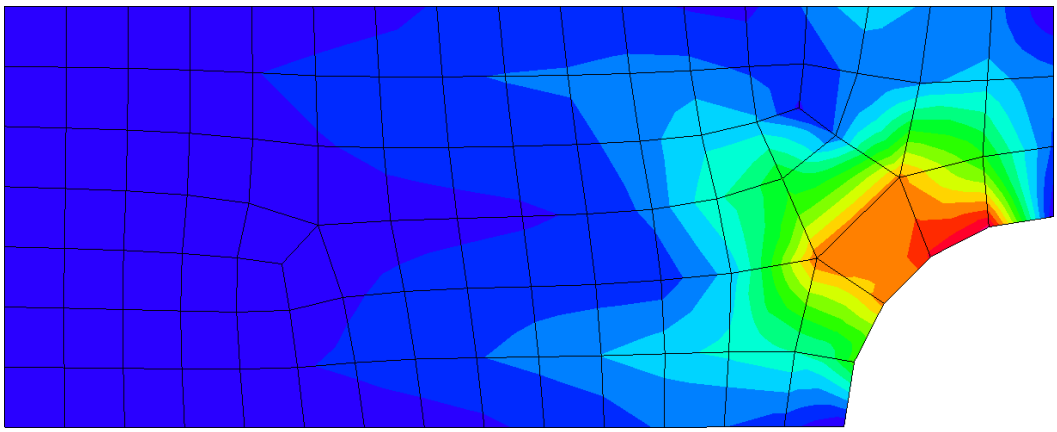

Figure 4-9: Band plot of stress error used to define sizing function for plate with hole in tension.

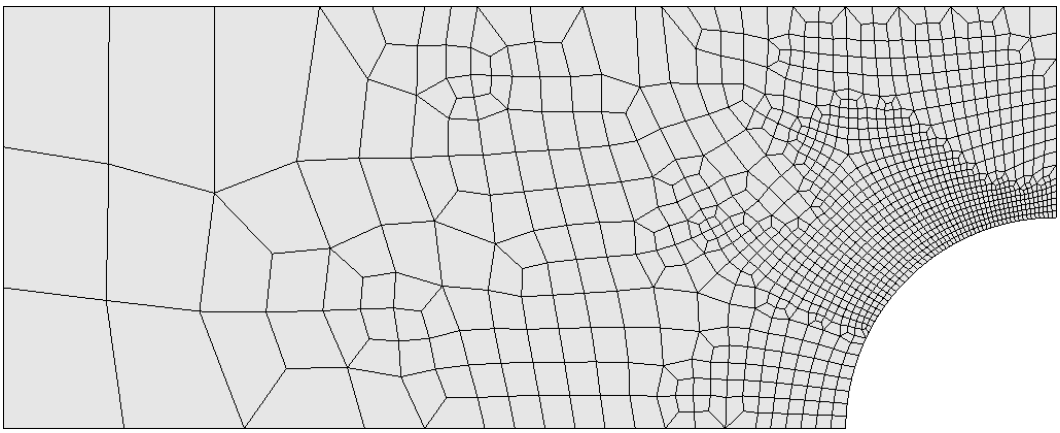

Figure 4-10: Adapted mesh of plate with hole in tension.

Table 4-4: Adaptation results of plate with hole in tension.

Size Ratio, f_s	Number of Faces	Percent of Total	Size Ratio, f_s	Number of Faces	Percent of Total
< 0.5	93	7.9%	1.0 - 1.1	7	0.6%
0.5 - 0.8	924	79.0%	1.1 - 1.2	0	0.0%
0.8 - 0.9	120	10.3%	1.2 - 1.5	0	0.0%
0.9 - 1.0	26	2.2%	> 1.5	0	0.0%
	1163	99.4%		7	0.6%

In addition to the initial coarse mesh and the adapted mesh, the quarter-plate was meshed with two other grids, each much more fine than the coarse base mesh. These meshes were used to show convergence to a solution as well as to compare error between analyses of each of the meshes. Band plots of the effective stress error found from a finite element analysis are shown in Figure 4-11. Note the significant reduction in error that occurs within the adapted mesh.

In addition to these band plots, numerical data comparing the meshes can be found in Table 4-5, which provides a comparison of mesh statistics of the four different meshes and numerical results at selected locations in each mesh. It also shows the maximum estimated error for each mesh. It is obvious by the similarity in stress and displacement results between the medium and fine meshes that the solutions are converging to values close to those recorded by the fine mesh. At each analysis location, the displacement results from the adapted mesh are within 1% of the solutions taken from the fine mesh. In fact, it appears that the adapted mesh provided a more accurate result for the maximum effective stress in the model.

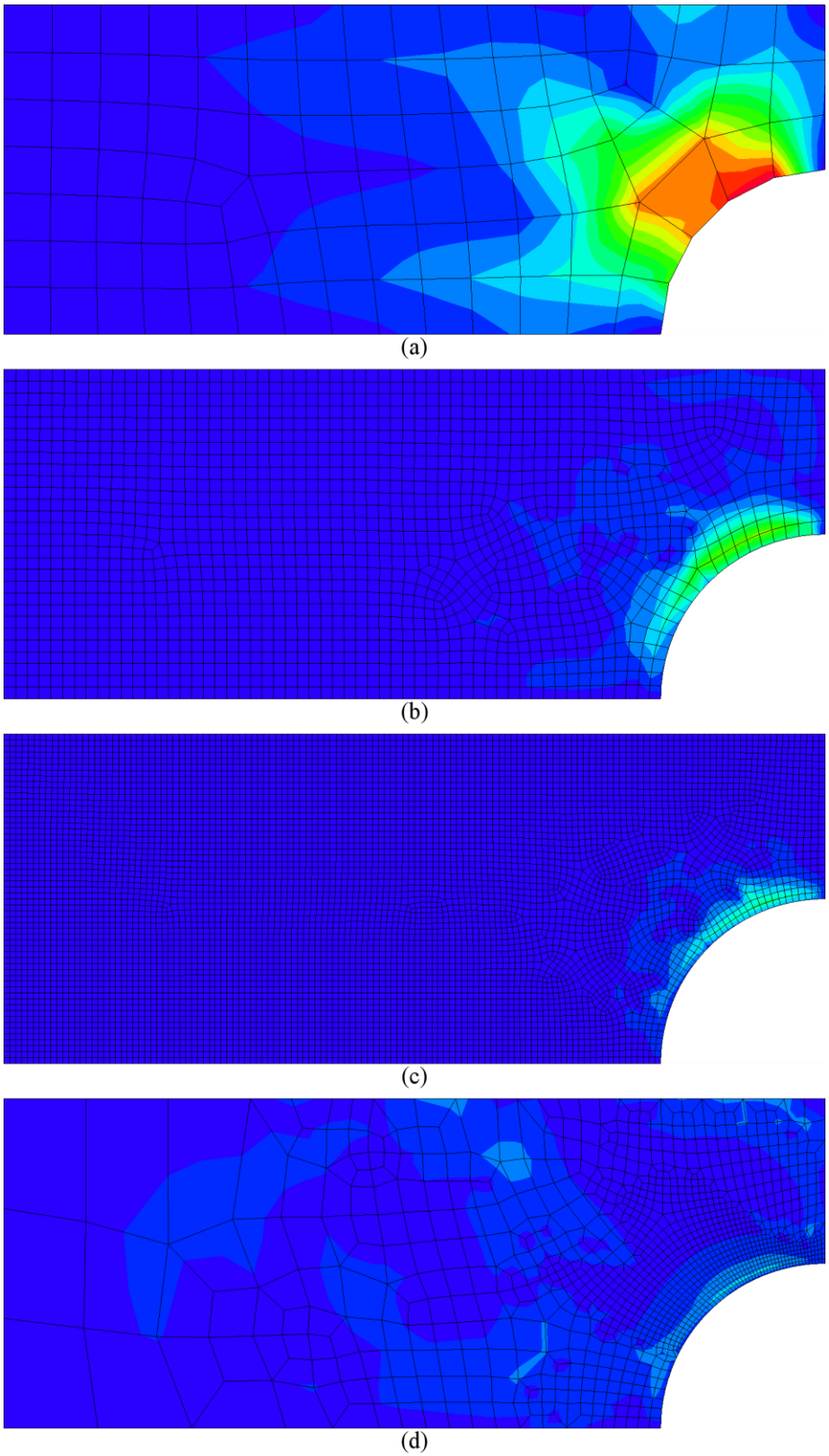

Figure 4-11: Band plots of effective stress error for plate with hole in tension ranging from 0.0 (blue) to 32.0 (red). (a) Coarse mesh, (b) Medium mesh, (c) Fine mesh, (d) Adapted mesh.

Table 4-5: Finite element analysis results of plate with hole in tension.

	Mesh Info		Effective Stress		Displacements		
	Nodes	Faces	Max Value	Max Error	Δx_1 at A	Δx_2 at B	Δx_2 at C
Coarse	121	97	94.3	32.08	0.00143	0.00279	0.00467
Medium	1770	1677	106.5	19.23	0.00154	0.00287	0.00474
Fine	6973	6788	107.9	12.33	0.00155	0.00288	0.00475
Adapted	1232	1170	108.3	7.53	0.00154	0.00287	0.00474

The maximum estimated stress error is significantly reduced by the adapted mesh. These results are significant when considering that the adapted mesh has only 70% of the nodes in the medium mesh and fewer than 20% of the nodes in the fine mesh. Not only does the adapted mesh provide virtually equal displacement values and superior stress values, it does so with fewer elements while reducing the estimated error. Although this is a small problem and the time savings were negligible, the savings of computational effort on a large problem can be significant.

The results of this analysis not only show the effectiveness of this adaptation algorithm in providing an efficient solution to a computational mechanics problem, but also the importance of mesh adaptation generally in finite element problems.

5 Conclusion

The ability to adapt a finite element mesh is critical to providing an efficient analysis to many finite element problems. Although there are currently effective quadrilateral adaptation techniques available, none of them are truly general in that they can modify the element density to match a sizing function by adding and removing elements without having to re-mesh all or part of the domain.

Recent developments in localized, automated quadrilateral coarsening have made it possible to combine quadrilateral refinement, coarsening, and quality improvement techniques into a conformal, all-quadrilateral adaptation method. Given a sizing function, this new adaptation technique iteratively coarsens and refines the mesh domain to provide a mesh with an element that matches the sizing function. As shown in examples, this method is an effective way to streamline the computational analysis of a finite element mesh by providing high element density in areas of the mesh that require high accuracy or geometric resolution while removing elements in less important areas of the mesh to decrease element density and save computation time.

5.1 Further Research

The adaptation technique described in this thesis effectively adds and removes elements resulting in an adapted mesh that improves accuracy or resolution where needed

while improving the efficiency of the analysis by removing elements away from the area of interest. Although the results shown in this thesis are promising, there are still improvements that can be made and more research that can be done.

One way to improve this algorithm is to provide adaptive smoothing with refinement and coarsening. The smoothing technique currently employed in this algorithm attempts to improve the quality of the mesh by re-distributing the nodes, but does not take into account the desired element size specified by the sizing function. In fact, the smoothing algorithm may work against the size function by attempting to create a uniformly sized mesh while the sizing function has specified a mesh with varying element density. Coupling the current *h*-adaptation method with an *r*-adaptation technique that considers the element size specified by the sizing function [37] would be a major improvement to this algorithm.

In some finite element applications, particularly computational fluid dynamics problems, anisotropic elements with a high aspect ratio are desired at mesh boundaries. This adaptation technique does not take isotropy into account and adds or removes elements based solely on their size. By more selectively choosing where to add elements, or even applying chord dicing capabilities, this adaptation method could be modified to allow for anisotropy in the mesh.

One purpose of this research was to provide a springboard into the development of an automated all-hexahedral mesh adaptation algorithm. Recent developments have been made in conformal hexahedral refinement [4, 38] that provide localized refinement and are robust on both structured and unstructured hexahedral meshes. Additional developments have been made in automated hexahedral coarsening as well. Woodbury

[39] recently introduced a new method that provides localized conformal coarsening to an all-hexahedral mesh. Woodbury's method isolates the coarsening region through pillowing and then uses chord-collapse operations to redirect hexahedral sheets so they are located entirely within the desired coarsening region. Additionally, this method does account for boundary and surface coarsening and therefore does not have the same limitations as the AQCRC algorithm used for quadrilateral adaptation in this thesis. One potential difficulty in the development of an automated hexahedral adaptation scheme, however, is providing quality improvement operations to ensure a high quality mesh. The improvement operators used in quadrilateral mesh improvement do not extend directly into 3-dimensions and topological restrictions in hexahedra make local topology changes very difficult.

References

1. J. Wan, S. Kocak, and M.S. Shephard, *Automated Adaptive Forming Simulations.* Proceedings, 12th International Meshing Roundtable, 2003: p. 323-334.

2. R. Drake and V.S. Manoranjan, *A Method of Dynamic Mesh Adaptation.* International Journal for Numerical Methods in Engineering, 1996. 39: p. 939-949.

3. X.Y. Li, S.H. Teng, and A. Ungor, *Simultaneous Refinement and Coarsening: Adaptive Meshing with Moving Boundaries.* Proceedings, 7th International Meshing Roundtable, 1998: p. 201-210.

4. M. Parrish, M. Borden, M.L. Staten, and S.E. Benzley, *A Selective Approach to Conformal Refinement of Unstructured Hexahedral Finite Element Meshes.* Proceedings, 16th International Meshing Roundtable, 2007: p. 251-268.

5. M.W. Dewey, *Automated Quadrilateral Coarsening by Ring Collapse.* Master's Thesis, Brigham Young University, 2008.

6. B.D. Anderson, J.F. Shepherd, J. Daniels, and S.E. Benzley, *Quadrilateral Mesh Improvement.* Research Note, 17th International Meshing Roundtable, 2008.

7. O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu, *The finite element method: its basis and fundamentals*. 6 ed. 2005: Butterworth-Heinemann.

8. D.W. Pepper and X. Wang, *Comparison of h-, p-, and hp-adaptation for convective heat transfer.* Computational Methods and Experimental Measurements XIII, 2007. 46: p. 495-504.

9. T.D. Blacker and M.B. Stephenson, *Paving: A new approach to automated quadrilateral mesh generation.* International Journal for Numerical Methods in Engineering, 1991. 32(4): p. 811-847.

10. W.R. Quadros, V. Vyas, M. Brewer, S.J. Owen, and K. Shimada, *A computational framework for generating sizing function in assembly meshing.* Proceedings, 14th International Meshing Roundtable, 2005: p. 55-72.

11. H. Borouchaki and P.J. Frey, *Optimization Tools for Adaptive Surface Meshing.* AMD-Vol. 220 Trends in Unstructured Mesh Generation, ASME, 1997: p. 81-88.

12. H. Borouchaki and P.J. Frey, *Adaptive Triangular-Quadrilateral Mesh Generation.* International Journal for Numerical Methods in Engineering, 1998. 41: p. 915-934.

13. Y.T. Feng and D. Peric, *Coarse mesh evolution strategies in the Galerkin multigrid method with adaptive remeshing for geometrically non-linear problems.* International Journal for Numerical Methods in Engineering, 2000. 49: p. 547-571.

14. J.G. Carpenter and D.S. McRae, *Improving Grid "Quality" through Grid Adaptation.* Proceedings, 6th International Conference on Grid Generation in Computational Field Simulation, 1998: p. 351-360.

15. Y. Ito, A.M. Shih, R.P. Koomullil, and B.K. Soni, *A Solution-Based Adaptive Redistribution Method for Unstructured Meshes.* Proceedings, 15th International Meshing Roundtable, 2006: p. 147-161.

16. S.K. Khatti, *An effective quadrilateral mesh adaptation.* Journal of Zhejiang University SCIENCE A, 2006. 7: p. 2018-2021.

17. B.K. Soni, R.P. Koomullil, D.S. Thompson, and H. Thornburg, *Solution Adatpive Grid Strategies Based on Point Redistribution.* Computer Methods in Applied Mechanics and Engineering, 2000. 189: p. 1183-1204.

18. M.E. Botkin and H.P. Wang, *An adaptive mesh refinement of quadrilateral finite element meshes based upon an a posteriori error estimation of quantities of interest: modal response.* Engineering with Computers, 2004. 20: p. 38-44.

19. J.R. Tristano, Z. Chen, D.A. Hancq, and W. Kwok, *Fully Automatic Adaptive Mesh Refinement Integrated into the Solution Process.* Proceedings, 12th International Meshing Roundtable, 2003.

20. P.L. Baehmann and M.S. Shephard, *Adaptive Multiple-Level h-Refinement in Automated Finite Element Analyses.* Engineering with Computers, 1989. 5: p. 235-247.

21. A. Diaz-Morcillo, L. Nuno, J.V. Balbastre, and D. Sanchez-Hernandez, *Adaptive Mesh Refinement in Electromagnetic Problems.* Proceedings, 9th International Meshing Roundtable, 2000: p. 147-155.

22. L. Branets and G.F. Carey, *Smoothing and Adaptive Redistribution for Grids with Irregular Valence and Hanging Nodes.* Proceedings, 13th International Meshing Roundtable, 2004: p. 333-344.

23. Y. Zhang and C. Bajaj, *Adaptive and Quality Quadrilateral/Hexahedral Meshing From Volumetric Data.* Proceedings, 13th International Meshing Roundtable, 2004: p. 365-376.

24. X. Jiao, A. Colombi, X. Ni, and J.C. Hart, *Anisotropic Mesh Adaptation for Evolving Triangulated Surfaces.* Proceedings, 15th International Meshing Roundtable, 2006: p. 173-190.

25. Y. Kallinderis and P. Vijayan, *Adaptive Refinement-Coarsening Scheme for Three-Dimensional Unstructured Meshes.* American Institute of Aeronautics and Astronautics Journal, 1993. 31: p. 1440-1447.

26. H.L. De Cougny and M.S. Shephard, *Parallel Refinement and Coarsening of Tetrahedral Meshes.* International Journal for Numerical Methods in Engineering, 1999. 46: p. 1101-1125.

27. S. Chalasani, D. Thompson, and B. Soni, *Topological Adaptivity for Mesh Quality Improvement.* Numerical Grid Generation in Computational Field Simulations, 2002: p. 107-116.

28. J.S. Sandhu, F.C.M. Menandro, and H. Liebowitz, *Hierarchical mesh adaptation of 2D quadrilateral elements.* Engineering Fracture Mechanics, 1995. 50: p. 727-735.

29. *Cubit 11.1 user documentation*. Sandia National Laboratories, Albuquerque, N.M. 2009; Available from: http://cubit.sandia.gov/documentation.html.

30. H. Borouchaki, F. Hecht, and P.J. Frey, *Mesh Gradation Control.* International Journal for Numerical Methods in Engineering, 1998. 13: p. 1143-1165.

31. P.-O. Persson, *Mesh size functions for implicit geometries and PDE-based gradient limiting.* Engineering with Computers, 2006. 22(2): p. 95-109.

32. R. Schneiders, *Refining Quadrilateral and Hexahedral Element Meshes.* Numerical Grid Generation in Computational Field Simulations, 1996. 1: p. 679-688.

33. M.L. Staten, S.E. Benzley, and M. Scott, *A Methodology for Quadrilateral Finite Element Mesh Coarsening.* Engineering with Computers, 2008. 24: p. 241-251.

34. *CUBIT 11.1 Geometry and Mesh Generation Toolkit*. 2009, Sandia National Laboratories, http://cubit.sandia.gov.

35. *ADINA-AUI 8.5.2*. 2008, ADINA R & D, Inc., http://www.adina.com.

36. T. Grätsch and K.-J. Bathe, *A posteriori error estimation techniques in practical finite element analysis.* Computers and Structures, 2005. 83: p. 235-265.

37. G. Hansen and A. Zardecki, *Unstructured surface mesh adaptation using the Laplace-Beltrami target metric approach.* Journal of Computational Physics, 2007. 225: p. 165-182.

38. N. Harris, *Conformal Refinement of All-Hexahedral Finite Element Meshes.* Master's Thesis, Brigham Young University, 2004.

39. A. Woodbury, *Localized Coarsening of Conforming All-Hexahedral Meshes.* Master's Thesis, Brigham Young University, 2008.

40. G.P. Nikishkov, *Finite element algorithm with adaptive quadtree-octree mesh refinement.* 12th Biennial Computational Techniques and Applications Conference, 2004.

41. J. Daniels, C. Silva, J. Shepherd, and E. Cohen, *Quadrilateral Mesh Simplification.* ACM Transactions on Graphics, 2008. 27(5).

42. P. Kinney, *Cleanup: Improving quadrilateral finite element meshes.* Proceedings, 6th International Meshing Roundtable, 1997: p. 449-461.

43. P.P. Pebay, D. Thompson, J.F. Shepherd, P.M. Knupp, C. Lisle, V.A. Magnotta, and N.M. Grosland, *New applications of the verdict library for standardized mesh verification: Pre, post and end-to-end processing.* Proceedings, 16th International Meshing Roundtable, 2007: p. 535-552.

44. P. Knupp, *Achieving finite element mesh quality via optimization of the jacobian matrix norm and associated quantities.* International Journal for Numerical Methods in Engineering, 2000: p. 1165-1185.

Appendix A. Quadrilateral Refinement

A straightforward way to refine a quadrilateral mesh is to subdivide the elements in the refinement region using a quadtree approach [40]. This method of quadrilateral refinement is illustrated in Figure A-1(b). Although this simple quadtree refinement is limited in that it introduces non-conforming elements into the mesh, it is a method of hierarchical refinement [28] that is easy to de-refine by removing quadtrees from the parent element.

If desired, the problem of non-conformity in quadtree refinement can be resolved by adding triangular elements to the mesh as shown in Figure A-1(c). This, however, violates the constraint that the resulting mesh be all-quadrilateral. To resolve the issue of having a quad-dominant mesh, elements can be further sub-divided to produce an all-quadrilateral mesh. This is shown in Figure A-1(d) where each triangle is divided into three quadrilaterals and the new edges are extended throughout the mesh to maintain conformity. Figure A-1(e) shows the results of subdividing all of the elements in the mesh to create a more uniform conforming mesh. Although these methods of further subdivision are valid solutions, they may provide more refinement than desired.

Another way to avoid the introduction of non-conforming or non-quadrilateral elements is by using a template based approach [4, 32, 38]. Although template based methods tend to introduce elements with a lower shape quality into the area around the

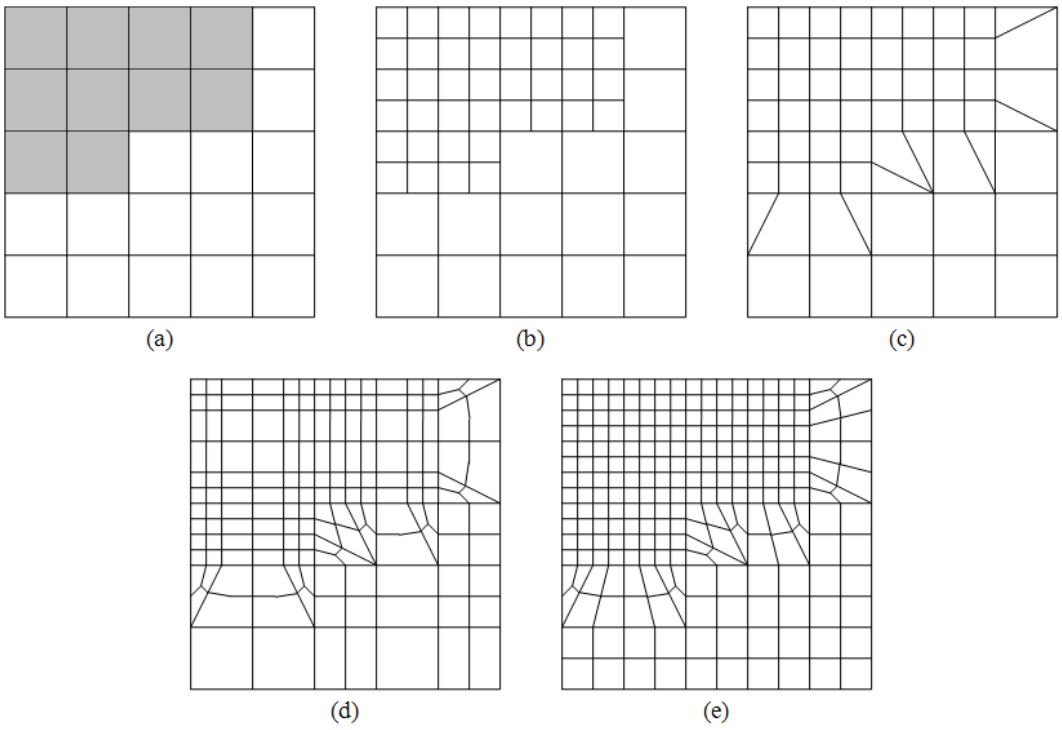

Figure A-1: Quadrilateral refinement. (a) Original mesh and shaded refinement region. (b) Refined non-conformal all-quadrilateral mesh. (c) Refined conformal quad-dominant mesh. (d) Quad-dominant mesh subdivided to create all-quadrilateral mesh. (e) Quad-dominant mesh subdivided further to create uniform all-quadrilateral mesh.

refinement region, they are still an effective way to refine a mesh and are often preferred over methods that create hanging nodes.

There are two main template based refinement methods; one uses a four element quadtree (referred to as 2-refinement) and the other uses a nine element quadtree (3-refinement). Refinement using these transition templates is shown in Figure A-2. Transition template refinement begins by dividing the elements in the region of interest using a 2- or 3-refinement quadtree. After the refinement region has been divided into the appropriate number of elements, the elements surrounding the refined region are marked as a transition zone and templates are inserted to maintain a conformal, all-quadrilateral mesh. Figure A-3 shows templates that are commonly used in both 2- and

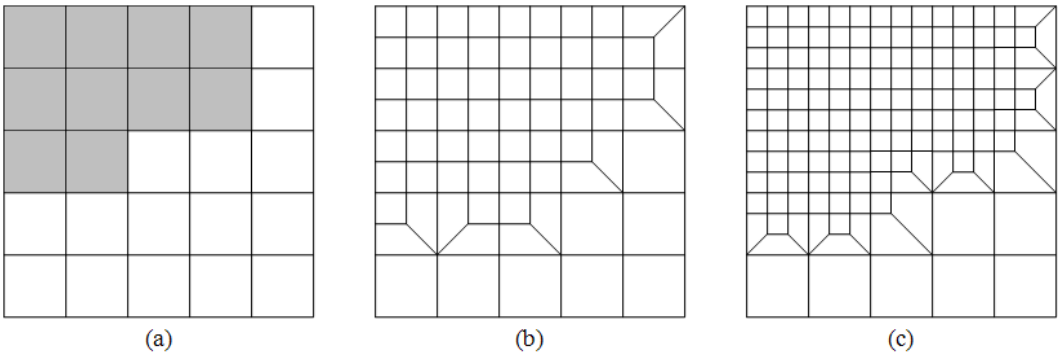

Figure A-2: Quadrilateral refinement with transition templates. (a) Original mesh and shaded refinement region. (b) Mesh refined with 2-refinement. (c) Mesh refined with 3-refinement.

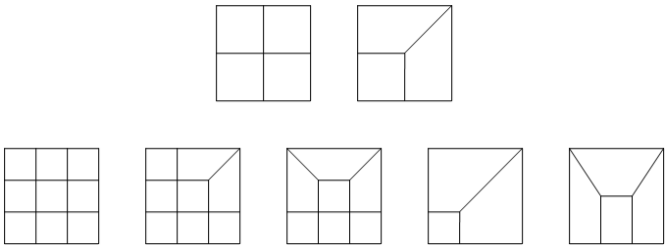

Figure A-3: Common refinement templates.

3-refinement. The two templates on the first row are used for 2-refinement and the five templates on the second row are used for 3-refinement.

Although the process is identical for 2- and 3-refinement, the number of elements generated and the size of the transition zone are different between the two methods because of the different templates. Since 2-refinement divides each interval in half, much less refinement takes place than with 3-refinement which divides each interval into thirds. 2-refinement also allows for a more gradual gradation from fine to coarse and makes it more likely that the resulting size is close to the desired size. However, since there are more templates available in 3- than in 2-refinement, 3-refinement offers more options for how much of the transition zone is affected by the refinement.

The differences in magnitude of refinement and size of the affected transition zone are illustrated in Figure A-4 where a single quadrilateral element is refined with 2- and 3-refinement. To show that several options exist when choosing templates for 3-refinement, the element in was refined by 3-refinement in two different ways. Note that in the 2-refinement example, Figure A-4(b), 15 elements are added and the transition zone affects six of the original quadrilaterals. In the first example of 3-refinement, Figure A-4(c), 20 elements are added and the transition zone only affects four of the original quadrilaterals. In the second example of 3-refinement, Figure A-4(d), 40 elements are added and the transition zone affects eight of the original quadrilaterals.

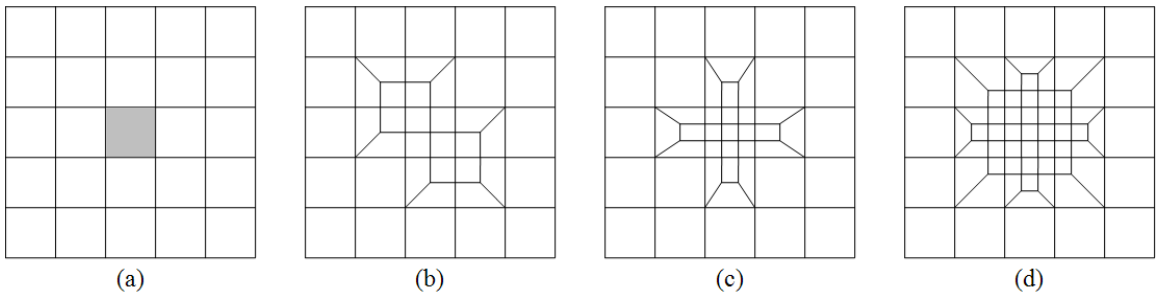

Figure A-4: Quadrilateral refinement of a single element. (a) Original mesh and shaded refinement region. (b) 2-refinement with templates. (c) One option of 3-refinement with templates. (d) Another option of 3-refinement with templates.

If more control over the magnitude of refinement is desired, 2-refinement is advantageous. However, if more control over the amount of area affected by the refinement is desired, 3-refinement is the best option. Since the adaptation technique developed in this thesis iteratively refines the mesh, there is no need to provide a high amount of refinement in a single iteration. 2-refinement can be used over multiple iterations to provide a sufficient amount of refinement.

Appendix B. Quadrilateral Coarsening

Quadrilateral coarsening (often referred to as simplification in literature related to computer graphics) is an area of meshing that has received relatively little attention. However, there have been recent developments made in improving and automating the coarsening process [5, 33, 41]. As shown in Figure B-1, basic coarsening techniques are accomplished by simply combining elements in the coarsening region. This simplistic coarsening method obviously has the same drawbacks as refining without templates. Additionally, by adding triangular elements, as shown in Figure B-1(c), we are actually back-tracking and refining elements that have just been coarsened.

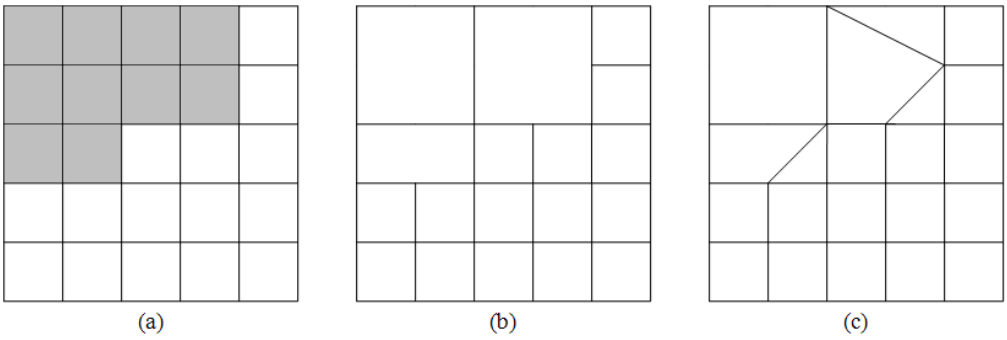

Figure B-1: Simple quadrilateral coarsening. (a) Original mesh and coarsening region. (b) Coarsened non-conformal all-quadrilateral mesh. (c) Coarsened conformal quad-dominant mesh.

Since the goal of coarsening is to reduce the number of elements in the coarsening region, Figure B-2 demonstrates that coarsening can be achieved by using local topology

operators such as the face close operation that remove elements while maintaining conformity in the mesh. Although this method is completely localized, it can result in high-valence nodes which are often indicative of poor element quality.

An extension of a single face close referred to as partial chord collapse is accomplished by performing several face close operations on a string of adjacent quadrilaterals as shown in Figure B-3. This method provides more coarsening than a single face close and can still be entirely localized to the coarsening region, but still often results in poor quality elements.

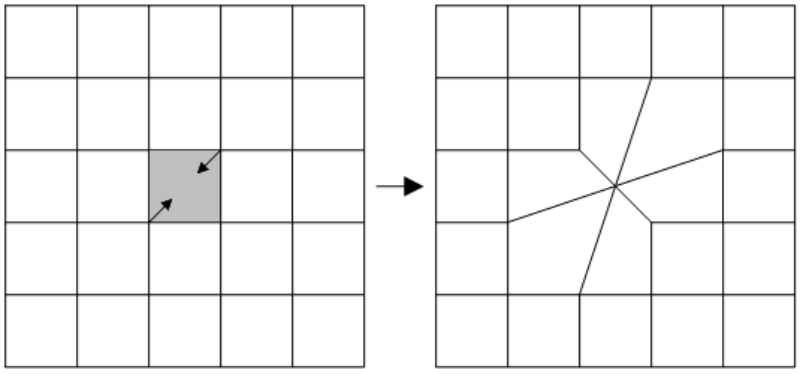

Figure B-2: Face close.

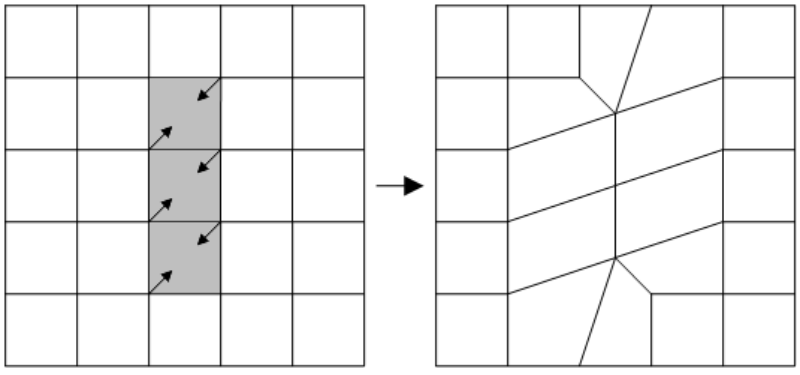

Figure B-3: Partial chord collapse.

In quadrilateral meshes, it is sometimes helpful to use the dual representation of the mesh. In the dual representation, a chord in a conformal, all-quadrilateral mesh is defined by connecting the midpoints of opposite edges of a quadrilateral element and continuing that chord in both directions, connecting opposite edges of faces until it either reaches the boundary of the mesh or circles back to an edge already marked by the chord. Almost any chord in a quadrilateral mesh can be removed, and will leave behind a mesh that is still conformal and all-quadrilateral. The removal of a quadrilateral chord is illustrated in Figure B-4. Since the removal of a chord reduces the number of quadrilateral elements in the mesh by the number of elements in the extracted chord and the resulting mesh meets conformity and element type requirements, it is an effective way to coarsen a quadrilateral mesh.

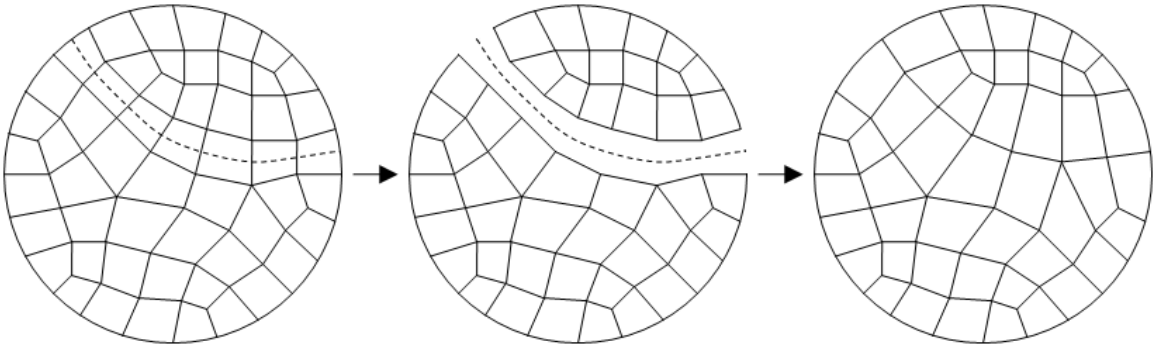

Figure B-4: Quadrilateral chord removal.

The main drawback with quadrilateral coarsening by chord removal is that this method does not guarantee coarsening local to the desired coarsening region. Completely localized coarsening by chord removal will only occur if the entire chord is contained within the coarsening region. In Figure B-5, the shaded coarsening region is coarsened

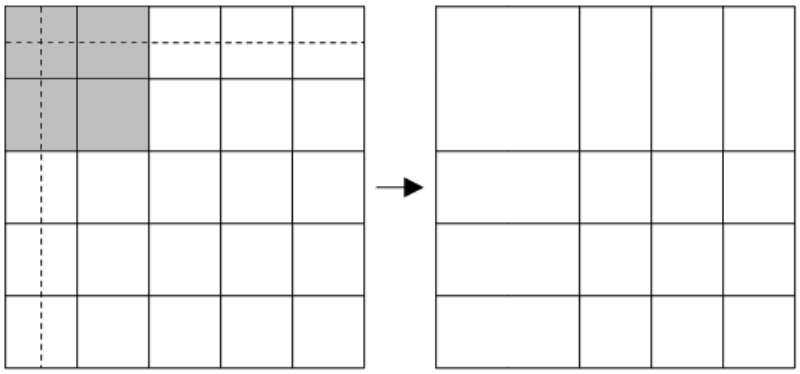

Figure B-5: Chord removal in two directions.

by the removal of the two chords marked with dashed lines. Note that the coarsened area of the mesh extends well beyond the desired coarsening region.

Staten, et al. [33] showed that by using specific topological operators (doublet insertion, face close, and edge swap), a chord can be re-directed to be contained entirely within the coarsening region, also illustrated in Figure B-6. In the first panel of the figure, the coarsening region is shaded and the chords bounding the region are shown as dashed lines. In the second panel, the chords in the upper left of the coarsening region are altered with the insertion of a doublet (a node with two edges connected to opposite nodes in the quadrilateral), the chords in the upper right of the coarsening region are altered with an edge swap, and the chords on the lower corners of the coarsening region are altered with face close operations. The third part of the figure shows the mesh after the redirected chord has been removed.

As an alternative method to redirecting and removing chords, Dewey [5] recently developed an algorithm called Automated Quadrilateral Coarsening by Ring Collapse (AQCRC) that provides coarsening without removing chords from the mesh. Instead, the algorithm creates a set of closed, concentric rings of adjacent quadrilaterals within the

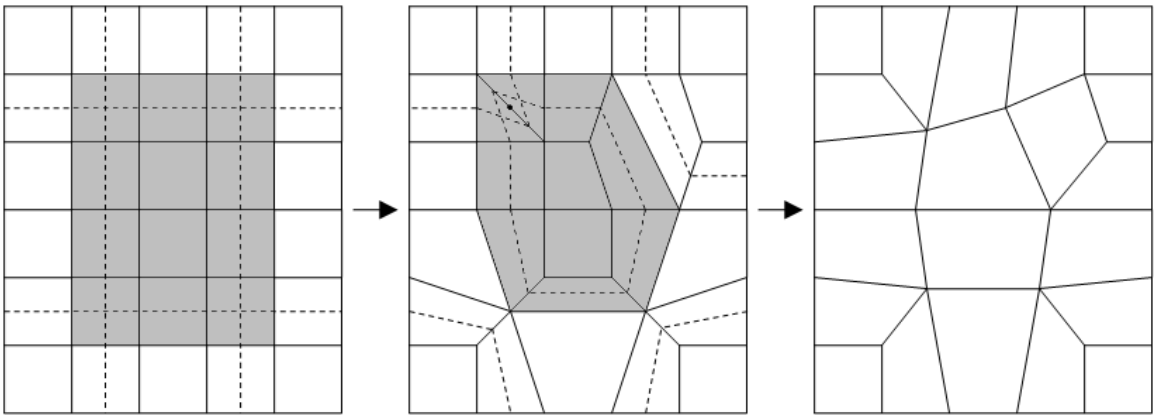

Figure B-6: Chord re-direction and removal.

coarsening region, selects rings from the set to be removed based on projected quality and magnitude of coarsening, and then removes the selected rings. Removal of coarsening rings, instead of quadrilateral chords, often results in poor quality elements in the mesh. However, the mesh quality can be significantly improved using clean-up operations [6].

A major assumption that Dewey makes in the AQCRC algorithm is that a coarsening ring is always closed ring. This assumption is made in order to maintain the current element size outside of the coarsening region. A drawback to this assumption appears when the boundary of the mesh that is not attached to any other meshed geometry is included at the edge of the coarsening region. In this case, the algorithm assumes that the quadrilaterals along the boundary cannot be modified in order to preserve the interval counts on the curve at the boundary of the mesh. It assumes that this boundary is already meshed at the correct size and does not require coarsening or there is another surface outside of the coarsening region that shares the curve in question that should not be altered. This case of the AQCRC algorithm is illustrated in Figure B-7.

Note that the shaded coarsening region includes the boundary of the mesh and the coarsening ring, shown as a dashed line, does not continue through the boundary but turns back to form a closed ring. Figure B-8 shows how the ring could still be formed and collapsed if the assumption of maintaining boundary intervals were relaxed.

Despite the drawback of the closed ring assumption, the AQCRC algorithm is still a very effective method to provide completely localized coarsening in an all-quadrilateral mesh and is used in this new adaptation scheme.

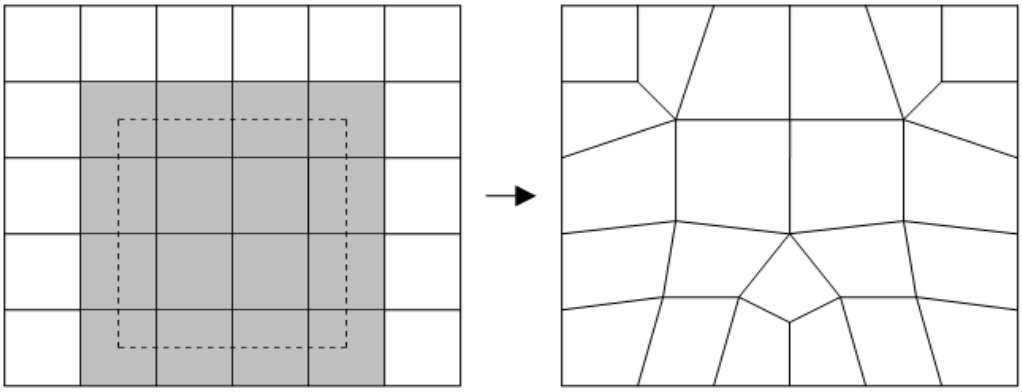

Figure B-7: AQCRC coarsening.

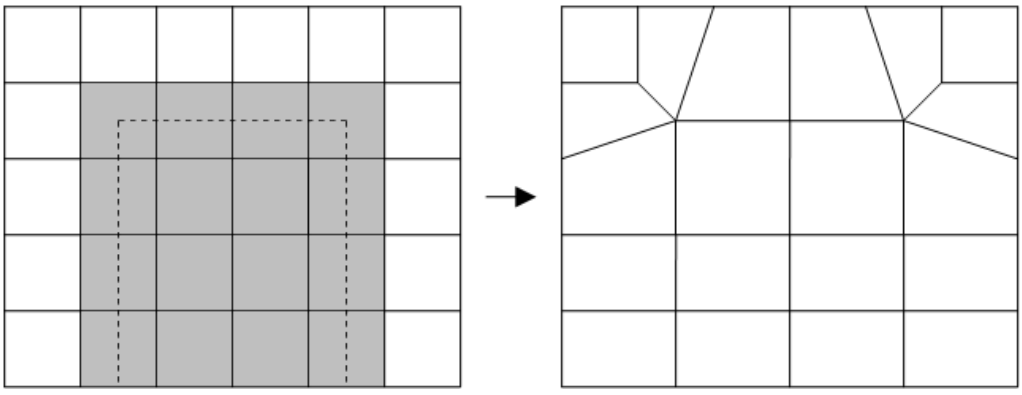

Figure B-8: Ring collapse with boundary coarsening.

Appendix C. Quadrilateral Mesh Improvement

As mentioned in the previous Appendix, quality improvement, (i.e. clean-up) through topological changes to the mesh is a necessary step in coarsening with the AQCRC algorithm. A quadrilateral improvement algorithm that improves the quality of the mesh through local topology modification was developed in conjunction with the AQCRC algorithm and has been implemented in its coarsening process [6]. This same quadrilateral improvement algorithm has also been employed in this new adaptation method to provide quality improvement in other parts of the adaptation process.

In addition to the formation of poor quality elements as a result of coarsening, high valence nodes may form as a result of the refinement of irregular regions. Figure C-1 shows a case where a concave refinement region forms a node with a valence of seven. This high-valence node can be removed with a face open procedure. This face open operation added one element to the mesh and reduced the valence of the node in question from seven to five. The removal of high valence nodes from a mesh is one of many capabilities of this quality improvement method.

The quadrilateral mesh improvement algorithm described in this appendix builds on the mesh improvement methods intended for parallel execution with the paving method [42] and introduces topological changes in an existing quadrilateral-only mesh to improve element shape and nodal valence. This discussion is intended to demonstrate the

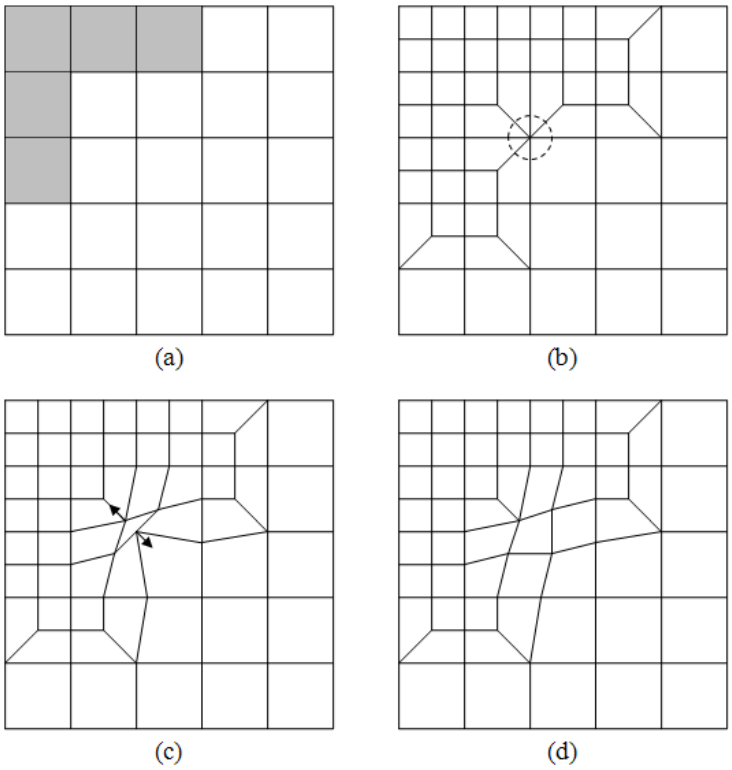

Figure C-1: Quality improvement required by refinement. (a) Shaded refinement region. (b) Seven-valence node formed from 2-refinement. (c) Mesh after quality improvement operation (face open on high-valence node). (d) Improved mesh.

effectiveness of utilizing mesh cleanup operations as a post-processing step to other mesh generation algorithms.

C.1 Quality Standards

Quadrilateral mesh improvement consists of altering the shape and/or topology of a mesh to obtain better characteristics. For most purposes, an ideal quadrilateral mesh exists when all quadrilaterals in the mesh are square in shape and each interior node within the mesh is connected to four quadrilaterals. The number of quadrilateral elements attached to each node is referred to as the node's valence. A structured quadrilateral mesh exists when every interior node of a quadrilateral mesh has a nodal

valence of 4, while an unstructured quadrilateral mesh accepts nodal valence other than four. Nodes with a valence of 6 or more are considered high valence nodes. With high valence nodes, the shape quality begins to deteriorate. Since the average angle between two adjacent edges with 6-valence nodes is less than or equal to 60°, these should be removed to provide a higher quality [42].

An unstructured quadrilateral mesh is improved topologically by reducing the number of unstructured nodal valences within the mesh. A quadrilateral mesh is improved geometrically by minimizing the deviation of each quadrilateral from square and equilateral. Geometric improvement is typically accomplished using nodal movement operations including mesh relaxation, mesh smoothing and mesh optimization techniques.

Nodal valence is used as an indicator of likely problem areas in the mesh. Since all of the interior nodes in a perfectly structured mesh have 4 emanating edges separated by 90°, anything other than that signifies an unstructured mesh. Nodes that have a valence of 3 or 5 are acceptable, but lower quality by varying the quadrilateral angles. Nodes with a valence of 2 indicate the presence of one or more degenerate elements, nodes with valence 6 are avoided, and nodes with a valence of 7 or more are not acceptable.

Many mesh quality metrics [43] exist to measure the quality of a quadrilateral mesh, however, we opt to use the scaled Jacobian [44]. This is a commonly used metric, utilized by computation simulation methods to describe the mapping of a given quadrilateral to an ideal quadrilateral element when determining the simulation shape functions. The scaled Jacobian metric quantifies the deviance of an element from a

perfect square. The acceptable range of this metric is assumed to be between 0.2 and 1.0 [29], where 1.0 corresponds to a perfect square.

C.2 Geometry Preservation

When performing local topological changes to a surface mesh, care must be taken not to alter the geometry of the model. Geometric features are identified a priori and template replacement that results in the removal of geometric features is disallowed. Additionally, steps are taken to preserve geometric characteristics that are not explicitly defined by curves or vertices. To locate implicit geometric characteristics, the normal of each pair of adjacent quadrilaterals is calculated and compared to adjacent quadrilateral elements. Large differences in the angles of the normal vectors of adjacent quadrilaterals indicate the presence of a geometric feature. If this is the case, the edge and nodes common to the quadrilaterals are treated as if they were on a geometric curve to preserve the feature.

C.3 Elementary Operators

Each of the quality improvement operations used in this algorithm is accomplished by applying one or more of the following elementary operators on specified topologies found within the mesh. These four operators are called quadrilateral edge swap, face close, face open, and doublet insertion and are shown in Figure C-2 through Figure C-5. Although there are some areas of the mesh where one or more of these operators cannot be used due to the presence of geometric curves, their benefit is that they guarantee to maintain a conformal, all-quadrilateral mesh after they are applied.

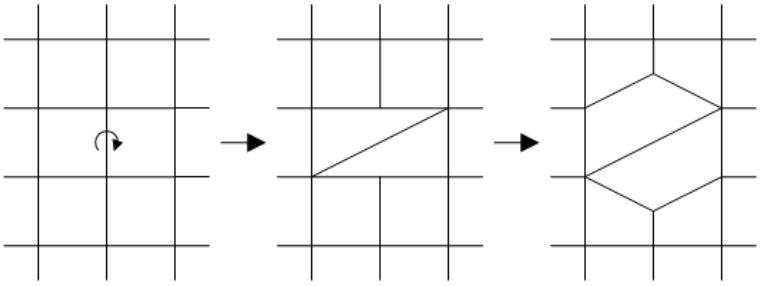

Figure C-2: Edge swap.

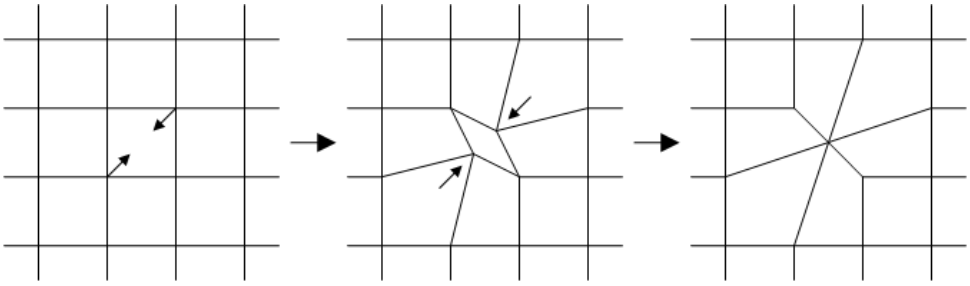

Figure C-3: Face close.

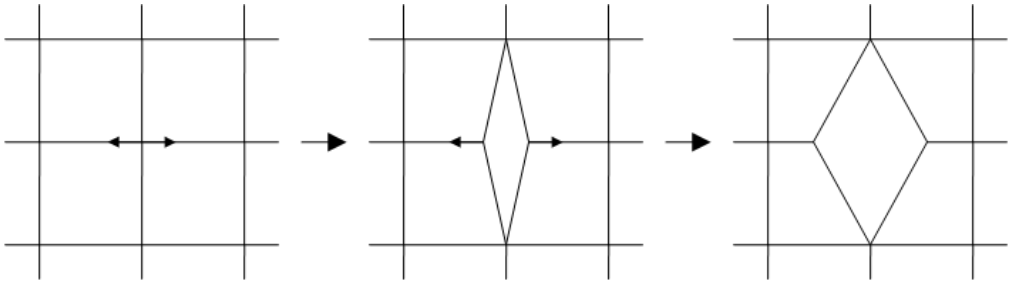

Figure C-4: Face open.

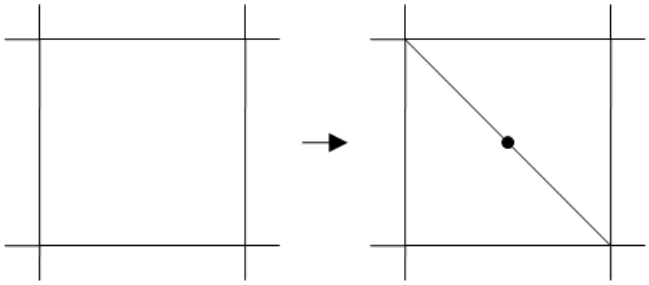

Figure C-5: Doublet insertion.

The edge swap operator shown in Figure C-2 alters the connectivity of the edge that is common to two adjacent quadrilaterals. The face close in Figure C-3 merges opposite nodes and two pairs of edges of a single face, removing the face completely. The face open in Figure C-4 reverses the process of a face close by separating a node and separating two non-adjacent edges attached to the node.

Figure C-5 shows a doublet insertion. The doublet insertion is actually a face open operation in which two adjacent edges are chosen resulting in the formation of a doublet; a situation in which two adjacent quadrilateral elements share two consecutive edges. Since the presence of a doublet actually signifies one or two degenerate elements, the doublet insertion is only an intermediate step in this clean-up process.

C.4 Diamond Collapse

Figure C-6 shows the definition and resolution to a set of quadrilateral elements with a unique topology called diamond quadrilaterals. The term diamond is used in reference to the shape typically observed in quadrilaterals with two 3-valence node opposite each other, illustrated in Figure C-6(a). Diamond quads do not usually cause unacceptably low quality; however, they are easy to identify their resolution results in a more structured mesh. Diamond quads are fixed with a simple face close operation in which the two 3-valence nodes opposite each other are merged as shown in Figure C-6(b). When the two nodes are combined, a 4-valence node is created as shown in Figure C-6(c). Additionally, the valence of each of the two other nodes in the diamond quadrilateral is reduced by one.

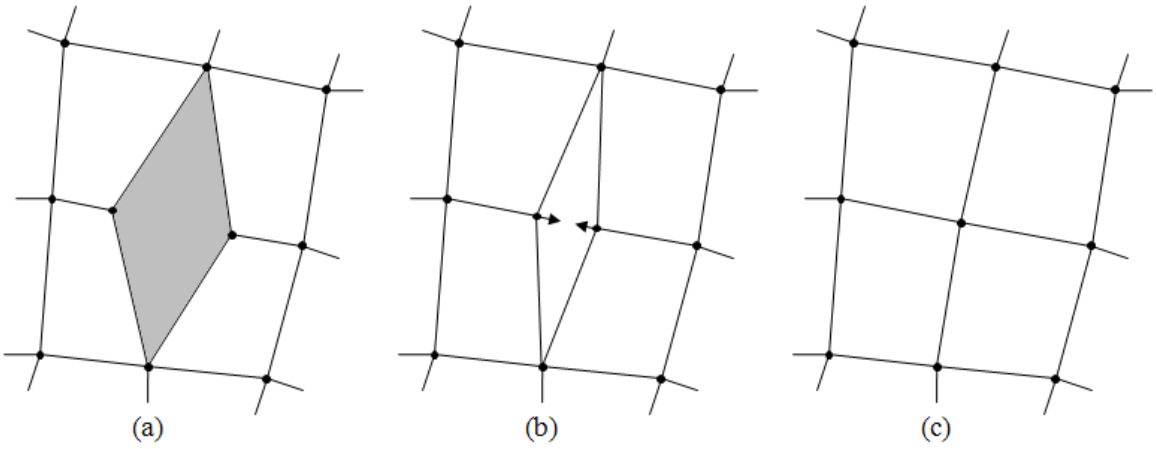

Figure C-6: Diamond quadrilateral collapse. (a) Shaded diamond quadrilateral. (b) Face close operation. (c) Resulting mesh.

C.5 Quadrilateral Edge Swap

The topology requiring a quadrilateral edge swap in this section is similar to the diamond topology already described in that it usually does not cause low quality elements; however, it is easy to identify and the result is always a more structured mesh. This case occurs when a quadrilateral element has a 5-valence node opposite of a 3-valence node and one of the edges on the 5-valence node is shared with another quadrilateral with a 5-valence node opposite of a 3-valence node, as illustrated in Figure C-7(a). By executing a quadrilateral edge swap, the 3- and 5-valence nodes used to define this case have all become 4-valence nodes. Note that in Figure C-7(a), five out of the six nodes in the region have valences other than four and after the operation only one of the six nodes has an unstructured valence, as shown in Figure C-7(b). To have high quality elements the mesh usually requires smoothing as well, as seen in Figure C-7(c).

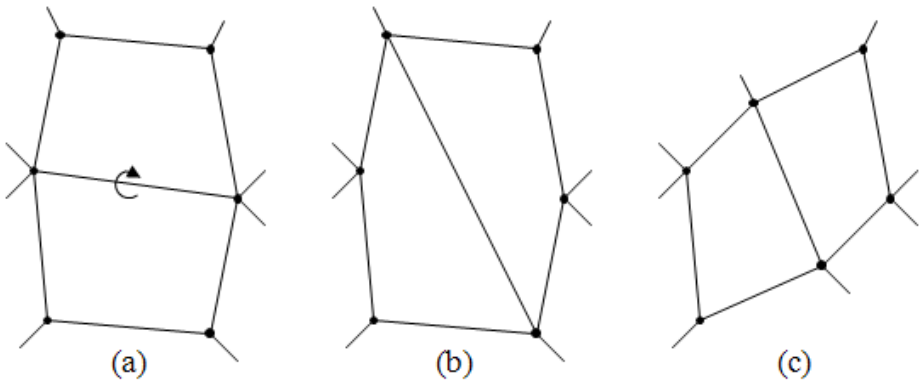

Figure C-7: Quadrilateral Edge Swap. (a) Edge to be swapped. (b) Resulting mesh before smoothing. (c) Resulting mesh after smoothing.

C.6 Doublet Removal

In quadrilateral meshes, a doublet occurs when two adjacent quadrilateral faces share two consecutive edges. Alternatively, a doublet exists at any interior node with a valence of 2. Because the two adjacent quadrilaterals share two edges, one or both of the elements will be degenerate. In coarsening applications, doublets occur regularly as elements are removed from the mesh. Doublets may also occur as a result of other topology modifications in this quadrilateral improvement algorithm. Doublets occurring on the interior of the mesh and can be resolved with a simple face close operation, illustrated in Figure C-8. Under these situations at least one of the quadrilaterals adjacent to the doublet node cannot have constrained edges.

This clean-up algorithm supports the preservation of annotated curves, defined by a set of marked edges that are not modifiable to maintain important geometric features (as shown by bold double lines in the figures). Sometimes, both of the quadrilaterals adjacent to the doublet node have one constrained edge, as shown in Figure C-9. Under

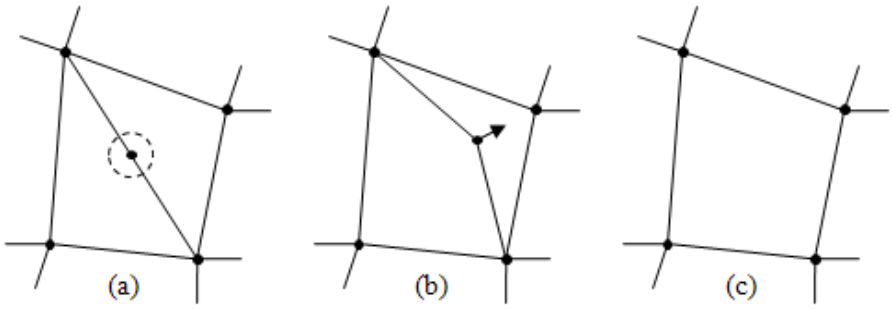

Figure C-8: Unconstrained doublet removal. (a) Doublet node shown in dashed circle. (b) Face close operation. (c) Resulting mesh.

such situations, a face-close operation will result in the formation of a very poor quality quadrilateral with two edges along the same geometric curve. Instead of using a face close, a second doublet is inserted within a neighboring quadrilateral, and an edge swap is executed to remove the two doublets.

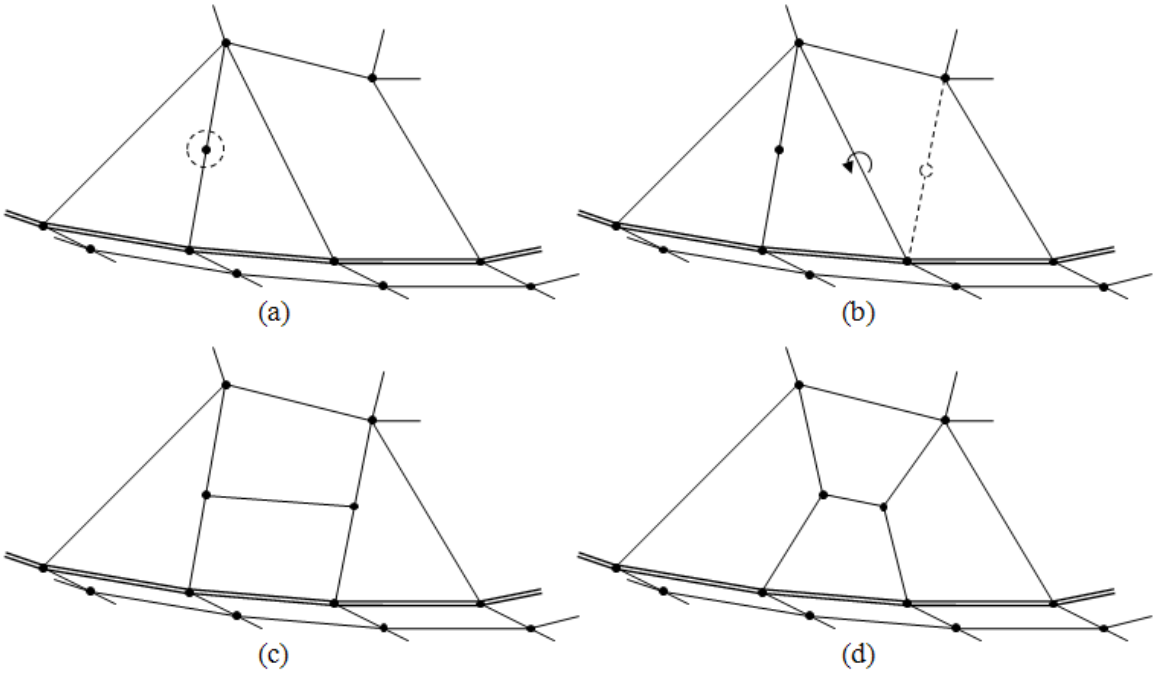

Figure C-9: Constrained doublet removal. (a) Doublet node, constrained by proximity to geometric curve, shown by dashed circle. (b) Doublet is inserted as shown by dashed line and then the marked edge is swapped. (c) Resulting mesh before smoothing. (d) Resulting mesh after smoothing.

Another special case configuration occurs along the feature edges, when both of the two quadrilaterals neighboring the doublet quadrilaterals have two constrained feature edges, illustrated in Figure C-10. Because of the close proximity of the doublet node to the geometric curve and the limitations caused by the constrained quadrilaterals on both sides of the doublet quadrilaterals, this case is considered highly constrained. The two previous methods are unable to resolve the doublet, requiring a third method,

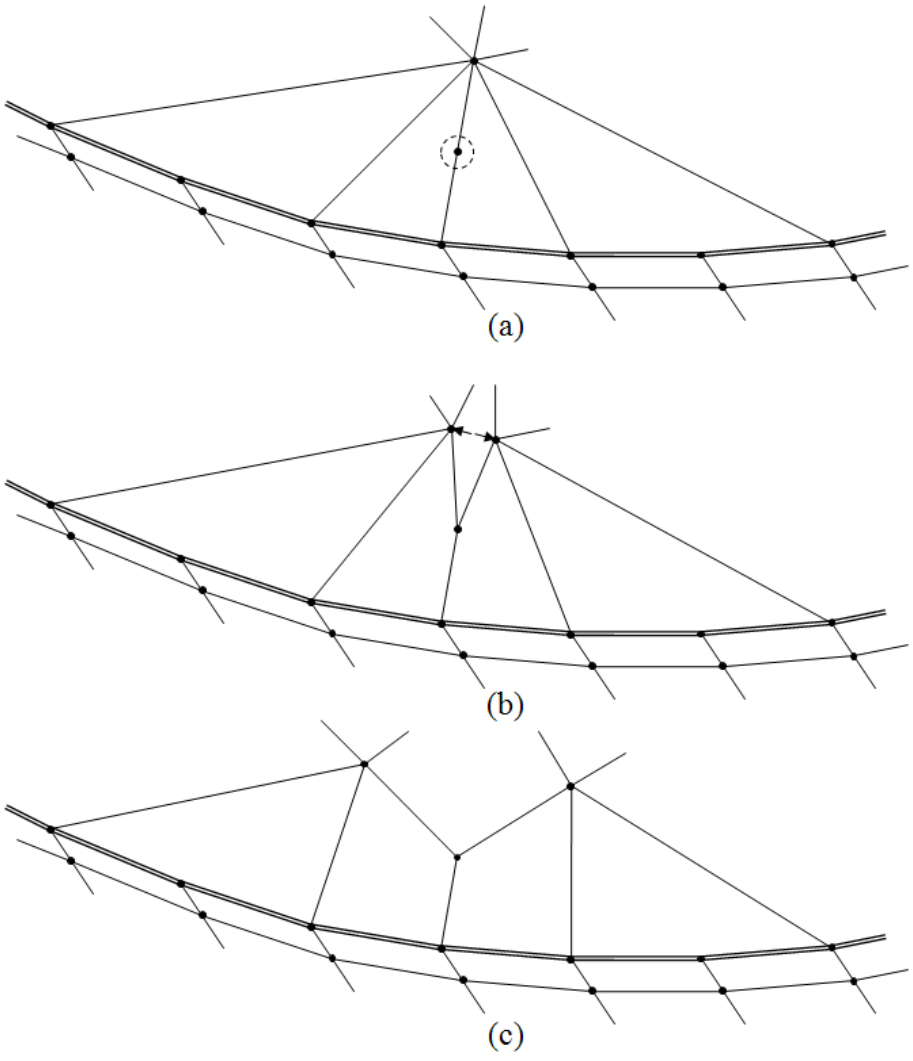

Figure C-10: Highly constrained doublet removal. (a) Doublet node shown by dashed circle. The doublet is constrained by the geometric curve and highly constrained by the triangular shaped quadrilaterals on either side. (b) Face open operation. (c) Resulting mesh.

implementing a face open. In this case, the node associated with both of the doublet quadrilaterals that is not the doublet node or the node on the geometric curve is used as the location for the face open. One of the doublet edges is also chosen to take part in the face open operation.

C.7 Constrained Quadrilateral Cleanup

Quadrilaterals that are constrained by geometric features, with 2 or 3 edges belonging to the same feature curve, are some of the most difficult to improve. When the geometric curve has little curvature with respect to the size of the quadrilateral element, the scaled Jacobian of the element approaches 0.0. Although constrained quadrilaterals usually have very poor shape quality, there are cases in which a constrained quadrilateral may have good quality, such as in a region of high curvature of the feature edges. For example, if a small circular feature curve is defined with a coarse mesh, the elements near the curve may have high quality and will not require any reconfigurations. To accommodate these cases, we ignore constrained quadrilaterals with reasonable shape quality.

There are two classes of constrained quadrilaterals, termed the triangle quadrilateral and the flattened quadrilateral. A triangle quadrilateral refers to the distinctive triangular shape typically observed in quadrilaterals that have two consecutive edges on the same geometric curve, illustrated in Figure C-11(a). Triangle quadrilaterals are resolved by inserting a doublet into the element as shown in Figure C-11(b). The resulting mesh, shown in Figure C-11(c), is a constrained doublet which can then be resolved by the method illustrated previously, in Figure C-9.

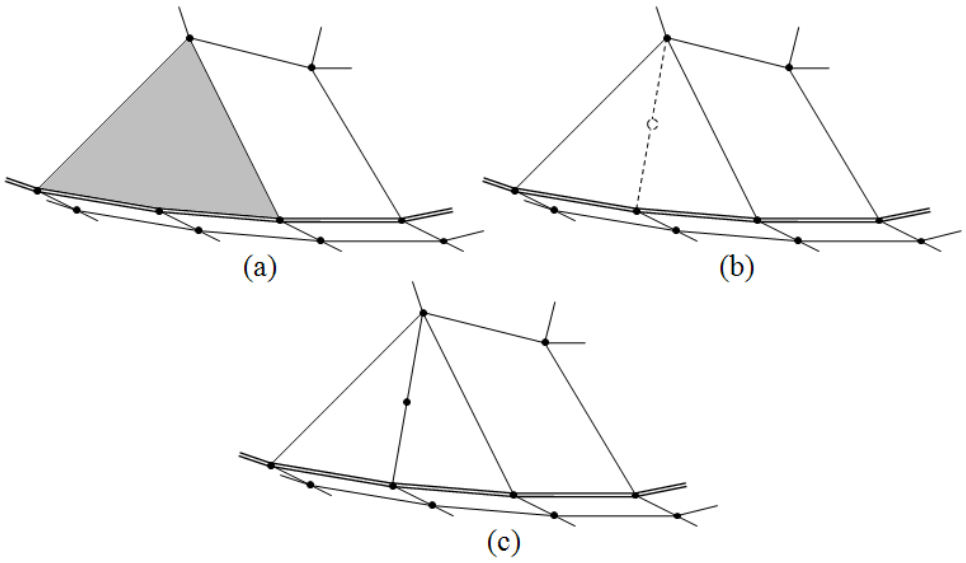

Figure C-11: Triangle quadrilaterals. (a) Shaded triangle quadrilateral. (b) Doublet insertion shown by dashed line. (c) Resulting mesh contains a constrained doublet.

The second constrained quadrilateral configuration, termed a flattened quadrilateral, occurs when the element has 3 edges on the same curve, shown in Figure C-12(a). An edge swap is used to improve the element quality as shown in Figure C-12(b). This generates a triangle quad, shown in Figure C-12(c) that can be improved using the method illustrated in Figure C-11.

C.8 High Valence Nodes

Nodes with a valence of 6 or more are considered high valence nodes. When the valence of a node reaches 6, the shape quality begins to deteriorate. Since the average angle between two adjacent edges at a high valence is less than or equal to 60°, these need to be removed to provide a higher quality.

There are two basic methods to reduce the valence of one of these nodes and can be performed depending on location of geometric curves, if any. The first method is

unconstrained and can easily be used on the interior of the mesh. This method employs a face open operation and is illustrated in Figure C-13. This method is generally preferred because, depending on which edges are chosen to participate in the operation, the valence can be reduced to one more than half of the original valence.

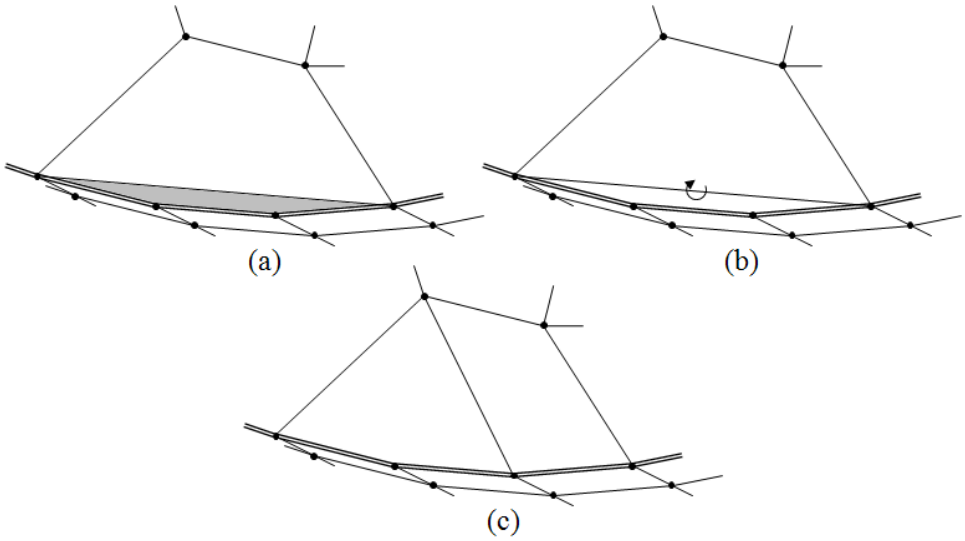

Figure C-12: Flattened quadrilaterals. (a) Shaded flattened quadrilateral. (b) Edge swap. (c) Resulting mesh contains a Triangle quadrilateral.

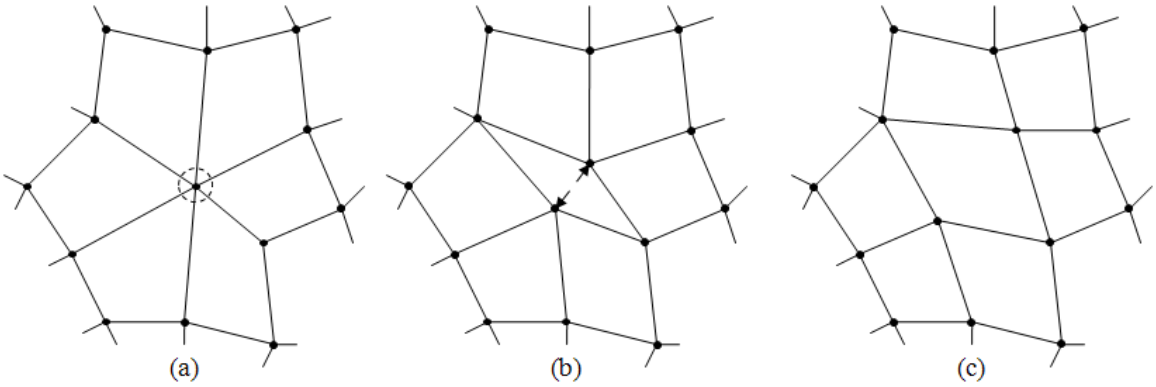

Figure C-13: Unconstrained high valence node removal. (a) 6-valence node shown with dashed circle. (b) Face open operation. (c) Resulting mesh.

If the high valence node is contained on a feature curve, a face open is impossible and a face close can be used as shown in Figure C-14. This case is less desirable because it only decreases the valence of the node in question by one and often creates a new high valence node. The creation of a new high valence node, however, is acceptable because it will not be on a feature curve and can be resolved by a face open operation in future iterations.

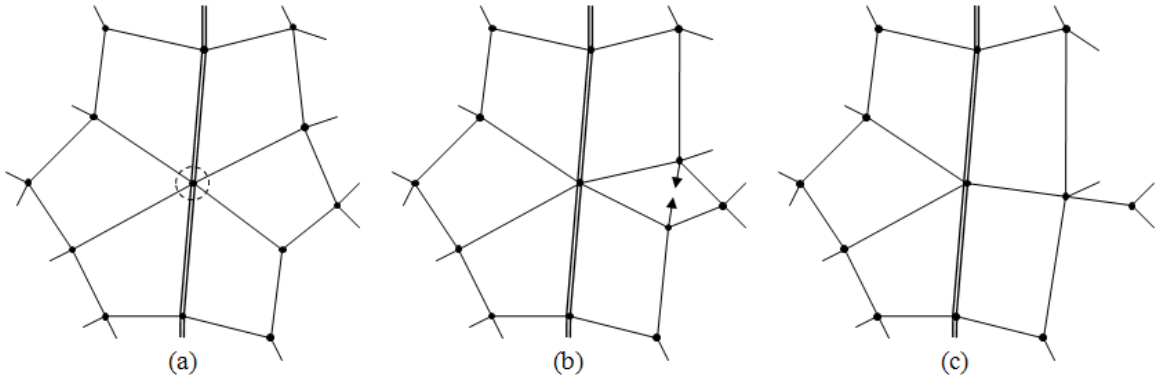

Figure C-14: Constrained high valence node removal. (a) 6-valence node constrained by curve shown with dashed circle. (b) Face close operation. (c) Resulting mesh.