



Theses and Dissertations

2008-03-20

Automated Quadrilateral Coarsening by Ring Collapse

Mark William Dewey
Brigham Young University - Provo

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



Part of the [Civil and Environmental Engineering Commons](#)

BYU ScholarsArchive Citation

Dewey, Mark William, "Automated Quadrilateral Coarsening by Ring Collapse" (2008). *Theses and Dissertations*. 1347.

<https://scholarsarchive.byu.edu/etd/1347>

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

AUTOMATED QUADRILATERAL COARSENING
BY RING COLLAPSE

by

Mark W. Dewey

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

April 2008

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Mark W. Dewey

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, Member

Date

Jason F. Shepherd, Member

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Mark W. Dewey 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 QUADRILATERAL COARSENING BY RING COLLAPSE

Mark W. Dewey

Department of Civil and Environmental Engineering

Master of Science

In most finite element analysis, a uniform mesh is not the optimum way to model the problem. Mesh adaptation is the ability to modify a finite element model to include regions of the mesh with higher and lower node density. Mesh adaptation has received extensive study in both computational mechanics and computer graphics to increase the resolution or accuracy of the solution in specific areas. The algorithm developed in this thesis, the Automated Quadrilateral Coarsening by Ring Collapse (AQCRC) algorithm, provides a unique solution to allow conformal coarsening of both structured and unstructured quadrilateral finite element meshes. The algorithm is based on dual chord operations and dual chord removal. The AQCRC algorithm follows six steps: 1) input of a coarsening region and factor, 2) selection of coarsening rings, 3) improvement of mesh quality, 4) removal of coarsening rings, 5) mesh clean-up and 6) coarsening iterations. Examples are presented that show the application of the algorithm.

ACKNOWLEDGMENTS

I wish to give thanks to the giants who have leant me their shoulders in implementing this algorithm: Steven Benzley, Matt Staten, and Mike Scott in particular. I would like to further thank those others who have helped me get involved in the meshing community: Mike Borden, Jason Shepherd, Karl Merkley, Randy Morris, Ray Meyers, and all the others at CSimSoft and Sandia National Labs.

I also feel I should acknowledge Bret Anderson who has helped me in coding clean-up operations, Adam Woodbury who has made the leap into three dimensions, and Mike Parrish who tirelessly helped me as a sounding board for my ideas.

Finally I would like to pay homage to my mother who taught me to write, my father who gave me a love for science, my brothers and sisters who have always been my best friends, and my wife Kamis who is the center of my universe and makes sure I eat three meals a day. I thank God for these friends and the many other blessings he has provided me.

TABLE OF CONTENTS

LIST OF TABLES	ix
LIST OF FIGURES	xi
1 Introduction.....	1
2 Background	5
2.1 Triangle Mesh Simplification and Coarsening	5
2.2 Quadrilateral Mesh Simplification and Coarsening.....	6
3 Automated Quadrilateral Coarsening by Ring Collapse	11
3.1 Defining the Coarsening Region and Removal Parameter	12
3.2 Selecting Coarsening Rings	15
3.2.1 Ring Identification	15
3.2.2 Ring Selection.....	20
3.3 Improving Quality.....	22
3.4 Collapsing Coarsening Rings.....	26
3.5 Mesh Clean-up.....	27
3.6 Coarsening Iterations	27
4 Results and Examples.....	29
4.1 Typical Applications.....	29
4.1.1 Membrane Mesh Example	30
4.1.2 Swept Mesh Example	32
4.1.3 Shell Mesh Example	35

5	Conclusions and Recommendations	39
5.1	Magnitude of Coarsening and Quality Considerations.....	39
5.2	Summary.....	40
5.3	Further Research.....	41
	References	43
	Appendix A. Basic Chord Operations	47
	Edge Swap Operation.....	47
	Face Close Operation.....	48
	The Doublet Insertion Operation.....	49
	Template Insertion Operation.....	49
	Face Open Operation.....	50
	Appendix B. Partial Chord Removal	53
	Single Face Collapse.....	53
	Stringing Face Closes Together.....	54
	Appendix C. Clean-up Operations	55
	Smoothing.....	55
	Diamond Quadrilateral Removal.....	56
	Doublet Removal.....	56
	High-Valent Node Reduction.....	57
	Triangle-shaped Quadrilateral Removal.....	59
	Appendix D. Implementation Concerns in CUBIT	63
	CUBIT Command Parameters.....	63
	Temporary Data Structure.....	63

LIST OF TABLES

Table 3.1: Rings available for selection	21
Table 3.2: Rings reordered with selected rings in italics	22
Table 4.1: Membrane mesh quality results comparison	32
Table 4.2: Membrane mesh area results comparison	32
Table 4.3: Swept mesh quality results comparison	34
Table 4.4: Swept mesh area results comparison.....	34
Table 4.5: Shell mesh quality comparison results	36
Table 4.6: Shell mesh area comparison results	37

LIST OF FIGURES

Figure 2.1: Quadtree and chord removal coarsening	8
Figure 2.2: Removing a chord from a mesh	8
Figure 2.3: Chord operations and removal	10
Figure 3.1: Algorithm flow chart	12
Figure 3.2: Coarsening region selection	14
Figure 3.3: A simple coarsening region with a ring selected	16
Figure 3.4: Double-wide and peninsula invalid ring cases	17
Figure 3.5: The three-valent node invalid ring case	18
Figure 3.6: Coarsening rings	19
Figure 3.7: Cases in which application of template insertion is advised	23
Figure 3.8: Star node group and resolution	25
Figure 3.9: Two options for dealing with stair-step node group	26
Figure 3.10: Collapsing node groups	27
Figure 3.11: Original mesh compared to coarsened region.....	28
Figure 4.1: Membrane mesh coarsening region	30
Figure 4.2: Membrane mesh at various levels of coarsening	31
Figure 4.3: Cylinder mesh before and after coarsening	33
Figure 4.4: Swept cylinder mesh	35
Figure 4.5: Shell mesh before and after coarsening	36
Figure A.1: The edge swap, face close and doublet insertion operations	48

Figure A.2: The template insertion operation	50
Figure A.3: The face open operation	51
Figure B.1: Face close operation	53
Figure B.2: Partial chord coarsening	54
Figure C.1: Diamond quadrilateral removal.....	56
Figure C.2: Simple doublet removal	57
Figure C.3: Resolution options for high-valent nodes	58
Figure C.4: Triangle-shaped quadrilaterals	60
Figure C.5: Basic resolution to triangle-shaped quadrilaterals	61
Figure C.6: Solution for triangle-shaped quadrilateral between two curves	61

1 Introduction

Finite element analysis (FEA) continues to push the limits of computing power. The computation time required even by the most powerful machines is hours or days on complex problems. Computation time increases proportional to the cube of the number of nodes in the finite element mesh. The accuracy of the finite element analysis is also proportional to the number of nodes in the mesh [1]. However, in many analysis situations there are areas of interest where accuracy of analysis is more important than in other locations or where greater resolution and detail is desired. For example, if the solution has a high gradient in a particular location, the errors produced by a low density mesh may transfer into other parts of the solution. A high node density in the areas of interest is crucial to accurate analysis; however, the need for this density may not transfer to other regions. If a uniform mesh is used, analysis time may be much greater than necessary while overall accuracy in the analysis may not improve significantly. The competing objectives of accuracy and computation time have prompted investigation into the field of mesh adaptation for the purpose of optimizing meshes.

Most research in mesh adaptation for computational mechanics centers on refinement algorithms which divide the mesh in the area of interest into smaller elements [2]. An algorithm which can coarsen an existing mesh by removing elements outside of the area of interest would be a powerful companion tool to refinement algorithms that

would allow more flexible adaptation of a mesh, or maintain computation speed when an analysis indicates that a more refined mesh is required to capture some detail. Unfortunately, coarsening is an area of research which has received limited attention in the field of computational mechanics. Coarsening methods have been explored in the field of computer graphics. The coarsening methods developed for computer graphics are generally referred to as simplification algorithms as their purpose is to simplify a model when a lower level of detail is required. The algorithm presented in this paper, Automated Quadrilateral Coarsening by Ring Collapse (AQCRC), provides a fully automated conformal coarsening algorithm suitable for use with generalized unstructured quadrilateral finite element meshes.

The ability to manipulate a mesh by both refining and coarsening significantly increases the ability to adapt a mesh. For example given an original, fairly uniform mesh, the mesh density in an area of interest may be increased by established refinement techniques [3] and decreased away from the areas of interest by a coarsening technique such as AQCRC. An initial analysis on a base mesh may be used to indicate locations where high density meshes and low density meshes are appropriate based on the gradients of the initial solution. Rather than remeshing the model, the initial mesh may be modified using refinement and coarsening tools. This would allow increased resolution and accuracy in the results while maintaining a similar computation time for the entire model. Furthermore, a given model may require adaptation in different locations depending on different load cases, adaptation by both refinement and coarsening from a single uniform base mesh may allow more efficient and robust generation of meshes appropriate for various load cases.

This paper describes the development and implementation of the AQCRC algorithm for use in quadrilateral finite element analysis. Chapter 2 presents the background of the problem and discusses the limitations of previously developed coarsening and simplification algorithms. Chapter 3 describes the implementation of the algorithm. Chapter 4 evaluates the merit of the algorithm and demonstrates a few primary applications. Conclusions are provided in Chapter 5.

2 Background

Mesh adaptation is a field which has received extensive study among both computational mechanics and computer graphics researchers. Generally these two fields have not colluded as many of the restrictions which apply in computational mechanics are not necessary in computer graphics. The adaptation algorithms developed for computer graphics are therefore rarely applicable to computational mechanics. For example, in finite element analysis, a mesh must accurately represent the geometry of the model by insuring that the nodes representing a curve or surface of the model do not move off the geometry, whereas in graphics a sufficiently low level of detail might justify combining surfaces and curves. Similarly, the degree to which the elements of an all-quadrilateral finite element mesh resemble a square has a large impact on the accuracy solution, but has no bearing on the representation of a computer graphics model. The following sections describe some efforts, similar to the AQCRC algorithm and the motivation for that work.

2.1 Triangle Mesh Simplification and Coarsening

In computer graphics and computational mechanics, many meshes are made entirely of triangles because triangle meshing methods are simple and have been well established. One of the foremost algorithms of mesh simplification for triangles was

developed by Garland, et al. [1]. His approach is fast, reliable, and generally applicable to any polygon mesh. The algorithm assumes that the mesh is composed entirely of triangles or can be broken into a mesh entirely composed of triangles. It is designed to combine surfaces and curves that are indistinguishable when rendered at a low level of detail. Another useful method of surface mesh simplification was developed by Hoppe, et al. [5]. This method respects geometry curves and surfaces in order to preserve sharp corners and edges. Numerous other triangular mesh coarsening algorithms are documented by Cignoni, et al. [6]. Additional reviews that compare smaller sets of algorithms are found in References 7 and 8. While triangle meshes have widespread use, quadrilateral meshes are often preferred in finite element analysis because they converge to the correct solution with fewer elements than triangle meshes [1]. Most of the algorithms developed for triangle meshes cannot be adapted for use on quadrilateral meshes.

2.2 Quadrilateral Mesh Simplification and Coarsening

Takeuchi, et al. [9], modified the approach developed by Garland, et al. [4], to simplify the quadrilateral meshes. The process is robust and well-generalized, but as noted above it does not respect the geometry of the model. It is also optimized for global simplification; however, localized simplification is preferred in most finite element analysis situations involving mesh adaptation. Furthermore, it may produce degenerate quadrilaterals which have a negative impact on finite element analysis accuracy.

One very flexible method of coarsening a mesh is to completely remesh the defined region. Kwak, et al., have made strides in the use of such algorithms [11]. For

this application, the geometry of the models they are working on change as the solver iterates, necessitating the need to change the mesh density in specific regions. The algorithm refines specific areas of the mesh, but carefully maintains coarseness away from the areas of interest. The drawback of this method is that remeshing is a global approach to a local problem which may require a great deal more computational time than simply to modify a portion of the existing mesh. It should be noted that this algorithm is primarily designed to be a refinement algorithm; however, it could just as easily be used as a coarsening algorithm.

Cheng, et al., developed a method of coarsening an all-quadrilateral structured mesh for use on autobody parts [10]. The algorithm has potential for general meshes, but as yet the generalized implementation of it has not been published.

Another coarsening method which has received attention in the computational fluid dynamics community is undoing refinement which has previously taken place. Choi describes such an algorithm which can be used to undo previous refinement on both quadrilateral and hexahedral meshes [12]. The reliance on previous refinement restricts the algorithm from being used on a base mesh that has not been refined.

Nikishkov developed a method for mesh adaptation that allows both refinement and coarsening [13]. However, his method produces unconformal elements as shown in Figure 2.1. The left panel shows the original mesh with the region to be coarsened highlighted in grey. The center panel shows the mesh once coarsening has been applied. The nodes marked A, B, C, and D are locations where the mesh is not conformal. Conformal quadrilateral meshes, where every internal quadrilateral is bounded by exactly four quadrilaterals, are required by most FEA solvers. Nikishkov addresses the

conformability issue by using special elements. These special elements restrict the use of this algorithm for general local coarsening by requiring the use of Nikishkov's elements.

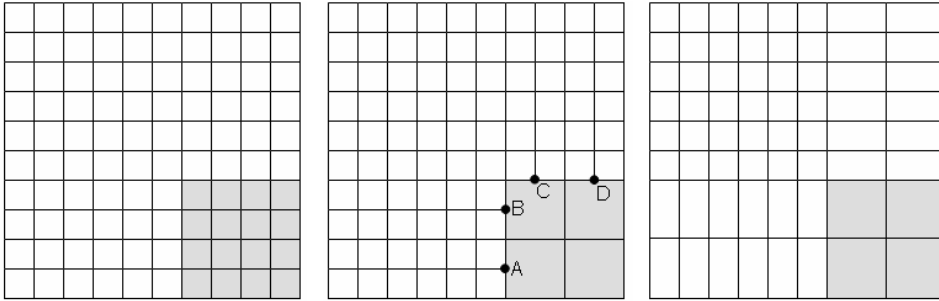


Figure 2.1: Quadtree and chord removal coarsening

Much of the basis of the AQCRC algorithm was developed by Murdoch, et al., when he defined the dual of the mesh [14]. The dual of a quadrilateral mesh is the set of all chords in the mesh. A chord is a set of quadrilaterals connected by opposite edges. Figure 2.2 shows a chord in the left panel as a dotted line running through the mesh.

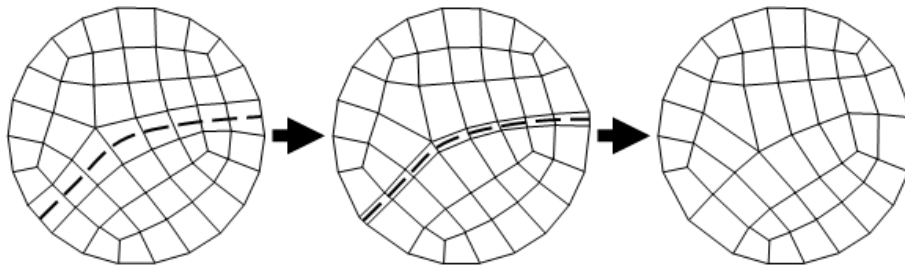


Figure 2.2: Removing a chord from a mesh

Borden, et al., recognized that it is possible to remove an entire chord from a quadrilateral mesh, maintaining conformal connectivity, by simply collapsing the defining edges of the chord as shown in Figure 2.2 [15]. The removal of a chord reduces

the number of quadrilaterals in the mesh and coarsens the quadrilaterals adjacent to the chord. The right panel of Figure 2.1 shows this kind of coarsening applied to the selected coarsening region. Unfortunately, while the local region is significantly coarsened, the effect of the coarsening extends well beyond the boundaries of the coarsening region. The research of Borden, et al., was continued in the paper by Benzley, et al. [16], where initial steps were made to localize the coarsening region.

Staten, et al., presented some of the concepts of the AQCRC algorithm by realizing that if a circular chord (i.e. a chord which forms a closed loop) could be created in a specific region, then the removal of that chord would produce localized coarsening [17]. Partial chords consisting of a continuous subset of the quadrilaterals of a chord are identified bounding the coarsening region. They show that if a set of bounding partial chords can be established, then simple chord operations, i.e. alterations to the mesh which change the connectivity of chords, can be performed at the intersections of these bounding partial chords to combine them into one continuous chord surrounding the region to be coarsened. The operations used to combine the partial chords are the edge swap, face close and doublet insertion operations documented in Appendix A. While not described as such by Staten, et al., a set of bounding partial chords can be considered a coarsening ring, as it is called for the remainder of this paper.

Figure 2.3 shows an example of the coarsening process developed by Staten, et al. [16]. The left panel shows the initial mesh. The dashed lines mark chords which bound the region to be coarsened. The quadrilaterals highlighted in grey are the bounding partial chords and define a coarsening ring. In the second panel, the quadrilaterals where two bounding partial chords intersect have been modified with

doublet insertion, face close and edges swap operations. The top left intersection is modified using a doublet insertion operation. In the center panel, the doublet node is circled for clarity. The bottom two intersections are modified by a face close operation. The top right intersection is modified by an edge swap operation. The merits of each operation are discussed further in Appendix A. Once the operations have been performed a single circular chord now bounds the coarsening region. This chord is removed by the same means as that shown in Figure 2.2; the right panel shows the mesh after this bounding chord has been removed.

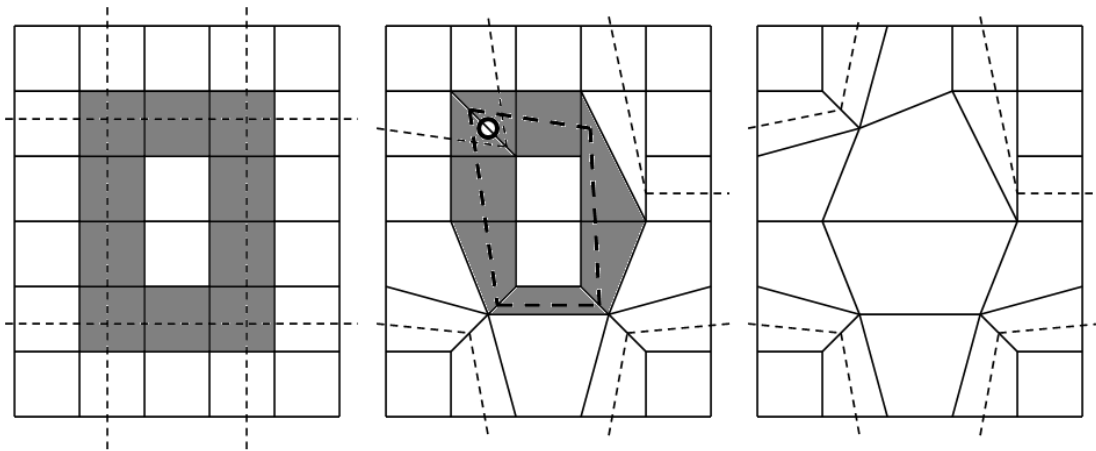


Figure 2.3: Chord operations and removal

3 Automated Quadrilateral Coarsening by Ring Collapse

The algorithm developed by Staten, et al., is a powerful quadrilateral coarsening algorithm [17]. The AQCRC algorithm, presented in this chapter, generalizes, develops and automates the work of Staten, et al. The AQCRC algorithm further advances this methodology by developing input parameters, creating and selecting from multiple coarsening rings simultaneously, introducing logic to maximize element quality, and eliminating the step of combining the bounding partial chords into a single circular chord.

One of the key developments of the AQCRC algorithm is the use of coarsening rings, i.e. a closed loop of quadrilaterals contained within the coarsening region, rather than circular chords in the element removal process. This is a generalization of the algorithm as any circular chord within the coarsening region is also a coarsening ring. The AQCRC algorithm does not take the step of modifying the bounding partial chords with chord operations to create a single chord as shown in Figure 2.3. The quadrilaterals highlighted in grey in the left panel are instead considered a coarsening ring and will be removed directly in a process similar to chord removal.

The AQCRC algorithm involves six key steps outlined below and shown in the flow chart in Figure 3.1. These steps will be examined in detail in the following sections.

1. A contiguous coarsening region and the final mesh coarseness are specified.

2. One or more coarsening rings are selected within the coarsening region containing a number of quads less than or equal to a goal number of quads.
3. The bounding partial chord intersections may be altered with chord operations to increase the final quality of the mesh or to prevent merging nodes illegally.
4. The identified coarsening rings are collapsed from the mesh and the mesh reconnected in a manner that retains its conformal properties.
5. The mesh is checked, cleaned up, and smoothed to ensure that elements have acceptable quality.
6. Steps two through five are repeated if insufficient coarsening has taken place in a given iteration of the algorithm.

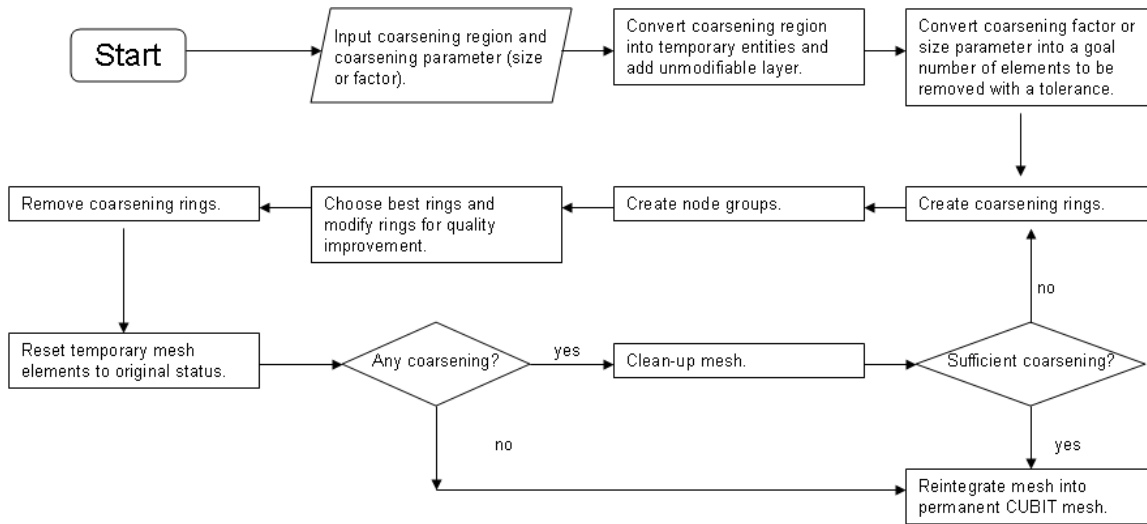


Figure 3.1: Algorithm flow chart

3.1 Defining the Coarsening Region and Removal Parameter

Figure 3.2 will be used to illustrate the input of a coarsening region. First, a base set of node, vertex, edge, curve, face, or surface entities is identified. In Figure 3.2 the

edges that have been marked with extra dark lines are the base set of entities used. A depth parameter is further specified to determine the number of layers of quadrilateral elements around this base set of entities. For the example, a depth of five was specified; the faces highlighted in dark grey are all within five faces of the base set of edges. This set of elements is the coarsening region. For convenience in later parts of the algorithm, an additional layer of elements, marked in light grey, is specified around the coarsening region. This layer of elements is marked as unmodifiable boundary elements and will form a transition layer between the coarsening region and the unchanged portions of the mesh. These elements may be partially altered but are defined not to be removed from the mesh. Occasionally there is not an additional layer of elements around the coarsening region because the border of the mesh at large is included within the coarsening region. In this case the elements on the boundary are marked as unmodifiable boundary elements in order to preserve the interval counts on the curves at the boundary of the mesh.

In addition to the selection of the coarsening region a target size or coarsening factor is specified to establish the goal number of elements which are to be removed from the mesh. A coarsening factor corresponds to the multiplicative increase in average area that should occur within the coarsening region during the execution of the AQCRC algorithm. A target size corresponds to the average length of the edges in the coarsening region after the execution of the AQCRC algorithm.

Equation 3.1 shows how a coarsening factor is used to determine the the number of quadrilaterals that should be removed.

$$N_{e-r} = E_t - \frac{E_t}{F} \quad (3.1)$$

where N_{e-r} = the number of elements to be removed

E_t = the number of elements in the coarsening region

F = the user supplied factor

Equation 3.2 converts the target size into an equivalent coarsening factor which is then converted into the number of elements to be removed by Equation 3.

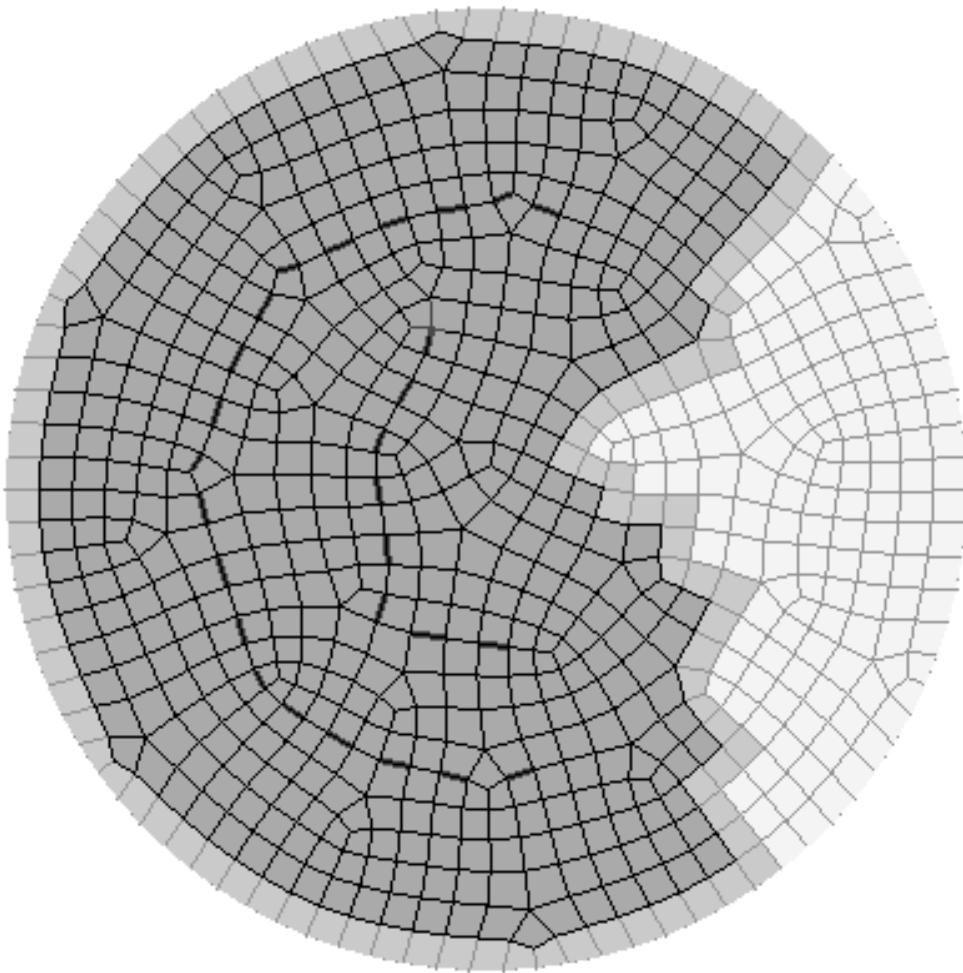


Figure 3.2: Coarsening region selection

$$F = \frac{l_f^2}{l_0^2} \quad (3.2)$$

where F = the equivalent factor

l_f = the final average edge length specified by the user

l_0 = the initial average edge length in the coarsening region

Once the number of quadrilaterals to remove is determined, a tolerance factor (t_f) of 10% of the original number of quadrilaterals to be removed is calculated such that the number of quadrilaterals actually removed is within $\pm t_f$. This tolerance factor is limited to being at least 3 and at most 50. The values of 10%, 3 and 50 are heuristics; a further examination of the determination of an appropriate coarsening factor is merited.

3.2 Selecting Coarsening Rings

Once the coarsening region has been established, the algorithm analyzes the mesh and creates a set of concentric coarsening rings for removal. The algorithm then selects from this set of rings the optimum rings to collapse based on which ring removals will preserve a high quality mesh and the goal number of elements to be removed.

3.2.1 Ring Identification

Each ring is created by locating an element in the coarsening region which is adjacent to or shares a node with an element marked as a boundary element. Initially, only the elements forming the unmodifiable layer are marked as boundary elements. The algorithm then proceeds to find additional, adjacent elements which are also along the boundary and add them until the ring closes. This complete loop around the outer

boundary of the coarsening region is stored as a new coarsening ring. If the algorithm successfully creates a valid ring, the elements are marked as boundary elements and the process is repeated. New rings are formed iteratively until there are no remaining elements in the coarsening region. Figure 3.3 shows a simple coarsening region, where the grayed out elements have been marked as boundary elements. The elements marked with a dashed line would be selected as a coarsening ring.

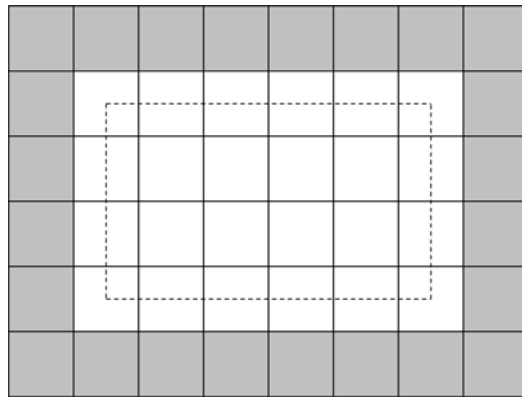


Figure 3.3: A simple coarsening region with a ring selected

During the process of identifying coarsening rings there are a few cases where the ring identified could not be collapsed or there are not enough elements to form a closed loop. These invalid cases are handled by marking some of the elements as boundary elements and then repeating the ring identification algorithm. Eventually all of the elements in the coarsening region will be marked as boundary elements which effectively ends the ring identification portion of the algorithm. Apart from not being able to complete a closed coarsening ring, the following three major invalid cases exist.

First, the loop may contain a region where it runs adjacent to itself; these regions are referred to as “double-wide” regions. The left panel of Figure 3.4 shows this case.

As before, the dashed line indicates the elements which have been selected as part of the ring being identified. The elements circled show double-wide region. Collapse of this ring results in a set of high-valent nodes. The valence of a node is the number of edges connected to the node. A node is considered high-valent if it has more than five edges connected to it. To resolve this problem, the two elements circled are marked as boundary elements, allowing the ring identification algorithm to be executed again, in this case, identifying the elements on the top and bottom of the figure as separate rings. It is noted from Figure 3.3, above, that once the elements in the ring have been marked as boundary elements, the only elements left will be double wide elements. These elements will not be used as a ring in the rest of the algorithm.

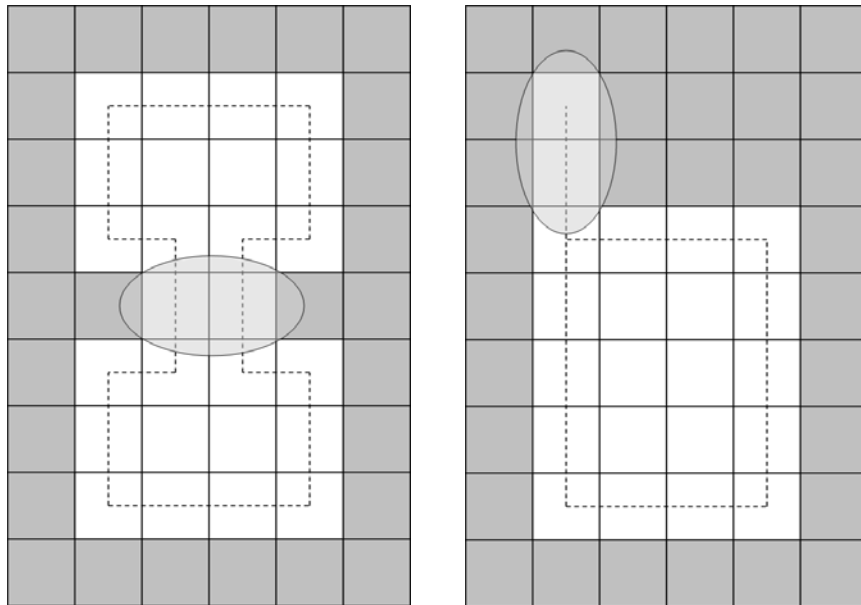


Figure 3.4: Double-wide and peninsula invalid ring cases

The second invalid ring case is when the ring intersects with itself in a location other than the starting location. If the ring has a three way intersection, it may not be

collapsed; the elements near the intersection cannot be conformally reconnected. In the unlikely event that a four way intersection is found, it can be collapsed, although this case is very rare. The right panel of Figure 3.4 shows a ring with a three way intersection. The two quadrilaterals circled on the right are considered peninsula quadrilaterals. These peninsula quadrilaterals are marked as boundary quadrilaterals and the ring identifying algorithm is once again employed. This time, the rest of the quadrilaterals previously identified as part of the ring will become the new ring.

The final case occurs in unstructured meshes where a three-valent node is adjacent to three quads that are all part of the ring as shown in Figure 3.5. The circled quadrilateral is unnecessary to closing the ring and is removed from the ring and marked as a boundary element. Furthermore, the inclusion of all three of these faces in the ring obscures future steps and leads to the creation of high-valent nodes.

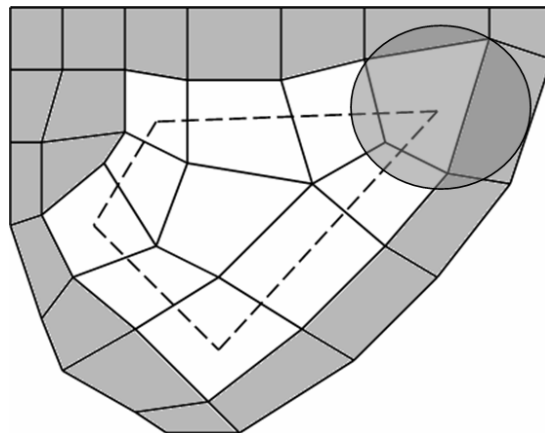


Figure 3.5: The three-valent node invalid ring case

In most cases the coarsening region is large enough that several rings are created concentrically. Figure 3.6 continues the example started with and shows the rings

developed within the coarsening region. The alternating numbered regions of darker and lighter shaded grey elements show the set of rings. The dark regions not numbered are locations of elements which are not included as rings because they were part of an invalid ring case. The initial set of edges selected for the coarsening region and the unmodifiable layer are still marked as in for ease of reference.

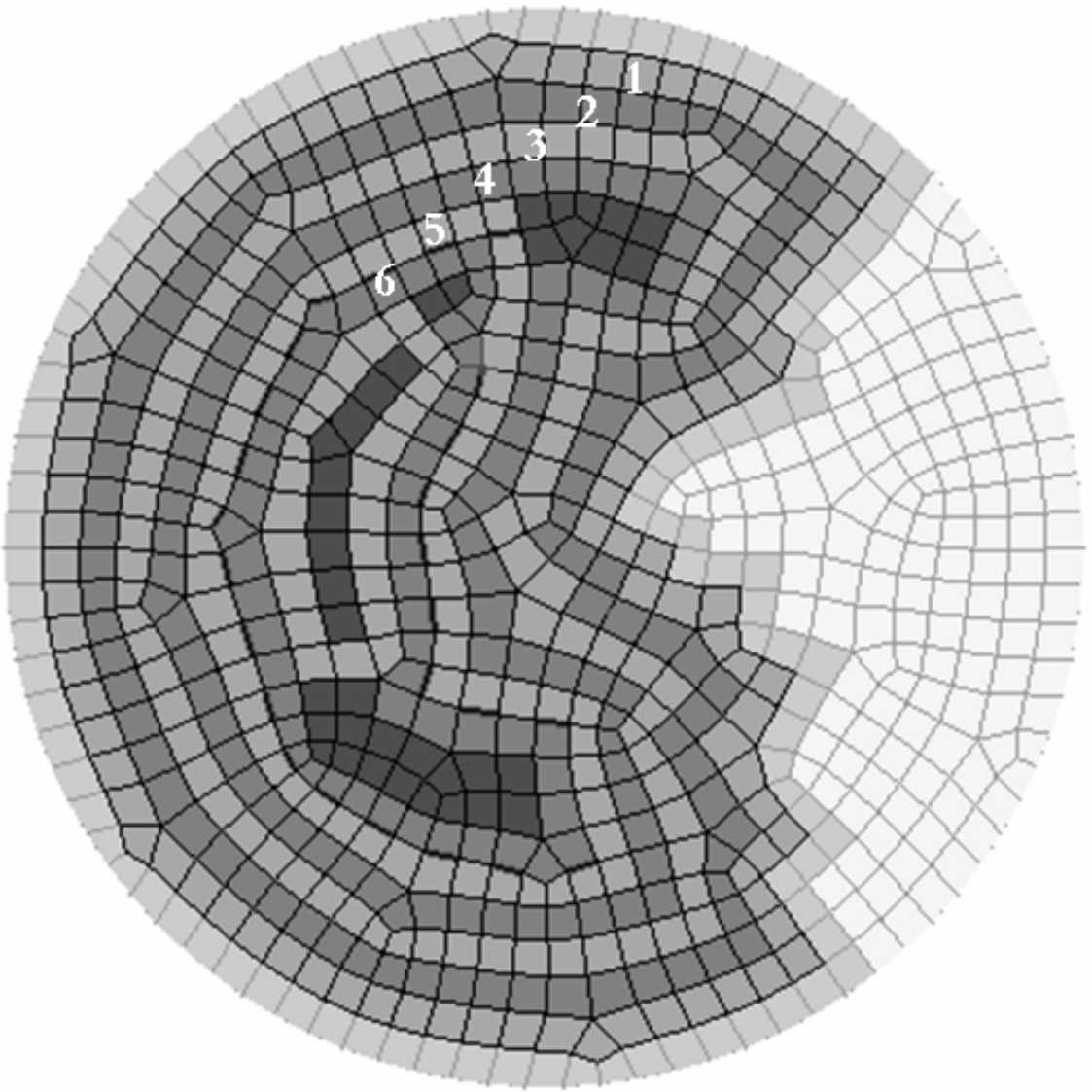


Figure 3.6: Coarsening rings

3.2.2 Ring Selection

Once the set of rings has been created, a subset of the rings is chosen for removal. To facilitate choosing an optimum set of rings for removal, each node connected to a ring is assigned to a node group. A node group is a set of nodes that will be merged into a single node when the coarsening ring is collapsed. A given node may only be associated with one node group per coarsening ring. Node groups are created by identifying edges that are between adjacent quadrilaterals in a given ring. The nodes on either end of these edges are then assigned to a node group. If one of the nodes is already assigned to a node group for the current ring, the other node is assigned to the same node group. If both have already been assigned to different node groups, the two node groups are combined into one node group. The node groups are established at this point in the algorithm so that a rough projection of the mesh connectivity and element shapes can be established before choosing which rings to collapse.

To facilitate choosing the best set of rings to collapse, each node group will be assigned a projected location. Generally, the projected location is the centroid of all the nodes in the group. However, this location is modified so that the projected location is on the most restrictive geometry entity any of the nodes is owned by, i.e. if one of the nodes is owned by a vertex the projected location is the location of the vertex, otherwise if one or more are on a curve the projected location is on the curve and if they are all on a surface the projected location is on the surface.

After the projected location of all the node groups is calculated, each ring is given a quality metric. This metric used here is equal to the minimum projected Scaled Jacobian [18] quality of the neighboring quadrilaterals once the ring has been collapsed.

The neighboring quadrilaterals use the projected location of the node groups rather than the current location of the nodes when making a quality assessment. Table 3.1 shows the results of this analysis for the example shown in Figure 3.6. The ring ID is a unique identifier assigned to each of the rings and corresponds to the numbering shown in Figure 3.6. The ring quality is the minimum projected Scaled Jacobian for all quadrilaterals adjacent to the ring assuming the ring is removed. The quadrilateral count is the number of quadrilaterals in the given ring.

Table 3.1: Rings available for selection

Ring ID	Ring Quality	Element Count
1	0.0425	90
2	0.2184	83
3	0.0297	78
4	0.1676	71
5	0.3107	58
6	0.0345	51

Once each ring has an associated quality metric, the rings are ordered according to quality using a quick sort algorithm. Any rings that have a quality metric less than 0 – suggesting that collapsing the ring would create a degenerate quadrilateral – are removed from the list. Table 3.2 shows the rings from Table 3.1 reordered according to their projected quality. Each ring is then considered for collapsing in order. Every time the collapsing of a ring is considered acceptable (see below), it is placed in a list and the number of quadrilaterals to be collapsed is updated. In Table 3.2, the rings to be collapsed have been marked in bolded italics.

A ring is considered acceptable if the current number of quadrilaterals to be collapsed plus the number of quadrilaterals in the current ring is less than the goal number of quadrilaterals to be collapsed plus the tolerance factor determined at the beginning of the program. Any ring that meets this qualification may be rejected if it is immediately adjacent to another ring that has previously been selected. The goal number of quads to be removed for the example given in Tables 1 and 2 is 369 quadrilaterals with a tolerance of 36 quadrilaterals. In this case the goal number of quadrilaterals is not reached; the total number of quadrilaterals collapsed in this iteration is only 141. After rings 2 and 10 are chosen, the others are not chosen because they are adjacent to these two rings. Additional iterations of the entire coarsening algorithm will be attempted until the goal number of quadrilaterals has been removed.

Table 3.2: Rings reordered with selected rings in italics

Ring ID	Ring Quality	Element Count
<i>5</i>	<i>0.3107</i>	<i>58</i>
<i>2</i>	<i>0.2184</i>	<i>83</i>
4	0.1676	71
1	0.0425	90
6	0.0345	51
3	0.0297	78

3.3 Improving Quality

In some cases, modifying the connectivity of a ring can improve the quality of the final mesh. As each ring is chosen for collapse, it is examined more closely to see if there is room for quality improvements. Quality is a major consideration in coarsening because nearly every time a ring is removed the quality of the mesh goes down. The

following procedures are intended to keep the algorithm from creating low quality elements as much as possible. Any node group containing three or more nodes is examined for possible improvements as discussed below.

A node group with three or more nodes indicates the existence of an intersection of bounding partial chords in the ring topology. If a three node group is near geometric curve boundaries, it may be advantageous to use a template insertion operation. Figure 3.7 shows this template insertion operation. The two top panels are the original meshes, while the two bottom panels show the meshes after a template has been inserted into a quadrilateral just inside the intersection of the two bounding partial chords. The double lines are geometric curves; the dashed lines indicate the quadrilaterals that are part of the ring being improved. The nodes circled are part of a node group; each node group has extra bold lines connecting the members of the group. Node groups where the template is applied are broken into three new node groups.

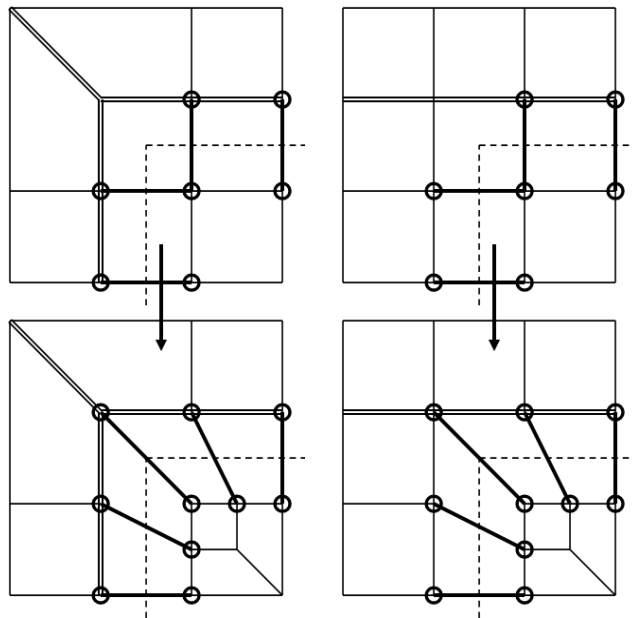


Figure 3.7: Cases in which application of template insertion is advised

The importance of using template insertion in each case is different. In the case shown on the left of Figure 3.7, the template insertion prevents merging of nodes on different curves, a situation that clearly needs to be avoided in order to respect the geometry of the model. In the case shown on the right, the use of the template simply prevents the creation of a triangular-shaped quadrilateral: if the nodes were merged, the quadrilateral to the left of the corner of the ring would have two edges on the same curve, a situation which would force the quadrilateral to be shaped like a triangle.

In node groups with four or more nodes, consideration is given to how the node group should be collapsed. A simple collapse may create high-valent nodes, reducing the element quality in the region. There are two problem cases which involve more than three nodes in a single node group. The first is referred to as the star case and the second is referred to as a stair-step case. When both cases are present in the same node group, the node group will be treated as the star case.

In unstructured meshes, multiple bounding partial chord intersections may share a node with a valence higher than four. This may result in all nodes in the node group linking back to a single node as shown in Figure 3.8. Star node groups are resolved by a simple collapse even though it may result in high-valent nodes. For improved quality, each sub-case could be resolved differently; however some of these cases require adding several new quadrilaterals (ie. through multiple template insertions), contrary to the goal of the coarsening procedure. For this condition it is preferable to collapse the node groups and address the high-valent nodes during the clean-up process.

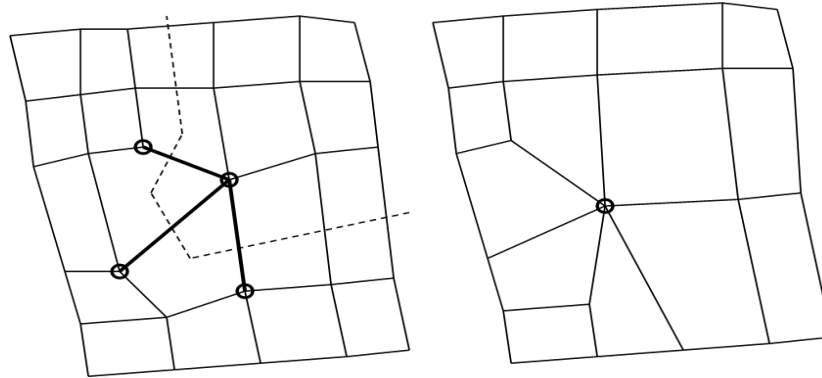


Figure 3.8: Star node group and resolution

The stair-step case is the most commonly encountered problem among node groups with four or more nodes. Figure 3.9 below shows the problem and resolution in detail. In a stair-step case the bounding partial chord intersections are adjacent to each other. The top left panel shows three adjacent intersections. The nodes circled are the members of the original group. The top right panel shows the mesh if the node group were simply collapsed: the resulting node valence is seven. There is no limit to the valence of the node group that may be created by this problem; while the clean-up operations described in Appendix C would address the example problem, it is better to resolve it as shown in the bottom two panels of Figure 3.9 and it resolves all stair step cases regardless of how many intersections are adjacent to each other. First, a face is created in the mesh by splitting one node and two of the edges connected to the node as shown in the bottom left panel. The original node group is broken up into three node groups which can be collapsed normally. This operation may be applied recursively for large stair step cases. The bottom right panel of Figure 3.9 shows the mesh once the ring has been collapsed. The quality and node valence of the surrounding elements are significantly better using this operation.

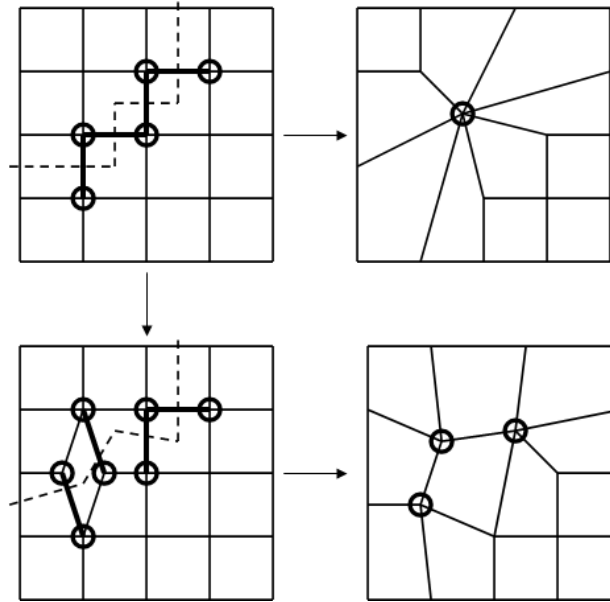


Figure 3.9: Two options for dealing with stair-step node group

3.4 Collapsing Coarsening Rings

Once the coarsening rings have been selected and the various quality improvement templates have been applied, the mesh is ready to be coarsened. Each of the selected coarsening rings is collapsed in succession using the following procedure. The nodes in each node group are moved to the projected position and the quadrilaterals that are part of the ring are deleted. As the quadrilaterals are deleted, any edge that is no longer associated with a quadrilateral is deleted and the nodes on either end of the deleted edge are merged together. At corners, where Staten, et al., recommend the application of chord operations, a simple collapse as shown in Figure 3.10 creates a conformal mesh identical to what would be produced by some of the operations they recommended (compare to the operation shown in Figure 2.3). In the left and center panels of Figure 3.10, the node groups are circled and extra-dark edges connecting each group. The

dashed line indicates the ring of quadrilaterals being collapsed. In the right panel the circled nodes are the locations of the merged nodes in the final mesh.

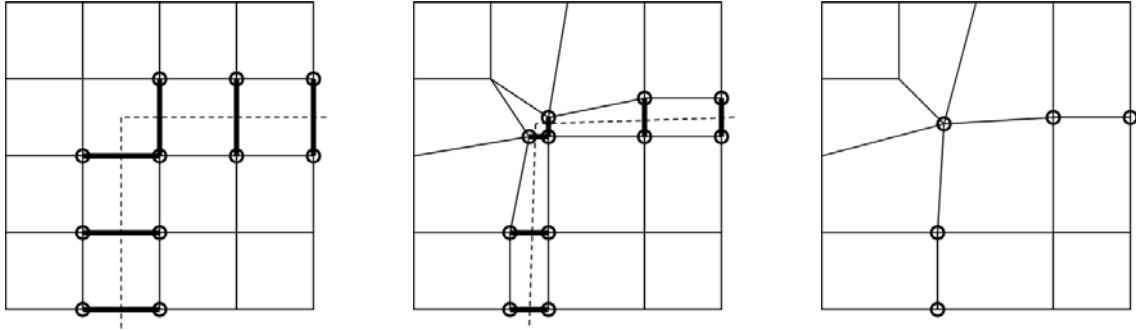


Figure 3.10: Collapsing node groups

3.5 Mesh Clean-up

Despite efforts to minimize high-valent nodes in previous steps, a few cases remain where high-valent nodes are formed. Furthermore, the collapse of two rings separated by only one layer of elements may have unexpected results that reduce the element quality of the mesh. Finally, collapsing quadrilaterals along geometric curves may result in elements with low quality which cannot be smoothed. To improve mesh quality, smoothing the mesh is always appropriate prior to further iterations of coarsening or to improve the quality of the final mesh. Therefore, after each coarsening iteration, clean-up and smoothing operations are performed. Most of these operations are defined in the paper published by Kinney [19] and are outlined in Appendix C.

3.6 Coarsening Iterations

At the end of each coarsening cycle the net number of quadrilaterals removed is determined, including any quadrilaterals added or removed by the clean-up procedures.

If too few quadrilaterals have been removed, the algorithm executes again. If a sufficient number of quadrilaterals have been removed or if no quadrilaterals have been removed, the program exits, integrating the coarsened region back into the original mesh. The algorithm provides a message if insufficient coarsening has taken place to reach the goal number of elements to be removed.

Figure 3.11 shows the example given in after it has gone through several coarsening iterations. The region that is completely outside of the coarsening region is identical in both meshes. However, fewer elements remain within the coarsening region.

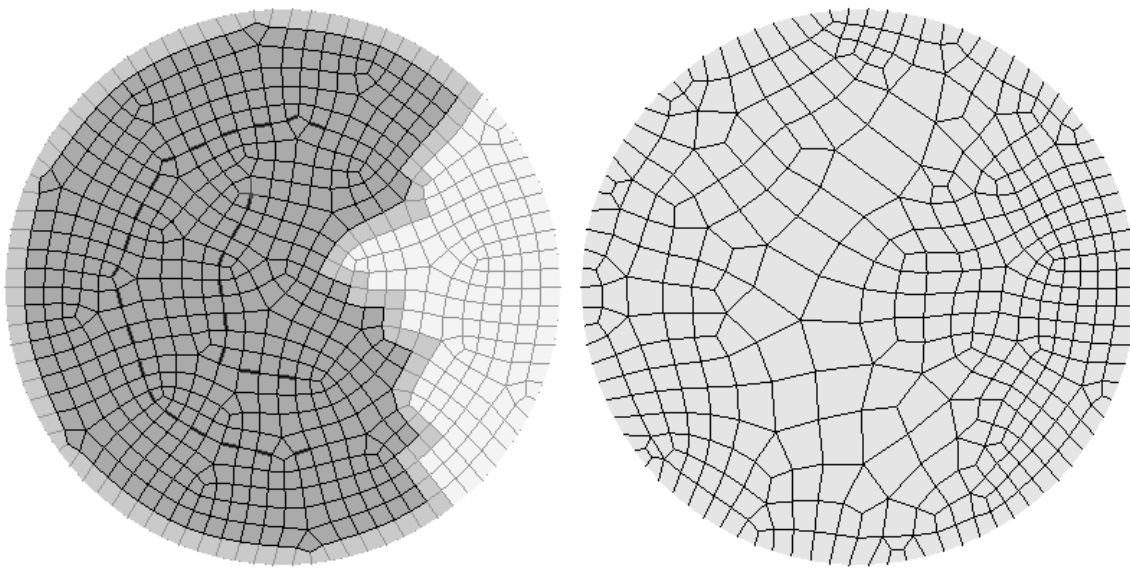


Figure 3.11: Original mesh compared to coarsened region

4 Results and Examples

Three example meshes are shown in this section that have been coarsened using the AQCRC algorithm. They are meant to represent three major applications of the coarsening algorithm. A discussion of the trade-offs between the magnitude of coarsening and the resultant quality of the mesh follows each of these examples.

The resulting area and quality of the meshes coarsened to various levels will be compared. The quality metric used is the Scaled Jacobian metric which ranges from -1.0 to 1.0 [18]. A Scaled Jacobian value of 1.0 represents a perfect square while anything below 0.0 is a degenerate element (0.0 typically being a triangle-shaped element). The element is generally considered acceptable for analysis accuracy if it has a quality above 0.2. Elements with a Scaled Jacobian value below 0.2 are considered marginal [20].

4.1 Typical Applications

The three examples provided to show the breadth of uses for the ring coarsening algorithms. In all three cases coarsening factors of 1.5, 4 and 10 were used to demonstrate the flexibility of the algorithm. The first example is mesh of a membrane with holes. The second example is the source surface of a swept mesh. The third example is the mesh of a quasi-spherical shell surface.

4.1.1 Membrane Mesh Example

Figure 4.1 shows a membrane mesh for a simple rectangular surface with bolt holes. The coarsening region is highlighted in darker grey lines. The coarsening region was chosen by selecting the five quads filled in with grey as a base set and using a depth of eight. The areas of interest in this problem surround the bolt holes where stress concentrations will arise. The coarsening procedure, appropriately applied, respects these high stress concentration zones by preserving the high quality, high density mesh in those regions.

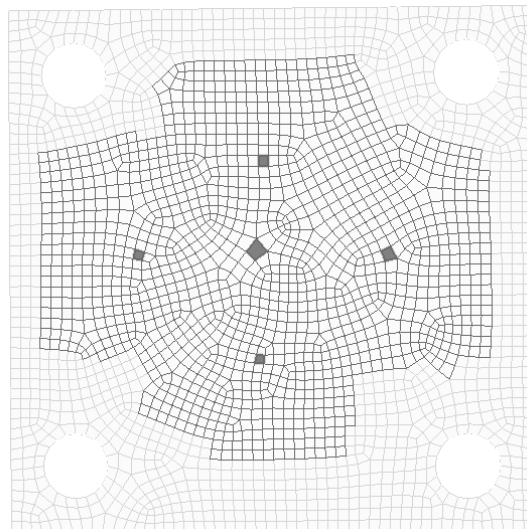


Figure 4.1: Membrane mesh coarsening region

The initial mesh and three coarsened versions of the mesh are shown in Figure 4.2. Each coarsened version of the mesh was created by coarsening from the initial mesh with a different coarsening factor. The top left mesh is the initial mesh, included for reference. The top right is coarsened to a factor of 1.5, the bottom left is coarsened to a factor of 4 and the bottom right is coarsened to a factor of 10.

Table 4.1 specifies the quality of the surface mesh for each level of coarsening. For reference purposes, the first row in the table provides the quality for the initial mesh. As shown in the table, the quality tends to degrade as coarsening increases except for when the coarsening factor 10 is used. However, the remaining examples in this section consistently demonstrate that the average and minimum quality of the mesh is reduced for larger amounts of coarsening.

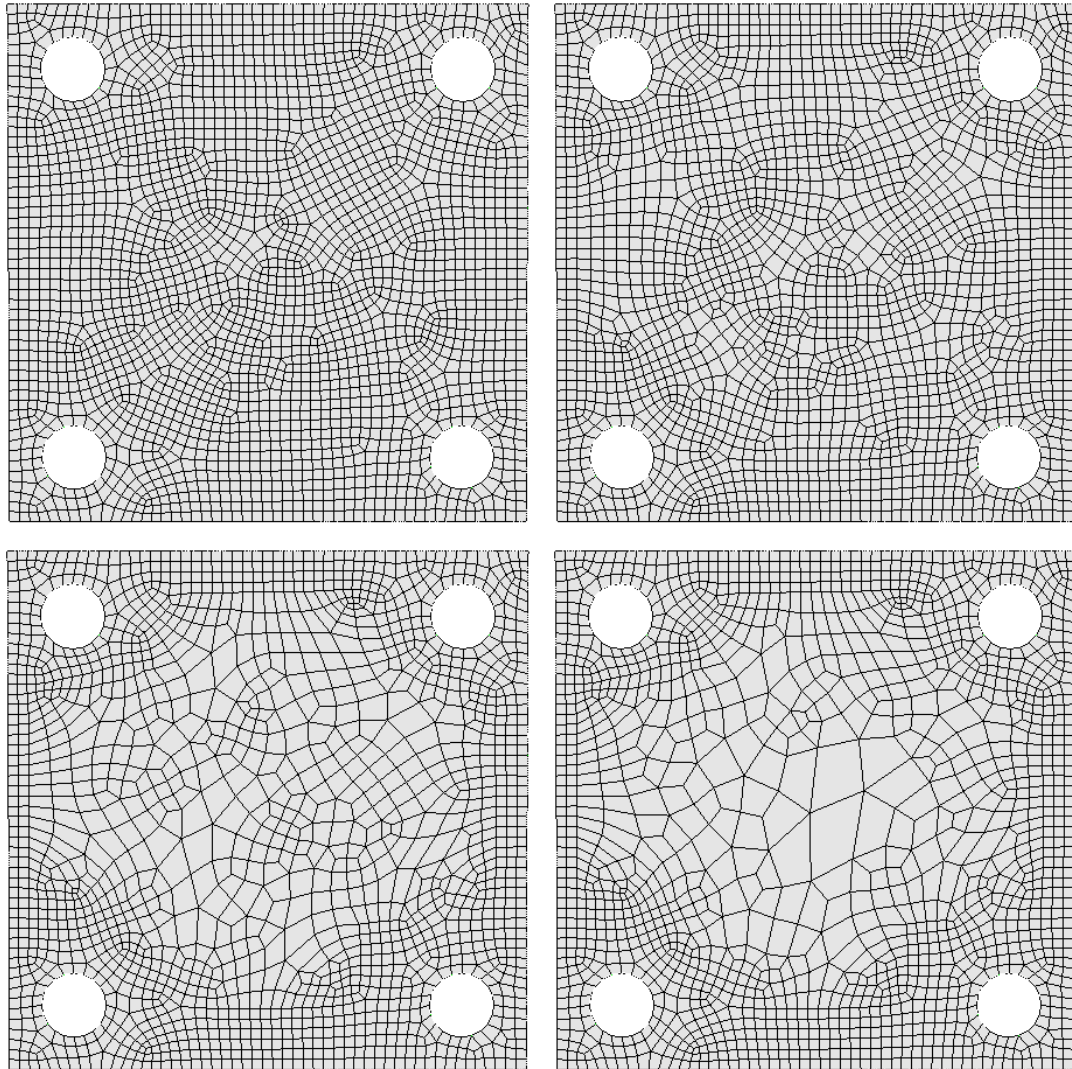


Figure 4.2: Membrane mesh at various levels of coarsening

Table 4.1: Membrane mesh quality results comparison

Test Name	Average Quality	Quality Std. Dev.	Minimum Quality	Maximum Quality
Membrane Mesh, Initial	0.976	0.044	0.719	1.000
Membrane Mesh, factor 1.5	0.965	0.052	0.705	1.000
Membrane Mesh, factor 4	0.946	0.069	0.556	1.000
Membrane Mesh, factor 10	0.946	0.070	0.579	1.000

Table 4.2 provides a comparison of the element areas of the four different meshes. The average area increases significantly. The maximum area of the elements in the mesh increases by a factor of 15 from the initial mesh to the factor 10 mesh. For reference, the number of elements within the initial coarsening region is 1126, the factor 1.5 coarsening removed 340 elements, the factor 4 coarsening removed 526 elements and the factor 10 coarsening removed 955 elements.

Table 4.2: Membrane mesh area results comparison

Test Name	Quad Count	% Quads Remaining	Average Area	Area Std. Dev.	Minimum Area	Maximum Area
Membrane Mesh, Initial	2429	100.0%	1.011	0.220	0.331	2.243
Membrane Mesh, factor 1.5	2089	86.0%	1.176	0.359	0.308	3.341
Membrane Mesh, factor 4	1563	64.3%	1.571	1.155	0.304	9.692
Membrane Mesh, factor 10	1474	60.7%	1.666	1.922	0.304	30.190

4.1.2 Swept Mesh Example

The boundary of a three dimensional mesh is of the most significance in some models. In computational fluid dynamics the boundary layer region may require a fine mesh. Figure 4.3 shows the circular cross-section of a cylinder. The initial mesh is shown in the top right corner. In this case the entire surface is the coarsening region. The other meshes shown have been coarsened to again to factors of 1.5, 4 and 10. The

original mesh is created with a sufficient number of elements at the boundary of the mesh; however this mesh density may not be necessary near the center of the mesh.

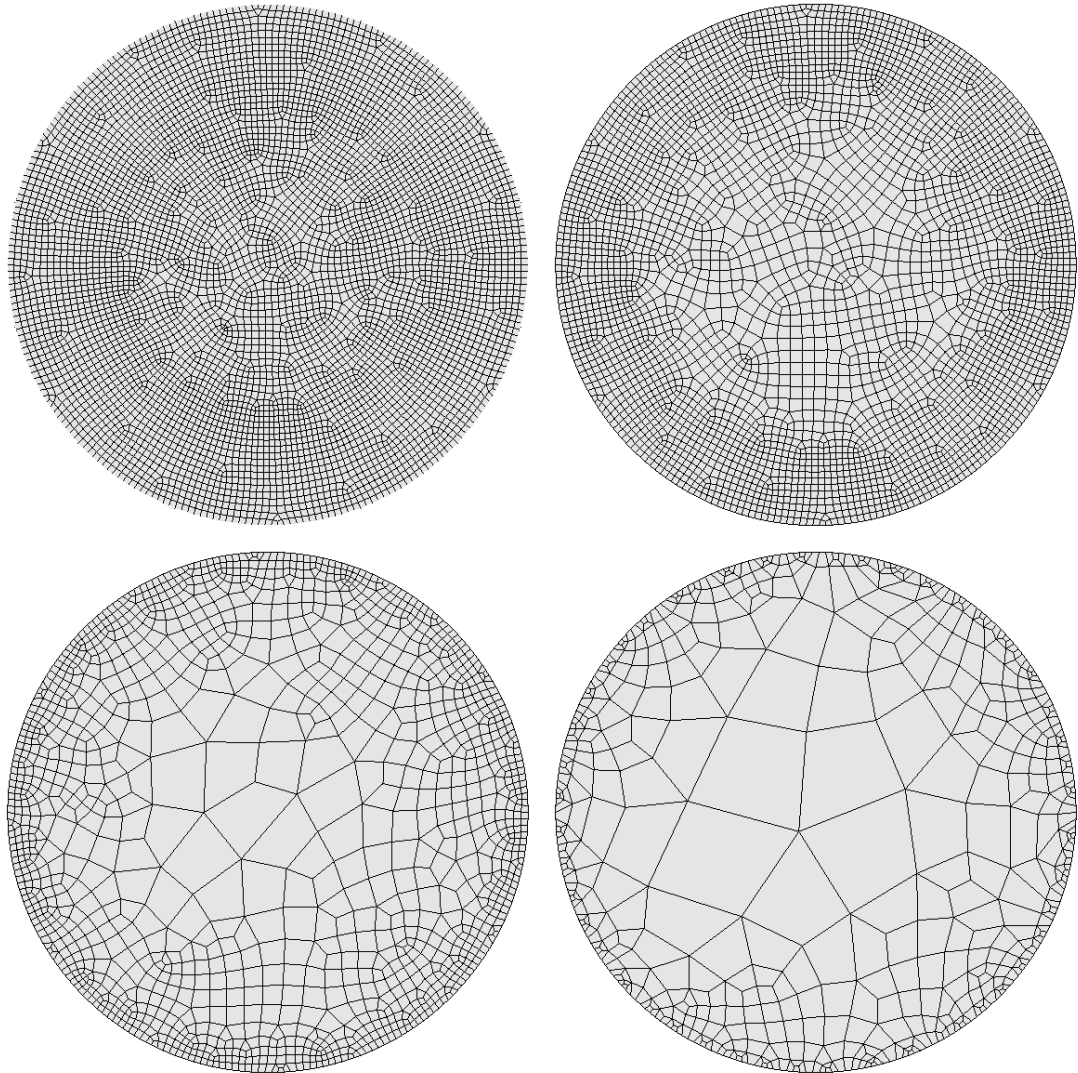


Figure 4.3: Cylinder mesh before and after coarsening

Table 4.3 and Table 4.4 show the quality and area results for these meshes respectively. The factor 10 coarsening attempt did not successfully complete the level of coarsening desired and the algorithm provided a warning that it was unable to coarsen

further. Once again it can be seen that as the mesh is coarsened more and more, the quality of the mesh is reduced. When coarsened to a factor of 10, the quality is barely within the acceptable range of quality. The transition between coarse and fine mesh in this example is very rapid and so the transition elements tend to have much shorter sides facing the outside of the mesh. This is typical in coarsening regions and is the primary reason for quality degradation.

After coarsening the surface can be swept through the cylinder to create an all-hexahedral mesh as shown in Figure 4.4. Densely packed, high-aspect ratio boundary elements are created around the perimeter with larger elements, good quality elements near the center.

Table 4.3: Swept mesh quality results comparison

Test Name	Average Quality	Quality Std. Dev.	Minimum Quality	Maximum Quality
Swept Mesh, Initial	0.974	0.041	0.755	1.000
Swept Mesh, factor 1.5	0.938	0.065	0.676	1.000
Swept Mesh factor 4	0.862	0.110	0.453	0.999
Swept Mesh factor 10	0.797	0.153	0.261	0.999

Table 4.4: Swept mesh area results comparison

Test Name	Quad Count	% Quads Remaining	Average Area	Area Std. Dev.	Minimum Area	Maximum Area
Swept Mesh, Initial	3233	100.0%	0.010	0.002	0.002	0.020
Swept Mesh, factor 1.5	2174	67.2%	0.015	0.006	0.002	0.055
Swept Mesh factor 4	801	24.8%	0.040	0.051	0.004	0.378
Swept Mesh factor 10	396	12.2%	0.077	0.153	0.003	1.053

4.1.3 Shell Mesh Example

For most FEA applications several different load cases are applied to the same model during the course of simulations. In these cases it is convenient to work from one base model which can quickly be adapted, by either refining or coarsening, to place computational emphasis on the areas of interest for a particular load case. Figure 4.5 shows a shell mesh of a nearly spherical model holes. Again the mesh is coarsened to factors of 1.5, 4 and 10. The number of edges at the surface boundary remains constant throughout the coarsening to preserve the edge count between itself and neighboring surfaces which are not coarsened.

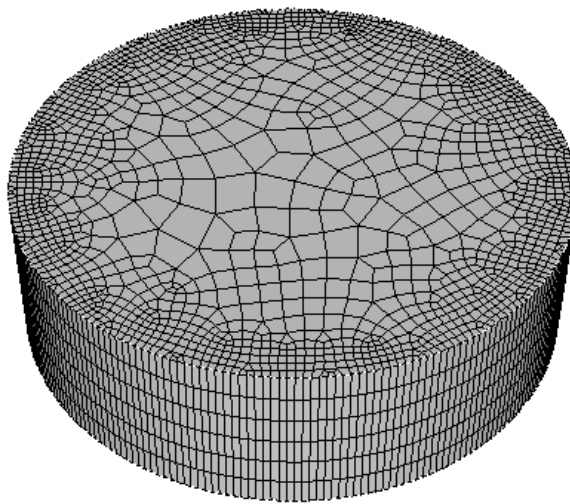


Figure 4.4: Swept cylinder mesh

In this case the factor 10 mesh has clearly been over coarsened. The curvature of the model is too high to be accurately captured by the large elements resulting from the coarsening procedure. However, the quality of the elements remains well within the acceptable range as is shown in Table 4.5.

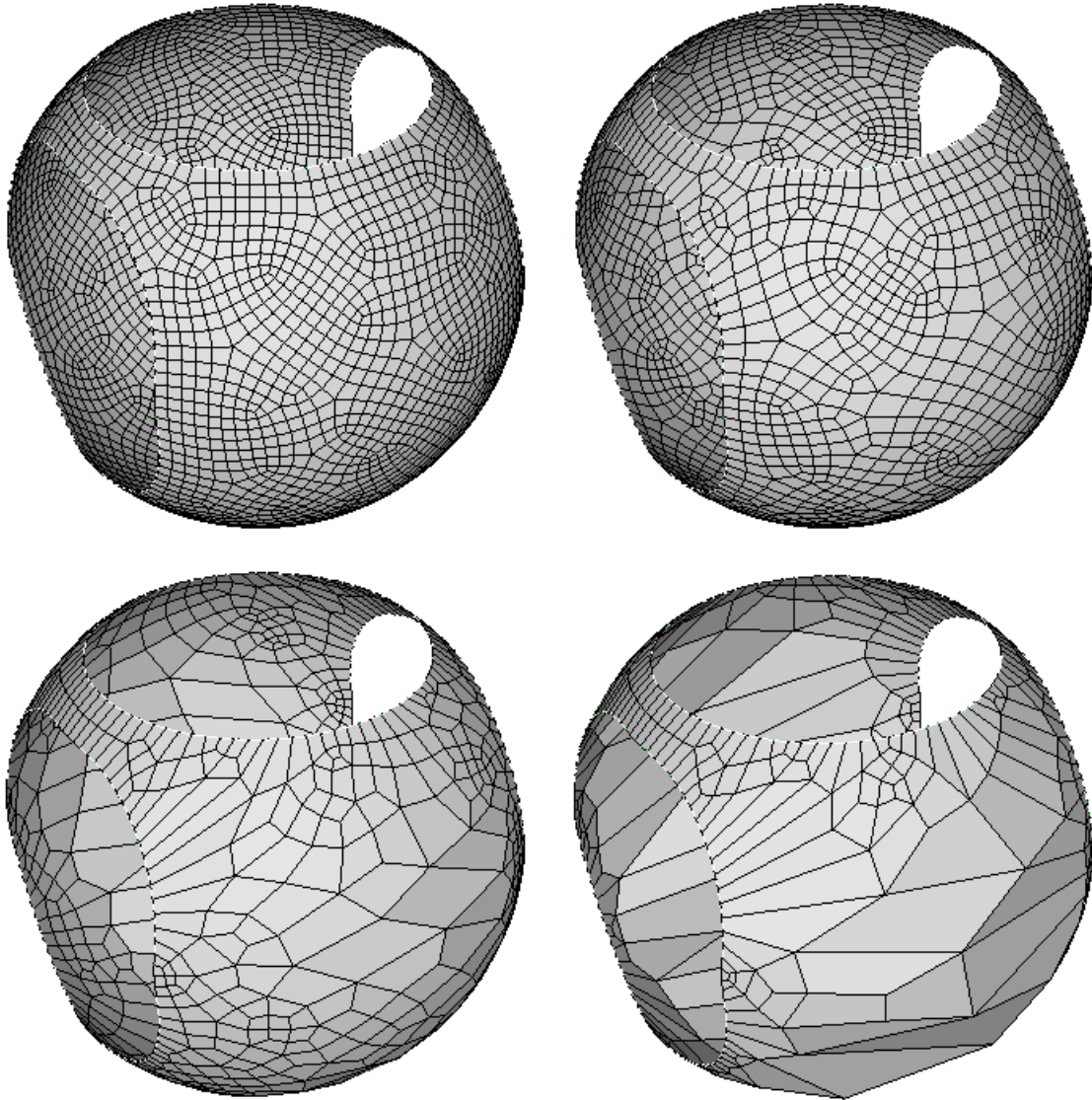


Figure 4.5: Shell mesh before and after coarsening

Table 4.5: Shell mesh quality comparison results

Test Name	Average Quality	Quality Std. Dev.	Minimum Quality	Maximum Quality
Shell Mesh, Initial	0.986	0.036	0.668	1.000
Shell Mesh, factor 1.5	0.980	0.043	0.656	1.000
Shell Mesh, factor 4	0.941	0.066	0.639	1.000
Shell Mesh, factor 10	0.843	0.093	0.499	0.995

Table 4.6 below shows a comparison of the areas and quads remaining in the original mesh. The algorithm has faithfully removed quads according to the coarsening factors of 1.5, 4 and 10. The coarsened factor is most readily apparent when examining the average area column.

Table 4.6: Shell mesh area comparison results

Test Name	Quad Count	% Quads Remaining	Average Area	Area Std. Dev.	Minimum Area	Maximum Area
Shell Mesh, Initial	5477	100.0%	0.014	0.003	0.002	0.037
Shell Mesh, factor 1.5	3638	66.4%	0.022	0.013	0.006	0.119
Shell Mesh, factor 4	1347	24.6%	0.058	0.105	0.007	1.280
Shell Mesh, factor 10	572	10.4%	0.137	0.349	0.007	3.758

5 Conclusions and Recommendations

5.1 Magnitude of Coarsening and Quality Considerations

The magnitude of coarsening to be achieved is limited primarily by element quality considerations. Generally speaking, the more the mesh is coarsened, the lower the quality of the mesh becomes. This can largely be attributed to the quality of the transition elements between the fine and coarse regions of the mesh degrading as the difference in sizes between the two mesh regions increases. At least one layer of elements must take on a trapezoidal shape where the edges near the center of the region are larger than the edges near the outside of the region. As the difference in length becomes more pronounced the quality of the mesh is reduced. This should not deter from the use of the coarsened mesh so long as the quality remains within acceptable ranges because typically the area of the mesh being coarsened is outside of the area where accuracy is most critical.

Consideration of the location of the transition elements also gives insight into the best way to choose coarsening rings. Instead of doing concentric coarsening rings, many smaller coarsening rings, perhaps centering around a single element, could have been created. Or rather than creating and considering several coarsening rings, only the outermost ring might be processed and removed automatically. Both of these options

were explored and rejected because the quality of the mesh created were significantly lower than when several concentric rings were used. In the first case, where many small non-concentric rings are used, the number of transition elements goes up sharply throughout the mesh. In the second case where only the outermost ring is used repeatedly, the coarsening often does not touch the center region and the transition elements are focused on the outermost rings of the region. This sharp transition results in low quality meshes like those in the factor 10 examples but occurring on much lower levels of coarsening. Using many concentric rings may not be the best method of choosing rings, but it has proven to be the most effective of the techniques tried so far.

Due to the triangularization of the stiffness matrix, an analysis loop for finite element analysis has a computational complexity proportional to the number of nodes in the mesh cubed. In the case of the swept mesh the number of nodes in the original mesh is 5603. The mesh after being coarsened to a factor of 1.5, 4 and 10 is reduced to 3764, 1473 and 698 nodes respectively. This reduction in node count would roughly correspond to a reduction in computation time by factors of 3, 50 and 500 respectively. This kind of speed increase is invaluable for large scale meshes that take hours or days to analyze.

5.2 Summary

The availability of a fast, robust coarsening algorithm for unstructured, all-quadrilateral, conformal meshes expands the tools available for computational modeling. The use of coarsening coupled with refinement provides an effective means to adapt a mesh more effectively, increasing model accuracy while reducing computation times.

This paper presented the Automated Quadrilateral Coarsening by Ring Collapse (AQCRC) algorithm. The algorithm conformally coarsens localized regions of quadrilateral meshes by creating, modifying and removing coarsening rings. The automation procedure optimizes final element quality while attempting to heavily coarsen the defined region.

Application of the AQCRC algorithm demonstrates that it is capable of removing enough nodes from the mesh to increase speeds by many times while maintaining a mesh quality sufficient for accurate analysis. In the examples shown the algorithm was capable of increasing speed by a factor of 3 without significantly decreasing the quality of the mesh or of increasing speed by two orders of magnitude while maintaining generally acceptable element quality.

5.3 Further Research

Further research in applying this algorithm is indicated in several areas. Clean-up operations are already a focus of significant effort to address problems coarsening around complex surfaces. A three dimensional version of this algorithm for use in hexahedral meshes is being developed [21]. An alternative method of coarsening, partial chord removal, is described in Appendix B. A basic version of this algorithm has been implemented and with further research could be used in conjunction with AQCRC coarsening or as a stand alone coarsening method.

One of the methods of coarsening discussed in the Chapter 2 was undoing refinement. The AQCRC algorithm could be adapted into a reversible coarsening procedure, allowing refinement of regions previously coarsened by storing the changes

made to the mesh in the remaining mesh entities. As some nodes or edges may be merged and then merged again, a set of changes could be pushed onto a stack so that the reverse operations could be performed one at a time, restoring the mesh to its original state.

The AQCRC algorithm is a step forward in coarsening methodology for quadrilateral finite element meshes. The potential speed benefits of coarsening algorithms could significantly reduce the time and cost of computational analysis of large scale problems.

References

1. Bathe K (1996) Finite Element Procedures, Prentice Hall, Upper Saddle River, New Jersey.
2. Schneiders R (2000) Octree-based hexahedral mesh generation, *Int. Journal of Comp. Geom. & Applications*, vol. 10, no. 4, p 383-398.
3. Parrish M (2007) A Selective Approach to Conformal Refinement of Unstructured Hexahedral Meshes, MS Thesis, Brigham Young University, Utah.
4. Garland M, Heckbert P (1997) Surface Simplification Using Quadric Error Metrics, Carnegie Mellon University, published online at <http://graphics.cs.uiuc.edu/%7Egarland/papers/quadrics.pdf>.
5. Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W (1993) Mesh Optimization, University of Washington, published online at <http://citeseer.ist.psu.edu/cache/papers/cs/797/ftp:zSzzSzftp.cs.washington.edu/ztrSz1993zSz01zSzUW-CSE-93-01-01.pdf/hoppe93mesh.pdf>.
6. Cignoni P, Montani C, Scopigno R (1997) A Comparison of Mesh Simplification Algorithms, *Computers & Graphics*, v 22, n 1, Jan-Feb 1998, p37-54.
7. Silva S, Silva F, Madiera J, Santos B (2007) Evaluation of Mesh Simplification Algorithms Using PolyMeCo: A Case Study, *Proceedings of SPIE – The International Society for Optical Engineering*, v 6495, *Proceedings of SPIE-IS and T Electronic Imaging - Visualization and Data Analysis 2007*, 2007, p 64950D.
8. Silva S, Madiera J, Ferreira C, Santos B (2007) Comparison of Methods for the Simplification of Mesh Models Using Quality Indices and an Observer Study, *Proceedings of SPIE – The International Society for Optical Engineering*, v 6492, *Proceedings of SPIE-IS and T Electronic Imaging - Visualization and Data Analysis XII*, 2007, p 64921L.
9. Takeuci S, Kanai T, Suzuki H, Shimada K, Kimura F (2000) Subdivision Surface Fitting Using QEM-based Mesh Simplification and Reconstruction of

- Approximated B-spline Surfaces, Computer Graphics and Applications, 2000. Proceedings. The Eighth Pacific Conference on 3-5 Oct. 2000 Pages 202-212.
10. Cheng A, Zhong Z (2002) Local Coarsening for Quadrilateral Meshes on Autobody Tool Surface, *Journal of Computer-Aided Design and Computer Graphics*, v 14, n 1, January, 2002, p 50-55.
 11. Kwak D, Cheon J, Im Y (2002) Remeshing for metal forming simulating-Part 1: Two-dimensional Quadrilateral Remeshing, *International Journal for Numerical Methods in Engineering*, v 53, n 11, Apr 20, 2002, p 2463-2500.
 12. Choi C (2004) Adaptive Mesh Refinement/Recovery Strategy for FEA, *Structural Engineering and Mechanics*, March/April 2004, vol 7, iss 3-4, pages 379-391.
 13. Nikishkov G (2005) Finite Element Algorithm with Adaptive Qaudtree-Octree Mesh Refinement, *Anziam J.* 46 (E) pages C15-C18.
 14. Murdoch P, Staten S, Blacker T, and Mitchell S (1997) The Spatial Twist Continuum: A Connectivity Based Method for Representing All Hexahedral Finite Element Meshes. *Finite Element Analysis and Design*, 28(2), 137-149.
 15. Borden M, Staten S, Shepherd J (2002) Hexahedral Sheet Extraction. 11th International Meshing Roundtable, Sandia National Laboratories, September 2002, 147-152.
 16. Benzley S, Harris N, Scott M, Borden M, and Owen S (2005) Conformal Refinement and Coarsening of Unstructured Hexahedral Meshes. *Journal of Computing and Information Science Engineering*, December 2005, Vol 5, 330-337.
 17. Staten M, Benzley S, Scott M (2008) A Methodology for Quadrilateral Finite Element Mesh Coarsening, to be published in *Engineering with Computers*.
 18. Knupp P (2000) Achieving Finite Element Mesh Quality via Optimization of the Jacobian Matrix Norm and Associated Quantities, *International Journal for Numerical Methods in Engineering*, vol 48, issue 8, p. 1165-1185.
 19. Kinney P (1997) CleanUp: Improving Quadrilateral Finite Element Meshes, 6th International Meshing Roundtable, Sandia National Laboratories, October 1997, 449-461.
 20. Shepherd J (2007) Topologic and Geometric Constraint Based Hexahedral Mesh Generation. PhD Thesis, University of Utah, Utah.
 21. Benzley S, Dewey M, Woodbury A, Staten M (2008) Enhanced Mesh Adaptation Using Localized Conformal Quadrilateral and Hexahedral Coarsening, to be

published in the Inaugural International Conference of Engineering Mechanics Institute.

22. Freitag L, Plassmann P (1997) Local Optimization-Based Simplicial Mesh Untangling and Improvement, Technical report, Mathematics and Computer Science Division, Argonne National Laboratory. Found online at: <http://citeseer.ist.psu.edu/277987.html>

Appendix A. Basic Chord Operations

The core of the idea put forward by Staten, et al. [16], is that if a bounding set of partial chords is selected, the chords can be united by a set of simple operations creating a circular chord entirely contained within the coarsening region. The operations are: edge swap, face close and doublet insertion. These original operations are no longer used in the AQCRC algorithm except in the context of clean-up operations; however, two additional operations, the template insertion and the face open operations were later developed and applied in the AQCRC. All of these operations are described here in detail for reference.

Edge Swap Operation

One of the simplest operations is the edge-swap. Figure A.1 demonstrates how an edge-swap would be applied to the two bounding partial chords. The panel on the top left shows the original partial chords at the intersection marked by a dotted line. The panel on the top right shows the resulting mesh after the edge swap has been performed; the two partial chords have now become one chord. The other two panels show the results of the face close and doublet insertion operations and will be discussed later. At first, the edge-swap creates a poor quality element; however, this element will easily become a high quality element with the application of a simple smoothing operation.

This method of joining two chords generally results in very good quality elements once the chord has been collapsed from the mesh. However, it reduces the number of quads collapsed from the mesh by one. If the low quality element is adjacent to a geometric curve, a simple smoothing operation is not sufficient to redeem the low quality quad.

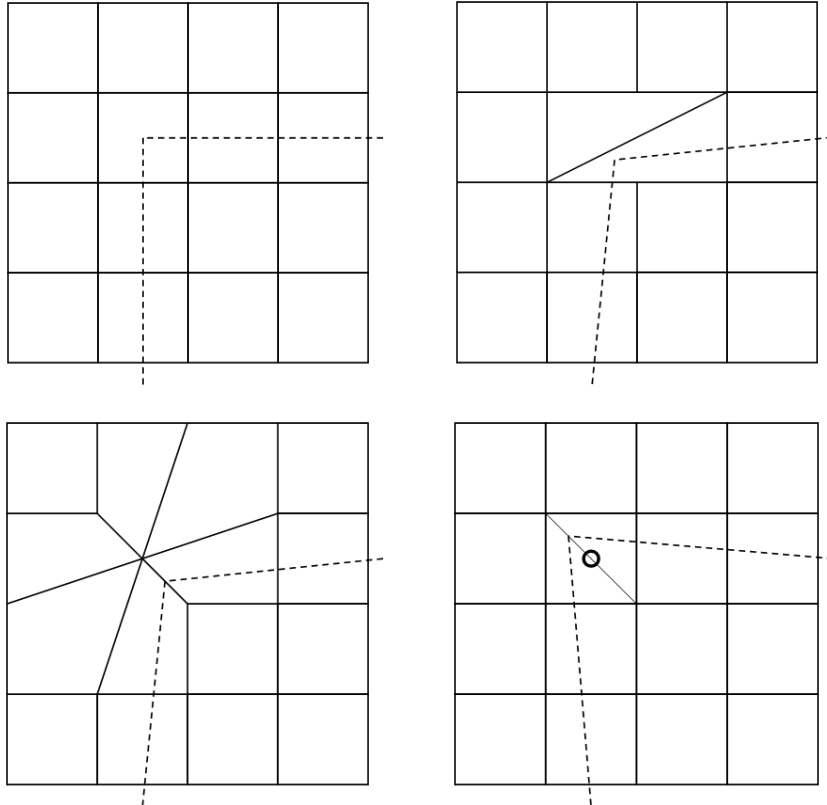


Figure A.1: The edge swap, face close and doublet insertion operations

Face Close Operation

Figure A.1 also shows the face close operation, the two partial chords are again combined, this time as shown in the bottom left panel. The face close operation is quick, simple and effective. At first it appears to create a high-valent node, which is generally considered bad due to the poor quality elements around it; however, once the joined

chord has been collapsed the node will be reduced again to a node with an acceptable valence. The draw back to this operation is that it generally produces slightly lower quality elements compared to the edge swap operation. The face close operation is very useful in a number of clean-up operations and can be seen to be the simplest method of coarsening, albeit on a very small scale.

The Doublet Insertion Operation

The doublet insertion operation is another simple way of combining the two partial chords. Initially it creates very poor quality elements; however, both of these elements are removed in the process of coarsening, making this operation perfectly acceptable in the long run. The doublet insertion operation is also used as an intermediate step in a number of clean-up operation. Typically one of the edges in the doublet is split during a face open operation immediately following the doublet insertion. Interestingly, when applied as part of the coarsening process, the face close and doublet insertion algorithms produce an identical mesh once the chord has been collapsed.

Template Insertion Operation

The template insertion operation is another option that can replace face close and edge swap operations in combining two partial chords. A simple one-to-three template is inserted into the quad just inside the intersection of the bounding chords. Edges are redistributed and created according to the right panel of Figure A.2.

Template insertion is advantageous because it guarantees that no poor quality elements or high-valent nodes are created near the bounding chords. However, it adds

two elements to the mesh and thus should be used sparingly in coarsening operations. The template insertion method can be used to defuse problematic boundary conditions as discussed in Section 3.3.

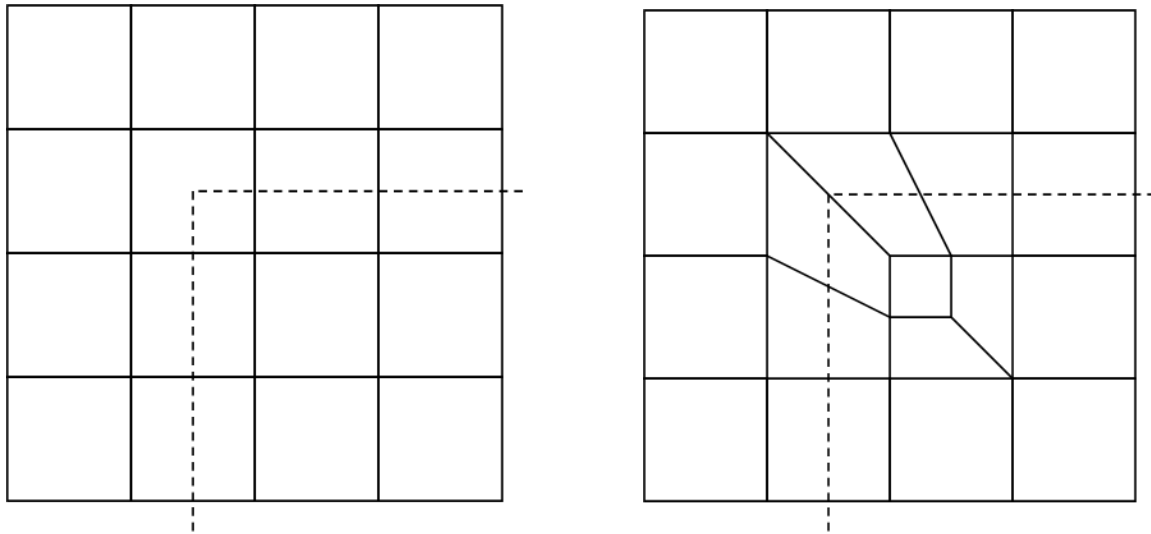


Figure A.2: The template insertion operation

Face Open Operation

The face open operation was added to the code because of a discovered need during the coding of the algorithm. It was found that commonly there were multiple bounding partial chords with intersections adjacent to each other. Figure A.3 shows such a situation in the left panel. The right panel shows the resulting combination of all three partial chords into one continuous chord by the application of a single face open operation. Each additional chord intersection requires an additional face open operation.

Note that if two face close operations or two edge swap operations were performed, the resulting collapse of the chord would create a high-valent node. Worse, if several of these jagged bounding chords were adjacent to each other there would be no

limit to the valence of the node created by the collapse of the chord, and the edge swap operation could not be used for more than the simple three chord case. The face open operation is an essential tool in joining three or more partial chords into a continuous bounding chord.

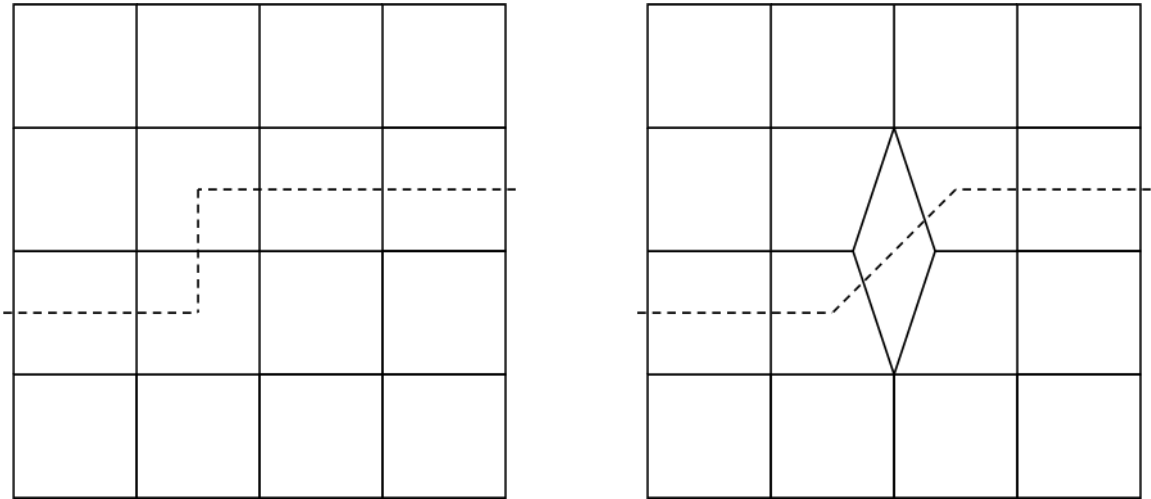


Figure A.3: The face open operation

Appendix B. Partial Chord Removal

Concurrently with the research presented in the main body of this paper, another algorithm appropriate for localized coarsening on quadrilateral finite element meshes has been developed. The algorithm, called Partial Chord Removal, is an excellent complement to the ring coarsening algorithm. It seems likely that with the development of intelligent decision making processes, a given coarsening region could have both ring coarsening and partial chord coarsening applied simultaneously and with great effect.

Single Face Collapse

The simplest coarsening method possible is the removal of a single element with a face close operation as shown in Figure B.1. Face close operations are not used to coarsen because they leave behind a high-valent node. As can be seen in the figure, this structured mesh now includes a six valent node. This tends to reduce the quality quickly and has been neglected as a valid means of large scale coarsening.

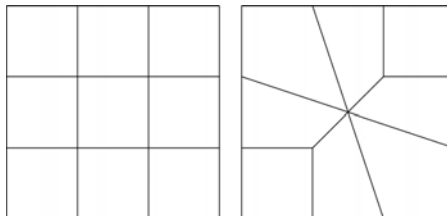


Figure B.1: Face close operation

Stringing Face Closes Together

Partial chord coarsening is simply a series of face close operations adjacent to each other along partial chord. Figure B.2 below shows a mesh being coarsened by partial chord removal. In the left panel, the nodes circled with dashed lines between them will be collapsed to the same point. In the right panel the final mesh configuration is shown. The region highlighted in grey is the partial chord selected for coarsening.

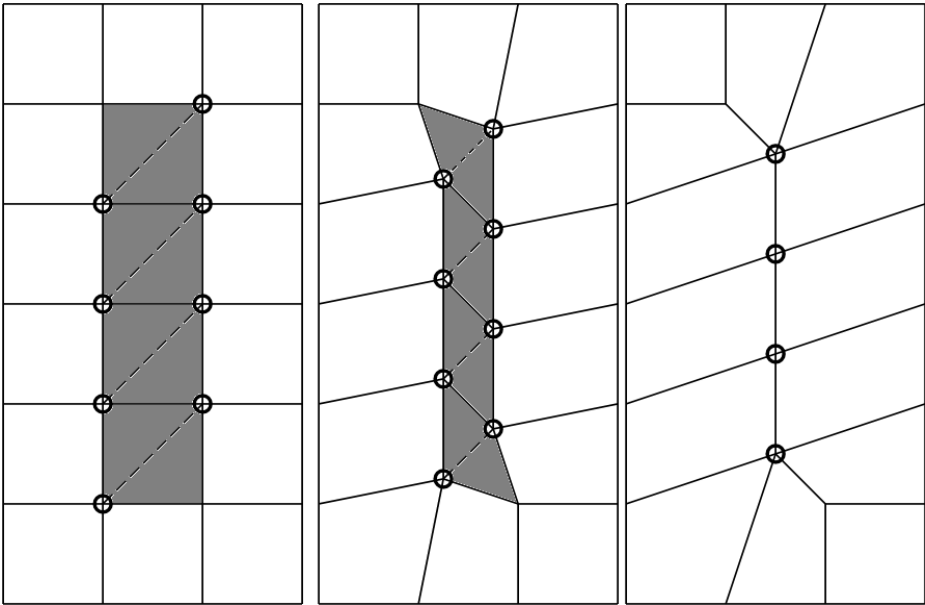


Figure B.2: Partial chord coarsening

Partial chord removal has two limitations. First, at least two quadrilaterals should be removed at the same time in order to avoid creating high-valent nodes. Along with this limitation, the end quadrilaterals should not collapse nodes with a valence higher than four or high-valent nodes will be created. Second, the partial chord cannot extend through or adjacent to a curve boundary. In either case, it will create a triangle-shaped quadrilateral as discussed in Appendix C.

Appendix C. Clean-up Operations

Clean-up is an increasingly important part of the coding of the AQCRC algorithm. In its current development stage there are still too many unknown factors that may result in the creation of very poor elements which cannot be smoothed into being good quality elements by any ordinary means. Clean-up operations are therefore important to ensuring that the mesh left after the coarsening operations have executed is still valid. Included in the coarsening suite is a simple smoother which is used to both enhance the functionality of the other coarsening tools and to allow the coarsening ring quality projections to be more accurate. The clean-up operations and smoother now in use in the AQCRC algorithm are detailed in the following sections.

Smoothing

The mesh is smoothed each time a coarsening iteration is performed so that the quality metrics used during coarsening more accurately reflect the actual location of nodes in the final mesh. It is also smoothed during each iteration of the clean-up operations because some of the clean-up operations are only activated on low quality elements. The smoothing algorithm during clean-up is a simple laplacian smoothing algorithm [22]. Each node is considered in turn and moved to the centroid of the quadrilaterals surrounding it. This procedure is repeated several times for all the nodes in

the mesh until the maximum distance moved by the nodes has been reduced to a convergence tolerance or a maximum number of iterations have been performed.

Diamond Quadrilateral Removal

A diamond quadrilateral is defined as a quadrilateral with two opposing three-valent nodes. These quadrilaterals are not a major concern, but reduce the overall quality of the mesh. They are removed by simply performing a face close operation, bringing the two three-valent nodes together to form a four-valent node as shown in Figure C.1. If a quadrilateral has more than two three-valent nodes it is not considered a diamond quadrilateral because the collapse of the quadrilateral would result in the creation of a doublet.

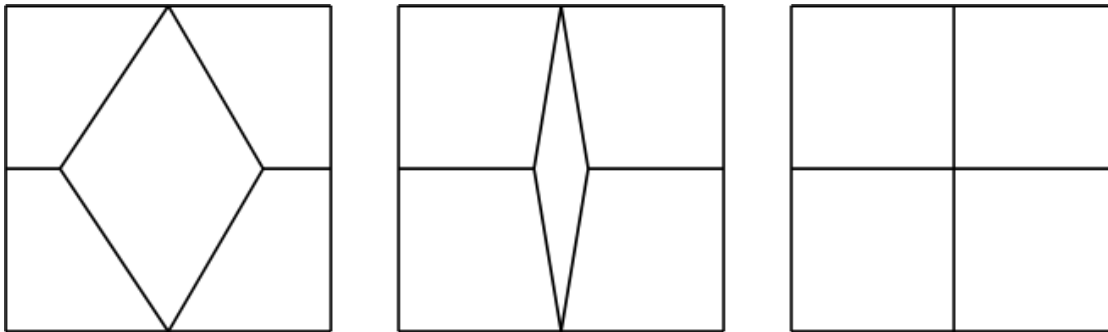


Figure C.1: Diamond quadrilateral removal

Doublet Removal

A doublet is a node with a valence of two. They are easily removed by performing a simple face close operation on one of the two adjacent quadrilaterals; thus, one quadrilateral takes the place that the two quadrilaterals previously occupied. Figure C.2 shows a doublet being removed. In a few cases it is not practical to delete one of the

quadrilaterals, either because it will create a triangle-shaped quadrilateral, or because both quadrilaterals were part of the original mesh just outside of the coarsening region. If a face close operation is not feasible, a second option may be employed by performing a face open operation on one of the edges connected to the doublet node and on another edge not connected to the doublet quadrilaterals. The doublet becomes a three-valent node. Inserting a doublet and removing it in this way is one of the primary tools used in resolving triangle-shaped quadrilaterals as discussed and illustrated in the Triangle-Shaped Quadrilateral Removal Section below.

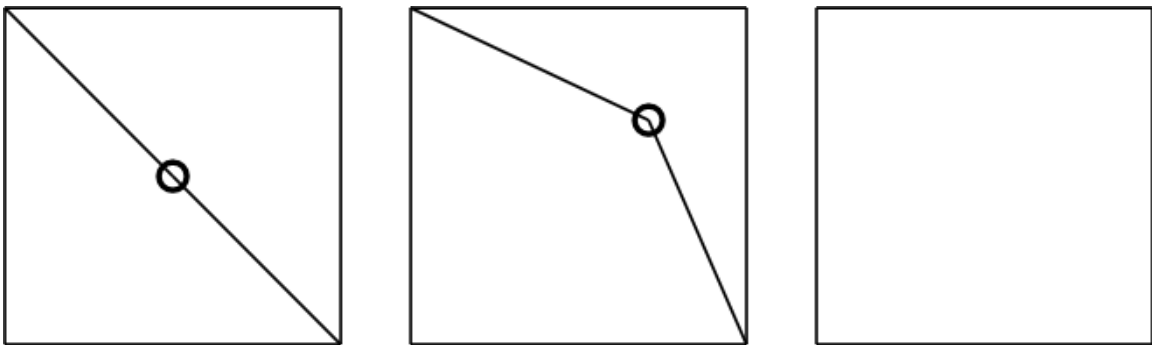


Figure C.2: Simple doublet removal

High-Valent Node Reduction

Nodes with a valence of more than five are considered high-valent nodes and tend to reduce the quality of the mesh. There are two major ways of dealing with these problems. First, two of the edges connected to the node may be opened. This results in two nodes, each with lower valence than the first. Second, a face close operation on one of the neighboring quadrilaterals will reduce the valence of the node. Both operations have been illustrated in Figure C.3. The top panel shows the original mesh with the

high-valent node circled for clarity. The face open operation is shown on the left; the face opened has extra bold lines. The face close operation is on the right; the two edges remaining where the face was previously to be are shown in extra bold lines.

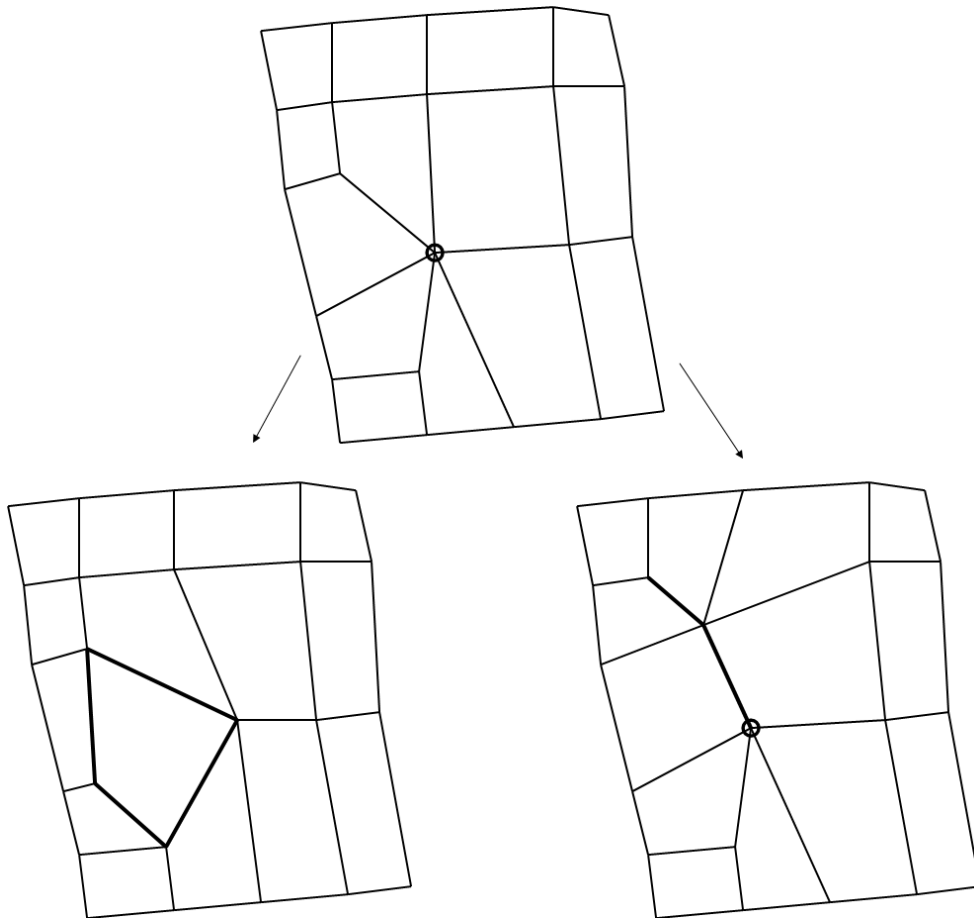


Figure C.3: Resolution options for high-valent nodes

In the face open operation the edges intersecting at the node are split between the two new nodes that take the place of the original node. However, because the edges of two of the adjacent nodes are split, those adjacent nodes increase in valence. This is advantageous if those nodes were three-valent nodes to begin with, and acceptable if they are four-valent nodes, but it may simply move the high-valent problem to another

location if one of the nodes has a higher valence than four. In the face close operation a similar dilemma is encountered: the two nodes which are merged in the face close operation combine their edges. If they are both three-valent nodes they will become a four-valent node, and if they are one three-valent node and one four-valent node they will become a five-valent node. However, if they have any higher combination of valence they will become a high-valent node themselves. There is no guarantee that there are two nodes of appropriate valence on the same face as is required for the face close operation, nor on different faces as is required for the face open operation; however, it is very common to find one or more in either of those possibilities. The program searches for the lowest valence nodes neighboring the high-valent node and performs a face close operation if they are on the same quadrilateral, or a face open operation if they are not on the same quadrilateral. In the example shown in Figure C.3 both options are equally valid and the decision to do one or the other is made arbitrarily.

Triangle-shaped Quadrilateral Removal

If a quadrilateral has two edges on the same geometry curve it may have a distinctly triangular shape. Such an element often has a low quality; because of the restrictions of the node locations on the curve, the shape quality cannot be improved by smoothing. In some cases, such as a curve with high curvature, this may be acceptable or even preferred; therefore, the quality of the quadrilateral is checked with the scaled Jacobian metric before having the clean-up operation performed. If the quality is above marginal, it is left alone. Figure C.4 shows these two cases. On the left is a triangle-shaped quadrilateral on a straight curve; the third node on the curve is circled for clarity,

and the double line is where the geometry curve lies. The right panel shows a quadrilateral that is topologically the same, and is only distinguished by having an acceptable quality.

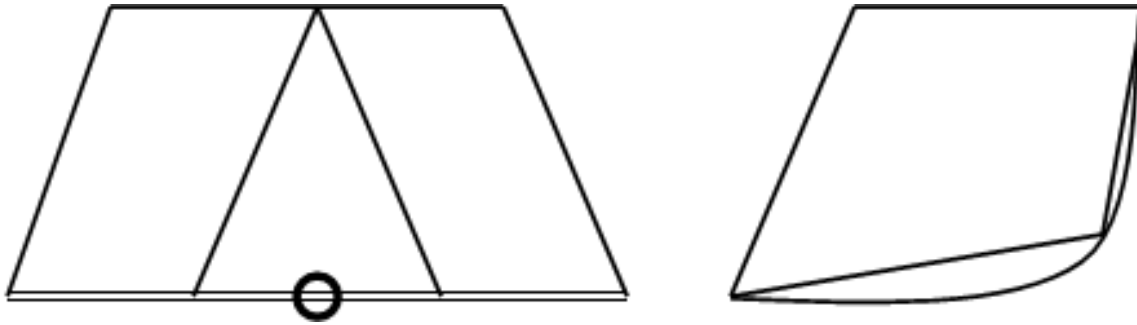


Figure C.4: Triangle-shaped quadrilaterals

As with the high-valent node case, triangle-shaped quadrilaterals have several options for remediation depending on the specific situation. The primary method used is to insert a doublet into the quadrilateral running from the circled node on the curve to the node opposite and then opening the far edge with another from the surrounding mesh. Figure C.5 depicts this method. On the left a doublet is inserted into the triangle-shaped quadrilateral from Figure C.4; the doublet node is circled. On the right a face open operation has been performed changing the doublet into a three-valent node. It should be noted that the other edge in the operation may increase the valence of the other node above acceptable levels. This is permitted temporarily because it is better to have a high-valent node away from the restrictions imposed by the geometry curves than to have a triangle-shaped quadrilateral. Even though it is acceptable to create a high-valent node, it is not preferred and the other edge in this operation is chosen to avoid this problem if possible.

Since the triangle-shaped quadrilateral has two edges on the same curve, only one of the nodes of the quadrilateral is not on the curve. If that node is on another curve, the standard procedure cannot be used. The solution, shown in Figure C.6, is to insert a doublet in the triangle-shaped quadrilateral and one of the adjacent quadrilaterals. An edge-swap operation of the edge between the original two quadrilaterals will then resolve the situation. This method generally results in marginal quality elements, but is still a significant improvement over the situation of the mesh prior to the operation.

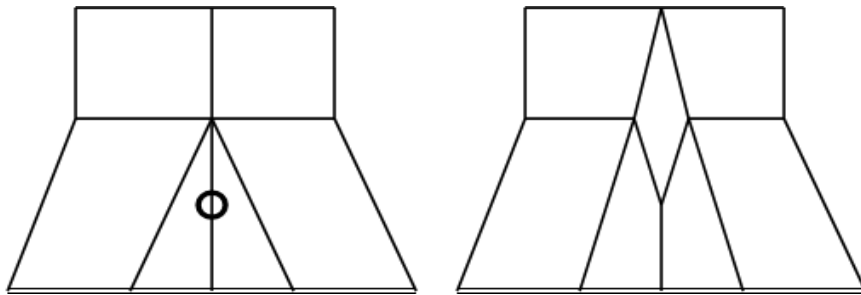


Figure C.5: Basic resolution to triangle-shaped quadrilaterals

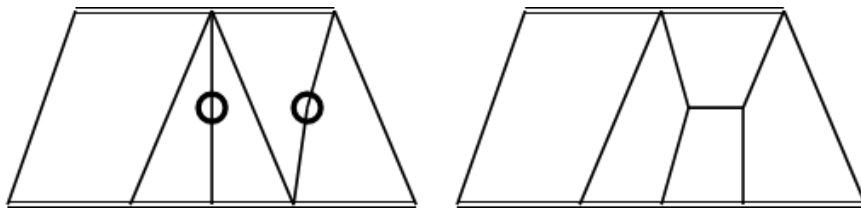


Figure C.6: Solution for triangle-shaped quadrilateral between two curves

Appendix D. Implementation Concerns in CUBIT

The AQCR algorithm is implemented into CUBIT. The following sections describe the command structure in CUBIT, a description of a temporary data structure developed to perform coarsening, and an algorithm flow chart.

CUBIT Command Parameters

Commands in CUBIT are generally denoted with the following syntax. Curly braces, { }, indicate that one of several parameters within the braces is required. The options are separated by a single vertical line, |, symbol. Angle brackets, < >, indicate that a number is required, between the brackets is the type of number (integer, range or double) that is allowed. Square brackets, [], indicate one or more optional parameters which may be separated by vertical line symbols if only one of the optional parameters is allowed. The command syntax for coarsening is as follows: `Coarsen {Surface | Curve | Vertex | Face | Edge | Node} <range> [Depth <int>] {Factor <double> | Size <double>}`.

Temporary Data Structure

Immediately after the command is read in, the cubit mesh entities within the region are determined and copied into a data structure designed and optimized specifically for the coarsening algorithm. This data structure mirrors the CUBIT data

structure in that it has nodes, edges and faces. Each face has edge uses which contain the edges and a definition of whether the edge nodes run clockwise or counter-clockwise on the face. These entities are created by and stored in a container class which further contains most of the utility functions used to modify and adapt the mesh during coarsening. This container class also contains the functions necessary to convert back and forth between the CUBIT entities and the temporary entities it uses.

The temporary entities all contain a pointer to the CUBIT entity from which they are copied (or NULL if they are a newly created entity), and the CUBIT entities which have been converted into adapt entities are stored in a map which couples them with the temporary entities.