

# **Anisotropic Mesh Adaptation: Recovering Quasi-structured Meshes**

by

Mahkame Sharbatdar

B.Sc., Mechanical Engineering, University of Tehran (Iran), 2010

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF  
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

The Faculty of Graduate Studies

(Mechanical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

November 2012

© Mahkame Sharbatdar 2012

# Abstract

An adaptive method for producing anisotropic quasi-structured meshes is presented in this thesis. Current anisotropic adaptation schemes produce meshes without any regular structure which can hurt accuracy and efficiency of the solution. By modifying the anisotropic adaptation schemes, producing aligned, quasi-structured meshes is possible which means that the accuracy and efficiency of the flow solution are improved. By using quasi-structured meshes, we can get the advantages of flexibility of unstructured meshes for complex geometries and accuracy of the high directional qualities of the structured meshes at the same time. The construction of the quasi-structured meshes from initial isotropic unstructured meshes is accomplished by assigning metrics to vertices based on the error estimation methods. The metrics are used to communicate the desired anisotropy to the meshing program. After assigning a metric to each vertex, the mesh is refined anisotropically using four mesh quality improvements operations to produce high quality anisotropic quasi-structured meshes: swapping to choose the diagonal of the quadrilateral formed by two neighboring triangles which results the maximum quality, inserting vertices for large triangles, vertex removal to eliminate small edges and vertex movements to optimize the location of the vertices so that quasi-structured meshes are created. The idea in the optimization process is to smooth a vertex location by seeking so that the final mesh contains target elements dictated by the metrics assigned in the three vertices of that triangle. The final, high quality mesh is produced by using these operations iteratively based on the metrics assigned to each vertex in an adaptive, solution-based process.

# Preface

The research ideas and methods explored in this thesis are the fruits of a close working relationship between Dr. Carl Ollivier-Gooch and Mahkame Sharbatdar. The implementation of the methods, the data analysis, and the manuscript preparation were done by Mahkame Sharbatdar with invaluable guidance from Carl Ollivier-Gooch throughout the process.

# Table of Contents

<b>Abstract</b>	ii
<b>Preface</b>	iii
<b>Table of Contents</b>	iv
<b>List of Tables</b>	vi
<b>List of Figures</b>	vii
<b>Nomenclature</b>	ix
<b>Acknowledgments</b>	xi
<b>1 Introduction</b>	1
1.1 Motivation	2
1.2 Mesh Classification	3
1.2.1 Structured and Unstructured Meshes	3
1.2.2 Quasi-structured Meshes	5
1.2.3 Isotropic and Anisotropic meshes	5
1.3 Objective and Outline	8
<b>2 Background</b>	11
2.1 Error Estimation Methods	11
2.1.1 Interpolation-based Error Estimation	13
2.1.2 Feature-based Error Estimation	14
2.1.3 Adjoint-based Error Estimation	15
2.2 Metric-based Anisotropic Mesh Adaptation	16
2.3 Centroidal Voronoi Tessellation	22



2.4	Optimization Algorithm . . . . .	23
2.4.1	Target Matrix Optimization Paradigm . . . . .	24
<b>3</b>	<b>Methodology . . . . .</b>	<b>29</b>
3.1	Mesh Quality Measures . . . . .	29
3.2	Mesh Operations . . . . .	32
3.2.1	Face Swapping . . . . .	33
3.2.2	Vertex Insertion . . . . .	34
3.2.3	Vertex Removal . . . . .	37
3.2.4	Vertex Movement . . . . .	39
3.2.4.1	Average Metric . . . . .	39
3.2.4.2	Isotropic Elements . . . . .	40
3.2.4.3	Anisotropic Elements . . . . .	41
3.2.4.4	Orientation and Order . . . . .	42
3.3	General Algorithm . . . . .	44
<b>4</b>	<b>Results . . . . .</b>	<b>47</b>
4.1	Preliminary Tests . . . . .	47
4.2	Verification Case . . . . .	51
4.3	Test Cases . . . . .	55
4.3.1	Subsonic viscous flow around an airfoil . . . . .	57
4.3.2	Transonic inviscid flow around an airfoil . . . . .	63
<b>5</b>	<b>Concluding Remarks . . . . .</b>	<b>71</b>
5.1	Summary . . . . .	71
5.2	Conclusion . . . . .	72
5.3	Recommendations . . . . .	73
	<b>Bibliography . . . . .</b>	<b>75</b>

# List of Tables

4.1	Quality improvement by mesh modifying operation . . . . .	49
4.2	Quality improvement by mesh modifying operation . . . . .	50
4.3	Quality improvement by mesh modifying operation . . . . .	51
4.4	Comparison of drag and lift for NACA 0012 airfoil: $Ma = 0.5$ , $Re = 5000$ , $\alpha = 0^\circ$ . . . . .	59
4.5	Comparison of drag and lift for NACA 0012 airfoil: $Ma = 0.8$ , $\alpha = 1.25^\circ$ . .	66

# List of Figures

1.1	Topology of (a) structured and (b) unstructured mesh. . . . .	4
1.2	Converting unstructured mesh to structured mesh . . . . .	6
1.3	Isotropic mesh around NACA-0012, Figure courtesy of Serge Gosselin's PhD thesis [25] . . . . .	7
1.4	Comparison between (a) Isotropic and (b) Anisotropic mesh for a rectangle . . . . .	8
2.1	The deformation tensors $F_p$ and $F_q$ . . . . .	18
2.2	Distance and angle measurements from a certain point . . . . .	19
2.3	Metric-based refinement. Figure courtesy of Douglas Pagnutti's Master thesis [54] . . . . .	22
2.4	The Jacobian matrix for the element being studied . . . . .	25
2.5	The Target matrix for the desired element of equilateral triangle . . . . .	26
2.6	Anisotropic target element . . . . .	26
3.1	Weighted Jacobian matrix definition . . . . .	30
3.2	Delaunay triangulation of a set of random points . . . . .	31
3.3	Face swapping . . . . .	33
3.4	Vertex insertion procedure . . . . .	35
3.5	Point insertion on a boundary face . . . . .	36
3.6	Comparison between a diametral circle (dashed) and diametral lenses . . . . .	37
3.7	Vertex removal . . . . .	38
3.8	Vertex removal for four triangles . . . . .	38
3.9	Vertex removal for a boundary point . . . . .	39
3.10	Correct order and orientation for smoothing . . . . .	42
3.11	Decision on target element . . . . .	44
3.12	General algorithm . . . . .	46
4.1	The effect of mesh modifying operations . . . . .	48

4.2	The effect of vertex movement . . . . .	50
4.3	Anisotropic quasi-structured mesh on a curved boundary . . . . .	51
4.4	Expected triangle aspect ratio . . . . .	52
4.5	A mesh on a $1 \times 1$ square being recursively refined . . . . .	53
4.6	Closeup view of the mesh in Fig. 4.5d . . . . .	54
4.7	Quality improvement for the mesh on a $1 \times 1$ square being recursively refined . . . . .	55
4.8	Initial mesh with 2543 vertices around the NACA 0012 airfoil . . . . .	56
4.9	Meshes and solutions for subsonic viscous flow over NACA 0012, $Ma = 0.5$ , $Re = 5000$ , $\alpha = 0^\circ$ . . . . .	58
4.10	Comparison between smoothed and non-smoothed mesh, viscous subsonic flow around NACA 0012 . . . . .	60
4.11	Drag convergence for NACA 0012, $Ma = 0.5$ , $Re = 5000$ , $\alpha = 0^\circ$ . . . . .	61
4.12	Lift convergence for NACA 0012, $Ma = 0.5$ , $Re = 5000$ , $\alpha = 0^\circ$ . . . . .	61
4.13	Selected adapted mesh for the subsonic laminar problem. Figure courtesy of Yano and Darmofal [41]. . . . .	62
4.14	Pressure coefficient distribution for NACA 0012, $Ma = 0.5$ , $Re = 5000$ , $\alpha = 0^\circ$ . . . . .	62
4.15	Wall shear stress distribution for NACA 0012, $Ma = 0.5$ , $Re = 5000$ , $\alpha = 0^\circ$ . . . . .	63
4.16	Meshes and solutions for transonic inviscid flow over NACA 0012, $Ma = 0.8$ , $\alpha = 1.25^\circ$ . . . . .	65
4.17	Comparison between smoothed and non-smoothed mesh, inviscid transonic flow around NACA 0012 . . . . .	67
4.18	Pressure coefficient distribution for NACA 0012, $Ma = 0.8$ , $\alpha = 1.25^\circ$ . . . . .	68
4.19	Mach number distribution for NACA 0012, $Ma = 0.8$ , $\alpha = 1.25^\circ$ . . . . .	68
4.20	Drag convergence for NACA 0012, $Ma = 0.8$ , $\alpha = 1.25^\circ$ . . . . .	69
4.21	Lift convergence for NACA 0012, $Ma = 0.8$ , $\alpha = 1.25^\circ$ . . . . .	69
4.22	Selected adapted mesh for the transonic inviscid problem. Figure courtesy of Yano and Darmofal [41]. . . . .	70

# Nomenclature

## Roman Symbols

$\vec{V}$	velocity vector
$A$	Jacobian matrix
$A_p$	adaptation parameter
$B_l$	shortest edge to area ratio of the ideal triangle
$B_R$	circumradius to area ratio of the ideal triangle
$C_f$	Skin friction factor
$C_P$	Pressure coefficient
$D$	determinant of a matrix
$d$	distance
$F$	deformation tensor, objective function
$f$	continuous function, given vector
$H$	Hessian matrix
$h$	mesh length scale
$I$	identity matrix
$i$	control volume index
$l_{min}$	shortest edge of a triangle
$M$	metric

$Ma$	Mach number
$N$	total number of elements
$p$	polynomial order
$R$	rotation matrix, circumradius of an element
$s$	positive scalar
$T$	weighted Jacobian matrix
$u$	perturbation in the flow field
$V$	eigenvector
$v$	adjoint solution
$W$	target matrix
$x, y$	Cartesian coordinates
$e$	error indicator
$Q$	suitable flow property

### **Greek Symbols**

$\Lambda$	size matrix
$\lambda$	eigenvalue
$\mu$	local quality function
$\tau$	tolerance
$\Xi$	sample element

### **Subscripts**

$F$	Frobenius matrix norm
$k$	element index
$p, q$	point index

# Acknowledgments

First I would like to thank my supervisor, Dr. Carl Ollivier-Gooch for his constant availability for discussion and for his invaluable help throughout the course of this research.

I would also like to thank my lovely family, the kindest person in the world, my mother, “Mahtab”, the most reliable person in the world, my father, “Amir”, and my dear brother “Kamran” for their endless encouragement and love.

Finally, my deepest appreciation is for my husband, “Alireza”, for his limitless patience, support, encouragement and love in my life.

# Chapter 1

## Introduction

Numerical methods are now commonly used as an important analysis tool in the solution of scientific and engineering problems. While the efficiency of such methods has been dramatically improved, there is still a strong industrial desire to produce more accurate numerical solutions. The equations governing the fluid behavior are a set of PDEs which are the focus of numerical analysis [54]. As a result of their complexity, there is not any general analytical solution for these equations. One classic example is the Navier-Stokes equations which describe the motion of viscous fluids. Although exact solutions are available for simplified flows, such as Poiseuille flows [28], the existence of a solution to these equations in the general case remains to be proved. On the other hand, since the solution has a multi-scale nature, experimental analysis is also costly. Considering these limitations, numerical solution is one of the best choices for problems involving fluid flows. Computational Fluid Dynamics (CFD) has shown remarkable capability for fluid analysis and also produces accurate and efficient (both in terms of memory usage and time) results. Nowadays, CFD is used extensively and successfully in industry throughout the design process, from preliminary design to shape optimization. Increasing computing power over the last two decades has resulted in the development of new computational techniques and algorithms, enhancing the versatility of CFD applications [45].

Modeling, mesh generation and solution are three essential components involved in numerical simulations [25]. Modeling includes the governing equations, the problem geometry and the boundary conditions. To solve the PDEs numerically, the continuous domain being studied should be tessellated into shapes recognized by the solver. This process, referred to as mesh generation, is one of the most time consuming parts of CFD solution in human time. These equations are then discretized over the domain and a system of algebraic equations which should be solved is obtained. Finally the system of algebraic equations is solved and the numerical solution for the flow of interest is obtained.

Since the discretization of the fluid flow equations is carried out on the mesh, the CFD solution may be significantly inaccurate without using a properly discretized domain [45].



Furthermore when dealing with complex geometries, generating an appropriate mesh is the most time consuming part of the solution process for a CFD user. According to a real aerodynamic case study in aerospace engineering, the mesh preparation time can be up to 45 times larger than the required computation time for the fluid flow simulation [21]. Hence, it is necessary to reduce the mesh generation time and there is a large potential to gain by automating this process. This thesis deals with the second step of this procedure, mesh generation.

## 1.1 Motivation

Numerical solution of PDEs is an important analysis tool for scientists and engineers. Ideally, numerical simulations should require only specification of the physics, geometry and boundary conditions of the problem and a solution with the desired accuracy should be obtained [54]. The construction of a grid which minimizes the number of nodes required to compute a flow field within a prescribed tolerance level, or even of a grid which approximates the geometry of the domain with sufficient accuracy to allow the computation of an initial coarse solution in an adaptive solution process, remains in fact a difficult and time consuming task with the commonly available software [58]. Among the many tasks which require substantial user input in grid generation, prescribing the mesh spacing for a complex domain is probably the most demanding in terms of human resources required.

The accuracy of the final solution is highly dependent on the mesh spacing; accordingly one approach for achieving accurate solution is to repeat the problem with increasingly finer meshes until an acceptable variation, i.e. within the tolerance, from one solution to another is obtained [54]. This approach is inefficient and time consuming. Assessing and minimizing discretization error by this method requires time consuming mesh dependence analysis or computation of solution on meshes that are finer than the required solution accuracy dictates. Furthermore, a local difference in element shape, orientation and size has a dramatic influence on the solution. Hence, it is only asymptotically guaranteed to obtain a more accurate solution by dividing the domain more and thus having finer mesh.

One method employed to overcome this problem is mesh adaptation, in which the mesh is locally fit to the particular features of the flow. Mesh adaptation seeks to automate the process of minimizing discretization error by first computing the solution on a coarse mesh and then successively refining the mesh so that the optimal mesh for the solution being computed is produced. A mesh generator should aim at achieving the best possible

mesh using as few mesh points as possible since the computational costs (both in terms of memory usage and time) are directly dependent on the number of elements in the mesh. It is useful to make the mesh finer in certain regions where the solution details must be captured, for example near boundaries, while keeping it coarse everywhere else (such as far fields). One of the most important properties of a proper mesh for numerical simulations is that the density of mesh points should be easily controllable and allowed to vary quickly over a short distance [25]. This property which is easily available in adaptive schemes saves simulation time since it prevents time from being wasted in regions that would otherwise be over-resolved.

## 1.2 Mesh Classification

The finite volume and finite element methods, which are two of the most commonly used approaches for spatial discretization of PDEs, are applied to a computational domain partitioned into simple entities. This spatial decomposition is called a mesh. Meshes are categorized as structured and unstructured according to their connectivity and are divided into isotropic and anisotropic meshes based on the aspect ratio or shape of elements. In the following section, structured and unstructured meshes are discussed in detail, then another category of meshes called quasi-structured meshes are introduced in Section 1.2.2. Features of isotropic and anisotropic meshes are described in Section 1.2.3.

### 1.2.1 Structured and Unstructured Meshes

The most common types of grids used in CFD applications are categorized according to their connectivity as structured and unstructured meshes. In a structured mesh, all cells and vertices have the same fixed and predictable topology. As a result each cell and vertex can be identified by indices. As an example, in a structured mesh (Fig.1.1a), cell  $(i, j)$  is always topologically between cell  $(i + 1, j)$  and cell  $(i - 1, j)$  where are on the topological right and left of it respectively. Structured meshes are generally composed of quadrilaterals in two dimensions and hexahedra in three dimensions.

Unstructured meshes on the other hand have elements with irregular and variable topologies in the sense that the connectivity is not predictable and must be explicitly declared. For an unstructured mesh (Fig.1.1b), there is not particular topology which indicates the neighbors of control volume 1, for instance. The mesh adjacency data must therefore be stored to know the neighbors of a given vertex [25]. Unstructured meshes are

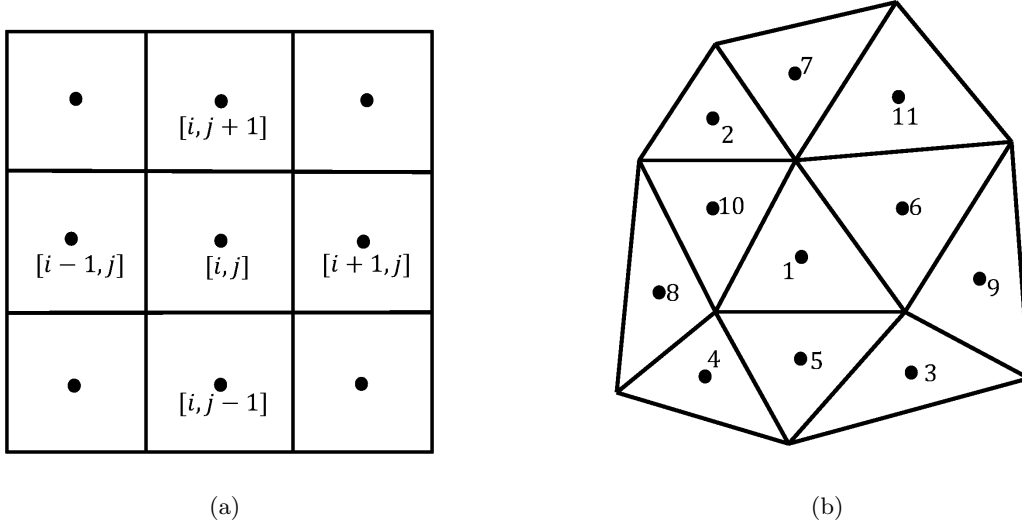


Figure 1.1: Topology of (a) structured and (b) unstructured mesh.

most commonly made of triangles or quadrilaterals in two dimensions and of tetrahedra, pyramids, and prisms in three dimensions. An unstructured mesh may also contain different types of entities in which case it is referred to as a mixed mesh.

From the numerical point of view, structured meshes with reduced memory usage as a result of their predictable topology are attractive. For more clarification, flux computation in finite volume method is considered. A fixed template is sufficient for flux computation in structured meshes; accordingly, faster solution with reduced memory usage can be developed capitalizing on their predictable topology [25]. However, the task of generating them around complex configurations has proved to be a considerable challenge. Obtaining a structured mesh of a complex domain often involves time-consuming human interventions that can easily offset the saving realized by the solver itself. Furthermore, generalizing the grid generation and adaptation in this approach is still not an easy task and needs significant effort.

On the other hand, for unstructured meshes evaluating the same flux as discussed above for finite volume method requires performing a polynomial reconstruction of the solution over each control volume. In spite of the higher cost for the solver, unstructured meshes are more applicable when arbitrary complex geometries are involved because of their automation capability and also their flexibility in refinement based on the geometry and

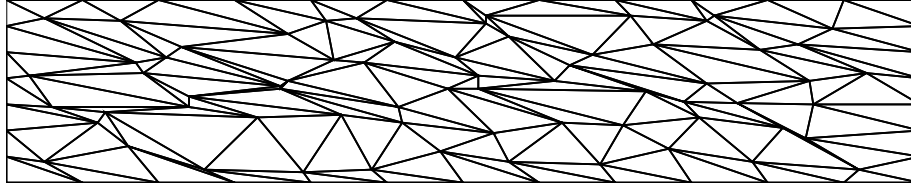
adaptation based on the solution features and gradients. Accordingly, the time and effort required by the user to produce an acceptable mesh reduces significantly [21]. Therefore, unstructured meshes are one of the most suitable choices for complex geometries; associated solvers are becoming more common in modern CFD applications and promise to be more capable and successful for complex aerodynamic problems.

### 1.2.2 Quasi-structured Meshes

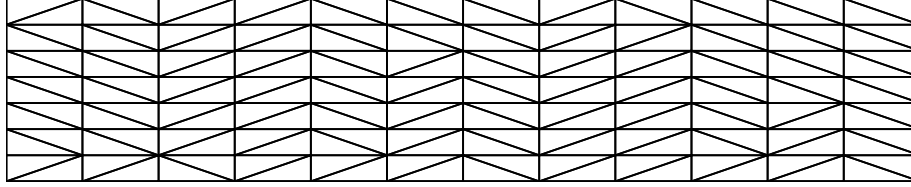
Many numerical examples have proved that the performance of a numerical scheme is highly dependent on the features of the discretization. Some features like boundary layers are easier to capture numerically by using structured meshes, so most unstructured codes strive to use quasi-structured meshes. In the vicinity of bodies, another type of meshes called quasi-structured grid is employed to capture the boundary layer in viscous simulations [4, 40, 61]. Fig. 1.2 shows an unstructured mesh (a) converted to a quasi-structured one (b) in the sense that by removing diagonals of the quadrilaterals formed by two neighboring triangles, a structured mesh (c) would be obtained. Therefore, the quasi-structured meshes can be defined as triangular meshes that are similar to structured quad meshes with diagonals added which means the topology is not predictable and consequently they are treated as unstructured meshes. The quasi-structured method takes advantages of the positive properties of structured grid elements in the regions which need them the most and uses automated unstructured grid techniques where not much is happening in the flow field. This combination of grid types not only allows the benefits of structured and unstructured grids to be attained simultaneously, but also allows high grid quality to be achieved throughout the domain due to the appropriate use of each element type [71]. The ability to control the shape and distribution of the grid locally is a powerful tool which can yield excellent meshes.

### 1.2.3 Isotropic and Anisotropic meshes

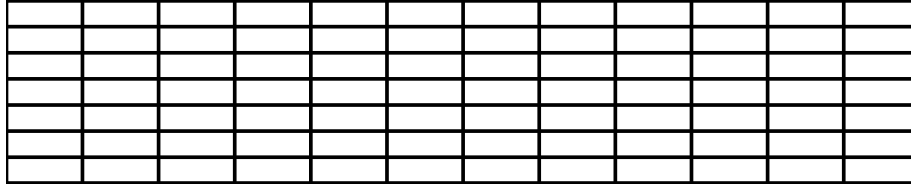
As mentioned earlier, the size and the shape of cells in the grid may have a dramatic influence on solution accuracy. When the physical behavior of the problem is isotropic, using an isotropic mesh (ideally, equilateral) whose cell size is such that the error per cell is equidistributed is generally preferable [54]. Therefore, the majority of mesh generation software tends to focus on creating cells of varying length scales but with unitary aspect ratios. However, many physical phenomena relevant to engineering problems have highly anisotropic solutions. The boundary layer in high Reynolds viscous fluid flows is a good



(a) Unstructured



(b) Quasi-structured



(c) structured

Figure 1.2: Converting unstructured mesh to structured mesh

example of this phenomenon. In a boundary layer strong velocity gradients develop in a direction perpendicular to the flow in comparison with velocity gradients parallel to the flow. Transonic and supersonic flows in which discontinuities or shock waves may appear are another example of these anisotropic phenomena: numerical gradients normal to shock waves are much larger than numerical gradients parallel to them. To capture these phenomena accurately, one could use a large number of very small isotropic triangles clustered at the boundary, as is done in the mesh depicted in Fig. 1.3, or near shock waves. The strong gradients in the perpendicular direction are accurately captured through such an approach but the simulation is inefficient as a consequence of over-resolved flow in the tangential direction [25]. The isotropic mesh around a NACA-0012 airfoil which is depicted in Fig. 1.3 has a very high density close to the airfoil. The size of mesh elements normal and tangential to the boundary layer are the same, which means the computation costs too much unnecessarily.

A better approach is to use anisotropic meshes aligned with the flow which means that the mesh contains long and thin elements that have been stretched according to a preferred

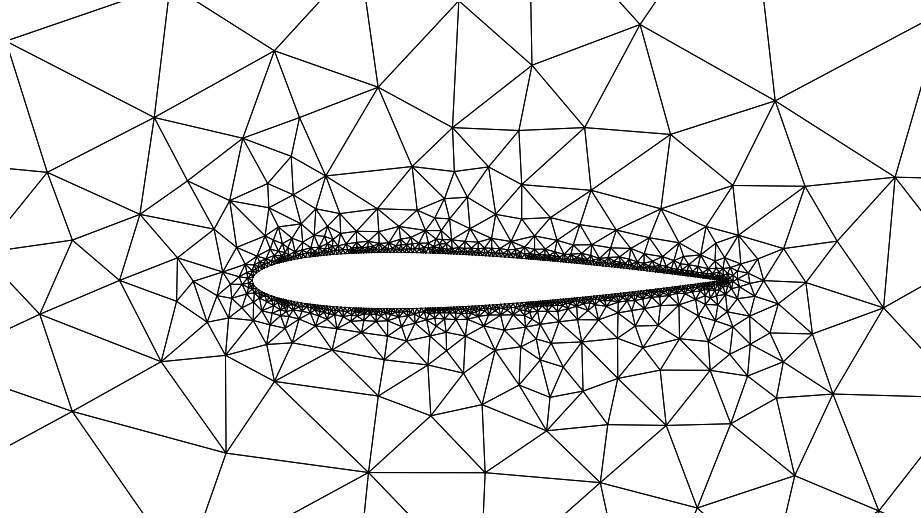


Figure 1.3: Isotropic mesh around NACA-0012, Figure courtesy of Serge Gosselin's PhD thesis [25]

direction i.e. parallel to the boundary layers or shock waves. Rippa [60] has found that for linear interpolation of smooth functions, elements should be long in directions where the magnitude of the second directional derivatives is small and thin where the magnitude of the second directional derivatives are large. Fig. 1.4 compares isotropic and anisotropic meshes for the same geometry. Elements are nearly equilateral triangles for the isotropic mesh and they are stretched in one direction (in this example in the x-direction) in an anisotropic mesh.

The anisotropic grid generation approach has been applied with success to the computation of inviscid and viscous flows around complex aeronautical configurations [3, 58, 65, 67]. In addition to providing unstructured grids of good quality formed by a relatively small number of elements, some desirable properties of the grid are also easily controllable. For instance, it is possible for aerodynamics problem to maintain the chord wise number of nodes along the wing span almost constant, which is difficult by using an isotropic unstructured mesh generation process. Accordingly, for a given number of mesh points, better resolution of boundary layers and shock waves are obtained by using anisotropic meshes since the mesh anisotropy matches the solution anisotropy. Anisotropic adaptation uses a feedback loop in a CFD algorithm in which the solution on an initial mesh guides the creation of an anisotropic mesh that fits the physical behavior of the problem. Since the number of mesh points dramatically decreases by using an anisotropic mesh, there is a

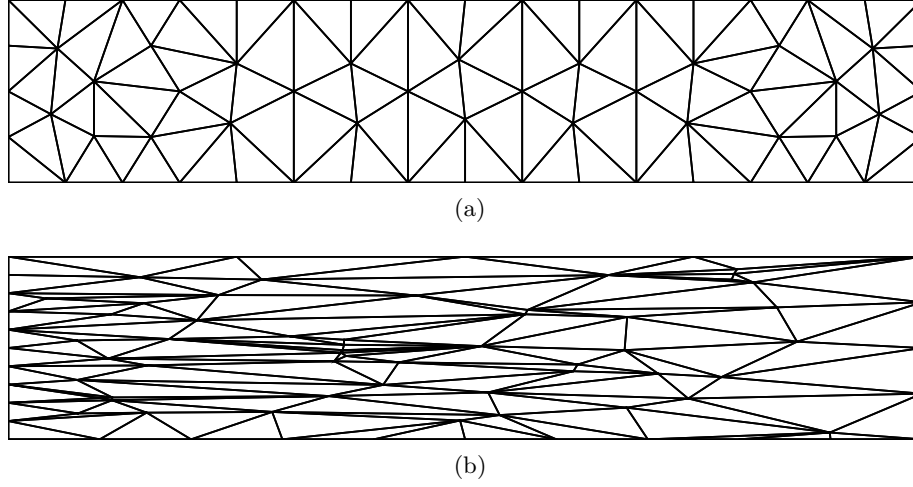


Figure 1.4: Comparison between (a) Isotropic and (b) Anisotropic mesh for a rectangle

significant reduction of computational cost for achieving the same accuracy. Another advantage of using an anisotropic mesh is that the dependence of final solution on the mesh significantly decreases for a given number of mesh points which means that there is more confidence in the results [54]. Considering the fact that mesh generation is often the most time-consuming part of CFD, by automating this process, the usability of CFD programs substantially increases.

### 1.3 Objective and Outline

Current anisotropic mesh adaptation schemes produce meshes without any regular structure which hurts both the accuracy and the efficiency of the flow solution. The best possible approach is to use mixed meshes relying on stretched quasi-structured elements to capture components of the flow with anisotropic physical behavior such as shocks and shear layers and unstructured isotropic triangles for the rest of the domain.

Structured meshing provides high quality meshes at a relatively low cost and as a result of inherent directional qualities also provides accurate and efficient results. However, they are restrictive when applied to a complex geometry. In contrast, the unstructured meshes have flexibility for complex shapes; however, the computational costs are relatively high.

Quasi-structured meshes combine both the computational efficiency of the structured meshes and the geometric flexibility of unstructured meshing. Since the advantages of one approach are the disadvantages of the other approach, the combination of them capitalizes

on the merits of both approaches.

The goal of my research is to use anisotropic smoothing to enhance an existing anisotropic adaptation scheme to produce quasi-structured meshes where they are needed the most. A mesh adaptation scheme capable of producing highly anisotropic meshes was developed by Pagnutti [54] and in current work, this method is modified to generate aligned, quasi-structured meshes in particular parts of the problem domain with highly anisotropic physical behavior such as boundary layers in viscous flows and discontinuities in transonic flows. By applying this method and producing quasi-structured meshes, the accuracy of the solution should be increased as a result of meshes with regular structure and those regions of the domain with anisotropic elements have ultimately regular structure which match the structure of the flow properties, i.e. boundary layers in viscous flows and discontinuities in transonic flows.

An overview of the advantages of anisotropic mesh adaptation techniques based on previous research in the field is provided in Chapter 2. The most common way of defining the anisotropy is through a metric which is discussed in this chapter, along with techniques for estimating error and converting it to the form of metric. After adaptation, the constructed anisotropic mesh should be modified to have higher quality. One of the methods applied to improve the quality of the mesh is node movements. This means that a mesh optimization process is required. Mesquite[23] is used as the mesh optimization package. The main idea of Mesquite and its algorithm are introduced in the last part of Chapter 2.

Chapter 3 is devoted to the construction of a high quality anisotropic quasi-structured mesh from an arbitrary initial mesh. After defining how to measure the mesh quality quantitatively, different operations to improve the quality of the mesh are introduced. These operations include point insertion for large elements, vertex removal for short edges, face swapping and node movement. As mentioned, the last operation is done by Mesquite and the other three operations are implemented by the GRUMMP meshing library [48]. The mesh topology operations should be modified for anisotropic meshing and the vertex movement operation should be modified to smooth vertices such that quasi-structured meshes, which are aligned, are generated. These modifications are described and this chapter ends with introducing the general algorithm applied for quasi-structured mesh generation.

In Chapter 4, it will be shown that the algorithm produces meshes well suited to the desired metric. Preliminary tests are demonstrated at the beginning of this chapter to show how each of those four techniques may improve the quality of the mesh. A fake boundary layer and shock is simulated in a domain of square shape as a verification case which



proves that the algorithm produces quasi-structured anisotropic meshes in the regions with the anisotropic physical behavior. For more practical examples, two different flows over a simple airfoil are examined to show the benefits of the applied method. Subsonic viscous flow and transonic inviscid flows which are two different flow regimes with different physical behavior are tested to show that the adaptation/smoothing procedure improves solution accuracy.

The thesis concludes in Chapter 5 with a summary of the research, describing the contributions, discussion of the results and providing conclusion based on them and finally recommending some future work.

## Chapter 2

# Background

This chapter provides an overview of the concepts on which the anisotropic smoothing algorithm is based. Anisotropy definition is based on error estimation methods which is discussed in Section 2.1. This overview is complemented by the discussion of previous work related to anisotropic mesh adaptation techniques based on error estimation methods.

Several methods developed for anisotropic mesh generation including metric-based mesh adaptation and Centroidal Voronoi Tessellation . The former which is used in this research is introduced in Section 2.2 and the later which can be an alternative approach is introduced briefly in Section 2.3.

To construct quality anisotropic meshes, mesh optimization is also needed. Mesh optimization via node movement using the Target-Matrix Optimization Paradigm is described in Section. 2.4. The optimization algorithm presented here seeks to find the set of node positions such that elements in the mesh are as close as possible to the target element dictated by the desired metric.

### 2.1 Error Estimation Methods

As mentioned earlier, modeling, mesh generation and solution are three essential components involved in numerical simulations. The final step is performing the actual approximation. The governing equations are typically discretized using one of three common approaches: finite difference, finite element or finite volume. Each family of these schemes converts the PDE to an algebraic system.

Using the finite difference method, the governing equations are discretized by replacing each differential operator of the PDE with an equivalent difference operator. For example, if

$$\mathcal{L}(f) = \frac{\partial f}{\partial x} = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \quad (2.1)$$

then an appropriate difference operator may be defined as

$$\tilde{\mathcal{L}}(f) = \frac{f(x+h) - f(x)}{h} \quad (2.2)$$

where  $h$  is now a finite value. By using the Taylor expansion of the solution, if we assume that  $f(x)$  is known at a series of discrete locations  $x_i = ih$ , then the solution at  $f(x)$  can be approximated using the difference formulation of the original PDE.

In the finite-element discretization, it is assumed that the solution is known at a discrete set of locations in the mesh similar to the finite-difference method. A polynomial of a certain degree is assigned to the solution between those locations. Since this polynomial does not necessarily satisfy the equation  $\mathcal{L}(f) = 0$ , some error called residual error are appeared. By applying a set of weight functions to the residual error, it is possible to choose polynomial coefficients minimizing the residual.

In the finite-volume approach, instead of trying to calculate quantities at specific locations, the average of those quantities is calculated by summing fluxes at the boundaries and then applying them to the conservation form of the PDEs. Each cell in the mesh is considered as a control volume<sup>1</sup> and the average of the solution within each cell is assumed to be known. Once the flux through each cell boundary is approximated, the cell average is updated. The advantage of using this method is that it is easily formulated for unstructured meshes and thus is used in many CFD packages. This method is used throughout this research.

Regardless of the discretization method used in the CFD simulation, the general approach for solving the governing equations over the domain is to assume that the solution is known exactly at a set of locations. Those known values are used to approximate the solution at other locations by Taylor series. The following 2-D example illustrates this.

Assume that a solution  $f(x, y)$  has an infinite Taylor expansion in the neighborhood of some location  $(x_0, y_0)$ :

$$f_{exact}(x, y) = \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} \frac{1}{n!m!} \frac{\partial^n}{\partial x^n} \frac{\partial^m}{\partial y^m} f(x_0, y_0) (x - x_0)^n (y - y_0)^m \quad (2.3)$$

If this infinite sum is approximated using only terms up to degree  $p - 1$ , the  $p$  order solver

---

<sup>1</sup>Similar to the control volume approach used analytically for thermodynamics and fluid dynamics systems.

will be obtained.

$$f_{app}(x, y) \approx \sum_{n=0}^k \sum_{m=0}^l \frac{1}{n!m!} \frac{\partial^n}{\partial x^n} \frac{\partial^m}{\partial y^m} f(x_0, y_0) (x - x_0)^n (y - y_0)^m \quad (2.4)$$

where  $k + l = p - 1$ .

The difference between the actual solution  $f_{exact}$  and the approximate one  $f_{app}$  is the discretization error. Solution-based adaptation schemes aim to generate a mesh which equidistributes this error. Since it is not possible to know the exact error without knowing the exact solution, estimating the discretization error is an essential step in this process.

A solution-adaptive approach typically consist of three components: (1) the formulation of cell-modification parameters from local error estimates, (2) a meshing algorithm that uses the modification parameters to construct an improved mesh , and (3) a mesh-adaptation strategy that minimizes the cost of the simulation [46].

Three main approaches typically used to estimate the error are interpolation-based, feature-based and adjoint-based error estimation methods which are briefly discussed here.

### 2.1.1 Interpolation-based Error Estimation

The main idea in the interpolation-based method is that for a triangular element  $n$  with a given area, the error (in various norms) for the linear interpolation of a function  $f$  at the vertices of  $n$  is minimized when  $n$  is anisotropic with its long direction aligned with the eigenvector associated with the smaller eigenvalue of the Hessian  $\nabla^2 f$  [12]. In two dimensions, the Hessian matrix may be written as:

$$H = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix} \quad (2.5)$$

The desired aspect ratio of the triangle  $n$  in this case is the square root of the ratio of the larger eigenvalue of  $\nabla^2 f$  to the smaller one. The globally optimal mesh can be characterized by the equidistribution of the interpolation error over each element [37].

If the Hessian matrix is positive-definite and invertible, then an ideal cell can be found by linearly transforming an isotropic cell by  $H^{-1}$ . This is the approach used for flow simulations by Castro-Diaz *et al.* [13] and for developing a solver-independent and user-independent mesh adaptation scheme by Habashi *et al.* [17].

Even for a perfectly equidistributed mesh in terms of interpolation-error estimation,

$p$  order accuracy would rarely be achieved as a result of the presence of convected error presence. Small errors from one location of the mesh can be convected to regions which are relatively far away since the error in values used in the Taylor approximation are passed on to the interpolated value which will affect other Taylor expansion terms and so on [54]. Those PDEs containing strong convective terms, such as the Euler equations, suffer from this drawback of the interpolation-based error estimation.

### 2.1.2 Feature-based Error Estimation

Feature-based grid adaptation uses solution gradient, solution curvature, or even identified solution features to drive the adaptation process. This approach relies on *a priori* knowledge of the flow, including whether the solution contains shock waves, boundary layers, etc., as well as a reliable feature-detection system.

The adaptation parameter ( $A_p^2$ ) can be defined by Eq. 2.6 in which each of the error indicators can isolate a particular flow feature [6].

$$A_p^2 = \{e_1, e_2, e_3\} \quad (2.6)$$

where  $e_1, e_2, e_3$  are the error indicators given by

$$e_1 = \max \left[ -\frac{\vec{V} \cdot \nabla Q}{|\vec{V}|}, 0 \right] \quad (2.7)$$

$$e_2 = \max \left[ +\frac{\vec{V} \cdot \nabla Q}{|\vec{V}|}, 0 \right] \quad (2.8)$$

$$e_3 = \left| \nabla Q - \frac{\vec{V}}{|\vec{V}|} \left( \frac{\vec{V} \cdot \nabla Q}{|\vec{V}|} \right) \right| \quad (2.9)$$

$Q$  is any suitable flow property and  $\vec{V}$  denotes the velocity vector. The first two error indicators, Eq. 2.7 and 2.8, captures expansions and compressions in the flow direction and Eq. 2.9 rcaptures gradients normal to the flow direction. For viscous flows, for instance, as a result of no-slip boundary condition ( $\vec{V} = 0$ ) in the boundary layer, the adaptation parameter ( $A_p^2$ ) reduces to the magnitude of the gradient of  $Q$ .

The main advantage of explicitly detecting features is that more specific refinement is possible. The feature-based adaptation refines shock waves, boundary layers and wake. On the other hand, feature-based adaptation may result in over-refined features while some other features are not refined enough [63]. In some cases, feature-based refinement can actually increase the solution error. Dwight [19] has shown the failure of feature-based adaptation for the inviscid transonic flow over an airfoil using an unstructured finite-volume discretization. Adaptation based on solution gradients initially shows a reduction in the discretization error, but then subsequent adaptation steps show an increase in the discretization error. Another disadvantage of using this method is that features may be detected in the wrong location as a consequence of convected errors as described for interpolation-based method.

### 2.1.3 Adjoint-based Error Estimation

Adjoint-based error estimation involves solving both the primal PDEs and the adjoint PDEs. Suppose that we wish to compute by postprocessing  $J = g^T u$  where  $u$  satisfies the linear system of equations  $Au = f$  arising from discretizing the PDE. The adjoint solution  $v$  satisfies the linear system of equations  $A^T v = g$  [24]. Therefore,

$$g^T u = \left(A^T v\right)^T u = v^T Au = v^T f \quad (2.10)$$

The term  $g^T u \equiv v^T f$  can be computed either by the direct approach, solving  $Au = f$ , or by the adjoint approach, solving  $A^T v = g$ . For a single design variable there would be no benefit in using the adjoint approach, but for multiple design variables, each has a different  $f$ , but the same  $g$ , so the adjoint approach is computationally much more efficient.

The error can be expressed as an inner product of the local residual errors and adjoint variables. So the error in a chosen functional can be directly related to the local residual errors of the primal solution through the adjoint variable.

By estimating the local contribution of each cell or element to the error in any solution functionals of interest (e.g., lift, drag, and moments) [63], this approach localizes mesh refinement to those parts of the computational domain that most influence the accuracy of the selected outputs [46]. Becker and Rannacher [8] have developed this output-based adaptive procedure by exploiting finite element orthogonality properties. This approach has been extended to the finite volume method; for example, Venditti and Darmofal [70] successfully applied this approach for inviscid and viscous flow over airfoils at various Mach

numbers.

For situations where the only desired output is a function of the solution, adjoint-based methods are ideal because they do not refine areas of the mesh where the solution has no effect on that function [54]. Analyzing the drag force on an airfoil is an example. The mesh created by adjoint-based refinement provides more accurate drag prediction than other meshes of similar size. However, using the same mesh to predict lift may not provide sufficient accuracy. An additional drawback for adjoint methods is their complexity and code intrusiveness.

## 2.2 Metric-based Anisotropic Mesh Adaptation

As mentioned earlier, anisotropic meshes can greatly reduce the number of mesh elements and therefore the computation cost (both in terms of time and memory) required for simulations. For many applications, anisotropic elements offer the best trade off between accuracy and speed. In fluid flow modeling, aspect ratios of 100,000 to one are sometimes appropriate [67]. Apel [3], Shewchuk [67] and Schoen [65] provide in-depth looks at the benefits of using anisotropic meshes. Anisotropic meshes are appropriate for simulation problems whose solutions have different behaviors in different directions.

Bossen and Heckbert [10] described a generalization of the isotropic mesh generation algorithm for anisotropic mesh generation. When large edge lengths near small input features were requested, the mesh generated by this algorithm could not make a guarantee on their quality.

Loseille and Lohner [39] introduced a robust local remeshing strategy which allows adaptation of different components of the flows by coupling anisotropic surface and volume remeshing and quasi-structured boundary layer mesh optimization/ generation. A local remeshing approach is utilized to generate anisotropic meshes around a wing for both supersonic and transonic flow.

Selmin *et al.* [58] compared simulation results for the turbulent flow around a wing-body configuration to experimental results. This Navier-Stokes simulation used a fully anisotropic unstructured/hybrid grid with a relatively small number of nodes. The anisotropic nature of the grid elements allows reduction the number of nodes by nearly one order of magnitude compared with the isotropic case. Comparison of numerical and experimental results reveals that they are in a good agreement.

Pagnutti [56] developed a general framework for an isotropically refining unstructured

meshes. Through several test cases including subsonic and transonic inviscid flow, the performance of this refinement procedure was compared with uniform mesh refinement. Uniformly refined meshes did not capture shock waves accurately for transonic flow around an airfoil, while on anisotropic grids the shock waves are extremely sharp and have very little overshoot. To better demonstrate the advantages of anisotropic meshing, viscous subsonic flow around an airfoil was also examined in this article.

Habashi *et al.* [2, 26] used anisotropic mesh adaptation for both structured and unstructured meshes and this method was tested for two-dimensional, inviscid and viscous flows, with anisotropic physical behavior, over a wide range of speeds. To control the size and orientation of the adapted mesh, the interpolation-based error estimation is used. It was shown that anisotropic grids generated by a full-coupling of solver and adaptation is the most logical way to obtain appropriate grids and to increase the accuracy and reliability of CFD calculations. The reversibility of the algorithm was also shown to validate the adaptation algorithm by recovering a perfectly uniform mesh by specifying a uniform second derivatives starting from highly distorted meshes.

Legrand *et al.* [20] showed the potential advantages of anisotropic unstructured meshes for modeling the hydrodynamics of the Northwestern European continental shelf, the continental slope and the neighboring ocean within the same mesh. An efficient strategy was developed in which the mesh refinement is controlled through a few design parameters. The analysis of the quality of the generated meshes demonstrates that, while major topographic features are correctly resolved in both the along-slope and the cross-slope directions, anisotropic graded meshes can be obtained with good geometrical properties.

Once the desired anisotropy has been defined by one of the error-estimation methods defined in Section 2.1, that anisotropy should be communicated to the meshing program to generate an anisotropic mesh. For two-dimensional problems, anisotropy is often represented by a  $2 \times 2$  symmetric positive definite matrix<sup>2</sup> called a *metric* at each point of the domain. A geometric interpretation of the metric tensor at a point  $p$  is that it describes how distances and angles are measured from  $p$ 's point of view. As in the work of Labelle and Shewchuk [36], for a  $2 \times 2$  metric  $M_p$ , a deformation tensor  $F_p$  can be defined such that

$$M_p = F_p^T F_p \quad (2.11)$$

where  $F_p$  is an invertible  $2 \times 2$  matrix. Then  $F_p$  maps the physical domain to a warped space

---

<sup>2</sup>A square symmetric matrix is positive definite if and only if all of its eigenvalues are positive.



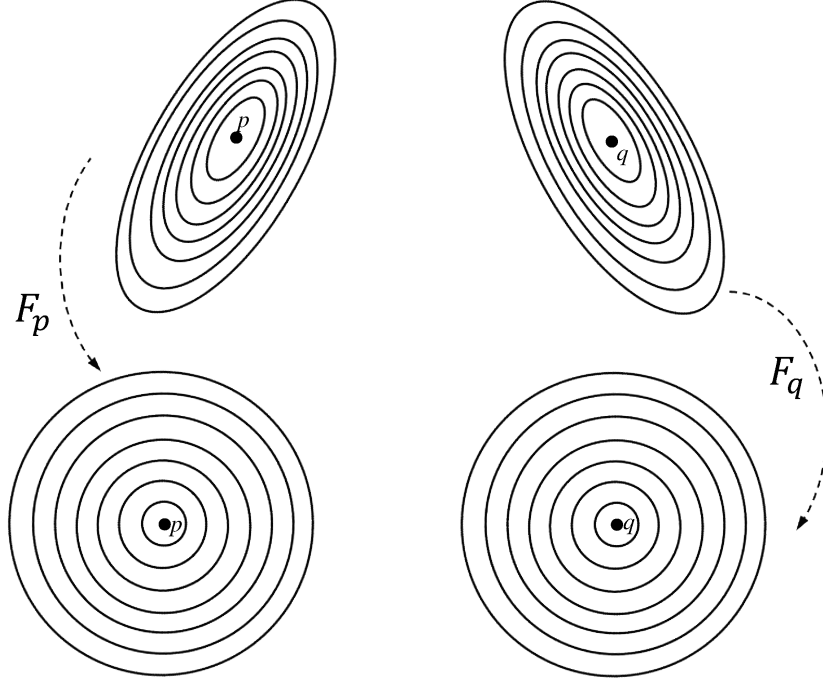


Figure 2.1: The deformation tensors  $F_p$  and  $F_q$

where distances and angles can be measured as they are seen by  $p$  simply by measuring them in the usual Euclidean way. In Fig. 2.1, ellipses around  $p$  represents isocontours which are sets of points that are equidistant from  $p$  when measured from  $p$ 's point of view. Likewise, the ellipses around  $q$  are  $q$ 's isocontours.

The ellipse around each point specifies the anisotropy by three components: a major radius  $r_1$ , a minor radius  $r_2$ , and an angle  $\theta$ . Hence, both orientation and position are defined by the metric. The metric is defined as a function of these three components as:

$$M = R\Lambda R^T = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} 1/r_1^2 & 0 \\ 0 & 1/r_2^2 \end{pmatrix} \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \quad (2.12)$$

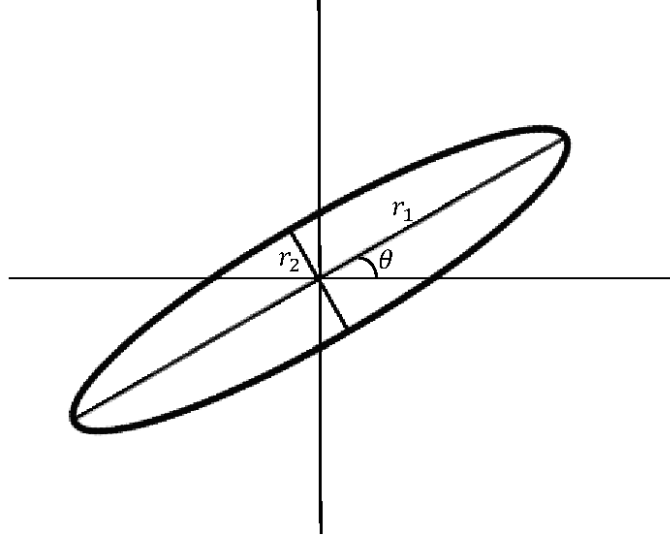


Figure 2.2: Distance and angle measurements from a certain point

A metric produces a distance measure that is positive for any two distinct points, regardless of the direction of travel between those two points, and satisfies the triangle inequality. Hence, if metric is a function  $d : X \times X \rightarrow \mathbb{R}$  for a vector space  $X$ , then the following conditions are satisfied.

$$d(p, q) = d(q, p) \quad (2.13)$$

$$d(p, q) > 0 \iff p \neq q \quad (2.14)$$

$$d(p, r) \leq d(p, q) + d(q, r) \quad (2.15)$$

For a uniform isotropic mesh, the metric is the standard distance measure  $\|p - q\|$ . If the distance is measured differently based on the orientation, anisotropy can be introduced. By linearly transforming the standard distance measure  $d_{std}(p, q) = \sqrt{(p - q)^T (p - q)}$  into an

anisotropic distance  $d_{\text{aniso}}(p, q)$ , the metric used in anisotropic meshing can be calculated as follows:

$$d_{\text{aniso}}(p, q) = \sqrt{(F(p - q))^T (F(p - q))} \quad (2.16)$$

$$d_{\text{aniso}}(p, q) = \sqrt{(p - q)^T (F^T F) (p - q)} \quad (2.17)$$

$$d_{\text{aniso}}(p, q) = \sqrt{(p - q)^T M (p - q)} \quad (2.18)$$

In which  $M$  is as defined in Eq. 2.11.

Geometric properties based on lengths have equivalent properties according to the metric function, so this function is ideal for creating meshes. The ideal anisotropic triangle has edge lengths that are equal according to the metric. Since any triangle is perfectly defined by the lengths of its edges, there is a mapped triangle in an anisotropic mesh from an isotropic triangle. This leads to an anisotropic control of the mesh spacing, which means control of not only the size of mesh elements but also their aspect ratio and orientation.

For the purposes of anisotropic mesh generation, the metric tensor field is assumed to be provided by those solution-driven adaptivity methods introduced in last section. The metric tensor field is often generated from the Hessian of a numerically computed solution. The idea of using the Hessian as a metric was popularized by results of D’Azevedo [15] and Simpson [15] on the linear interpolation error in a function and its gradient, respectively.

The principal difficulty in most problems is that different metrics must be used in different areas of the domain particularly in anisotropic meshing algorithms. While the metric function is valid at a specific point, the interpolated metric space between any two points might not be valid. By defining the metric discretely at vertices, it is possible to overcome this problem. The size of any triangle is measured according to the average metric of the three vertices of that triangle. When a new vertex is inserted in a triangle, the metric at that vertex is assigned based on a linear interpolation from the three vertices of that triangle. Alternatively, when coarsening the mesh the length of each edge is estimated by calculating the average of lengths of that edge according to the two adjacent triangles. These two definitions give unique triangle quality and edge length measures which can then be used for mesh improvement operations.

Many metrics have been proposed in the literature to measure quality based on quantities such as element size, shape, and smoothness. Borouchaki *et al.* [9] have tested metric-based mesh generation on different test cases in  $\mathbb{R}^2$  domains. These application ex-

amples showed the capabilities of the method. One example which considers an anisotropic map in the domain of a rectangle is discussed here to show the robustness of the method. The metric map is defined by the diagonal matrix of order 2:

$$M(x, y) = \begin{pmatrix} h_1^2(x, y) & 0 \\ 0 & h_2^2(x, y) \end{pmatrix} \quad (2.19)$$

where  $h_1$  and  $h_2$  are functions of  $x$  and  $y$  respectively. After three iterations, both the size and aspect ratio of the elements match very well to the requirements dictated by the metric. A supersonic scram jet, transonic viscous flow around a NACA-0012 and viscous supersonic flow around a cylinder are other test cases in this paper show the capability of the method.

Pagnutti [54] used a metric-based meshing with the capability of creating anisotropic meshes. In one test case, the desired anisotropy is defined by a metric of

$$M = \begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix} \quad (2.20)$$

where  $a$  and  $b$  are as follows:

$$a = \begin{cases} 1 \times 10^4 & 0.475 < x < 0.525 \\ 1 & \text{Elsewhere} \end{cases} \quad (2.21)$$

$$b = \begin{cases} 1 \times 10^4 & 0 \leq y < 0.05 \\ 1 & \text{Elsewhere} \end{cases}$$

for a  $1 \times 1$  square domain. The metric is assigned first to a coarse mesh as shown in Fig. 2.3a and then the mesh is refined so that the metric-based area of triangles is approximately halved. The refined mesh, Fig. 2.3b, achieved the desired anisotropy since by scaling regions with different metrics, by a factor of 100 in the  $y$ -direction for  $0 \leq x < 0.05$  and a factor of 100 in the  $x$ -direction for  $0 \leq y < 0.05$ , an isotropic mesh is obtained.

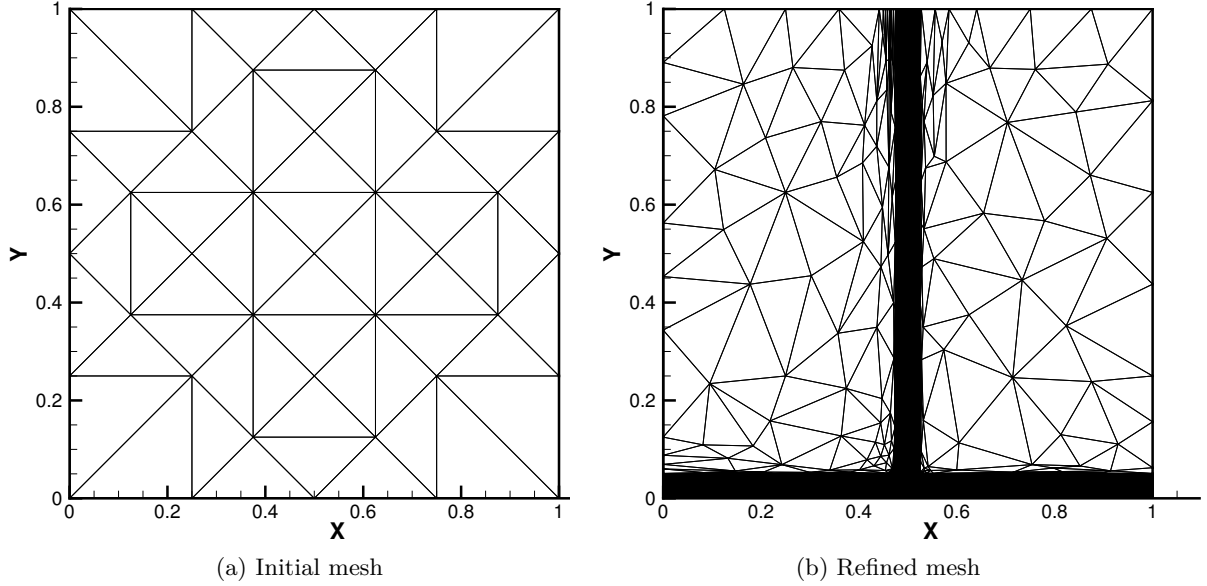


Figure 2.3: Metric-based refinement. Figure courtesy of Douglas Pagnutti’s Master thesis [54]

## 2.3 Centroidal Voronoi Tessellation

Several strategies exist for anisotropic meshing, such as tracing the curvature lines [53], or directly minimizing the approximation error [14]. Labelle and Shewchuk [36] propose a generalization of Voronoi diagrams with anisotropy attached to the vertices. Another method used is the centroidal Voronoi tessellation, CVT, which is a Voronoi tessellation whose generating points are the centroids (centers of mass) of the corresponding Voronoi regions.

CVT is defined as the minimizer of the objective function referred to as the CVT energy:

$$F(X) = F([x_1, x_2, \dots, x_n]) = \sum_i \int_{\Omega_i \cap \Omega} \|y - x_i\|^2 dy \quad (2.22)$$

where  $\Omega_i$  denotes the Voronoi cell of vertex  $x_i$ , and  $\Omega$  denotes the domain to be meshed [47]. To minimize this objective function, all the points  $x_i$  are moved at the centroid of their Voronoi cells. A new objective function  $F_{L_p}$  controlled by a given anisotropy field is proposed by Levy and Liu [5] which is the generalized CVT in a way that can take a directional-only anisotropy into account.  $L_p$  - CVT is defined as the minimizer of the

$L_p - CVT$  objective function  $F_{L_p}$ , obtained by injecting both the anisotropy term and the  $L_p$  norm into CVT energy:

$$F_{L_p}(X) = \sum_i \int_{\Omega_i \cap \Omega} \|M_y[y - x_i]\|_p^p dy \quad (2.23)$$

where  $\|\cdot\|_p$  denoted the  $L_p$  norm. The domain is first decomposed into a set of simplified cells on which the expression  $F_{L_p}$  is simple, represented by a list of integers and from this combinatorial representations, the value of  $F_{L_p}$  is obtained.

This method was applied for different test cases, including anisotropic surface meshing and automatic feature-sensitive remeshing for a variety of meshes with unconformities and proved that the fully automatic  $L_p - CVT$  method can generate meshes with smaller variations in angles and edge lengths, suitable for numerical simulations that rely on homogeneous sizes. An advantage of  $L_p - CVT$  is that boundaries are well respected regardless their orientation. However, it might not handle properly geometrically inconsistent models and misses triangles when a Voronoi edge passes through a gap, also more triangles may be missed when the number of vertices is increased.

The most important disadvantage of  $L_p - CVT$  is that under extreme anisotropy, which are the desired anisotropy in many practical aerodynamics problems for high speed and/or high viscosity flows, the convergence of this method becomes much slower because of the conditioning of the Hessian of  $F_{L_p}$ . The significant slower convergence of  $L_p - CVT$  is the main reason why we do not use this method in the current research.

## 2.4 Optimization Algorithm

Anisotropic grids can be obtained by the means of metric-based mesh generation and adaption processes, including mesh optimization based on local retriangulations. In the context of the latter approach, objective functions are used. The use of various possible objective functions and their effect on the resulting meshes are studied by Bottaso [11]. Objective functions are used for measuring the quality of each element in the grid with respect to a desired goal expressed by the metric defined in the last section, throughout the error estimation process defined in Section 2.1. Mesh quality refers to geometric properties of a mesh such as shape, size and orientation of its elements. The quality of the mesh is an important consideration since the accuracy and efficiency of the solution are dramatically

influenced if it is not properly controlled. In addition to solution accuracy, mesh quality can also affect the amount of time required to obtain the numerical solution.

A general-purpose algorithm called the Target-Matrix Optimization Paradigm (TMOP) is one of the methods used for mesh optimization via node movements (smoothing) with the aim of improving the quality of the mesh.

Other mesh optimization paradigms include Harmonic Maps [18], elliptic grid generators [69], and Laplace-Beltrami systems [38]. Many of the older paradigms were proposed prior to unstructured meshing, and therefore are somewhat out of date with respect to current meshes.

In contrast, TMOP addresses many mesh quality issues in a modern setting. TMOP formulates the mesh quality problem directly in terms of mesh entities, such as length of edges, area of elements, angles and so on; consequently the method is transparent to mesh type.

Mesquite<sup>3</sup>, which is a linkable software library, applies this algorithm to improve the quality of a given mesh. We call Mesquite directly from the general code of our anisotropic mesh adaptation and it does the smoothing part of the quality improvement. Mesquite is designed to be as efficient as possible, so that large meshes can be improved, and can deal with a wide variety of mesh types (e.g., structured, unstructured, 2D, 3D, hybrid and/or high-order nodes). It provides the capability of optimizing an existing mesh based on the metric defined before. The metric is used to define the desired elements in the optimal mesh.

### 2.4.1 Target Matrix Optimization Paradigm

To get the desired element in the optimal mesh, the Jacobian matrix should be defined first. The Jacobian  $A_k$  matrix for each element of the mesh is :

$$A_k = A(\Xi_k) \quad (2.24)$$

where  $\Xi_k$ ,  $k = 0, 1, \dots, K - 1$  denotes the sample element being studied within the active mesh [22]. The matrix is indirectly a function of the coordinates of the vertices of the element and varies from one element to the next. For a three-dimensional mesh, the Jacobian matrix is  $3 \times 3$  and for two-dimensional mesh, it is a  $2 \times 2$  matrix. Fig. 2.4 shows a triangular element with vertex coordinates  $x_0$ ,  $x_1$  and  $x_2$  and  $A$  is a matrix whose columns

---

<sup>3</sup>Mesquite Mesh Quality Improvement Toolkit

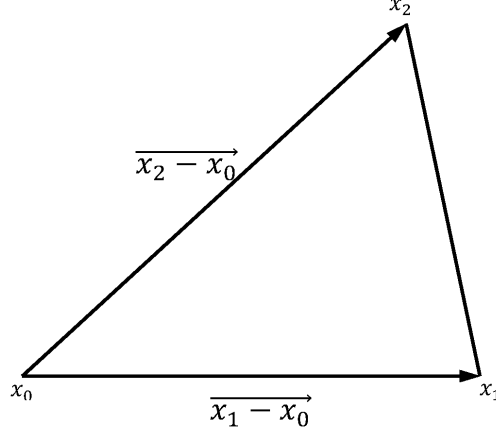


Figure 2.4: The Jacobian matrix for the element being studied

are the two edge vectors emanating from vertex zero of the element and defined as:

$$A = \left( \begin{array}{cc} \overrightarrow{x_1 - x_0} & \overrightarrow{x_2 - x_0} \end{array} \right) \quad (2.25)$$

To supply a high-level quality the Target matrix  $W$  is defined which is derived from the metric and must be constructed prior to the mesh optimization step. Target matrices play a critical role in TMOP because they define the desired Jacobian matrices in the optimal mesh. Fig. 2.5 demonstrates the desired element for an isotropic mesh, an equilateral triangle. The target matrix is defined as the Jacobian matrix for this target element, i.e. the columns are the two edge vectors emanating from vertex zero of this equilateral triangle. In this case,  $W$  is defined as:

$$W_{iso} = \left( \begin{array}{cc} 1 & \frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} \end{array} \right) \quad (2.26)$$

This matrix can be referenced to any ideal element, such as anisotropic elements in which we are interested. An example of an anisotropic element stretched by a factor of ten in the  $x$ -direction is shown in Fig. 2.6 for which the target matrix can be defined as:



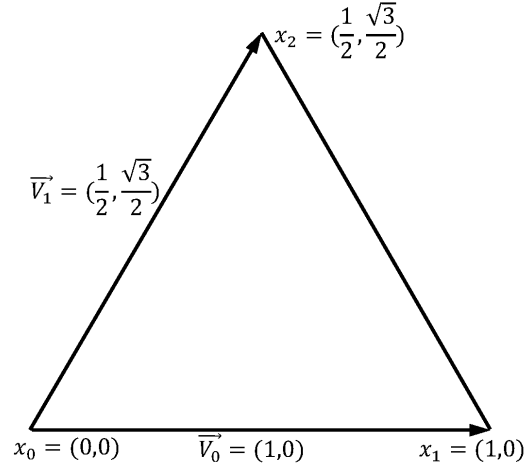


Figure 2.5: The Target matrix for the desired element of equilateral triangle

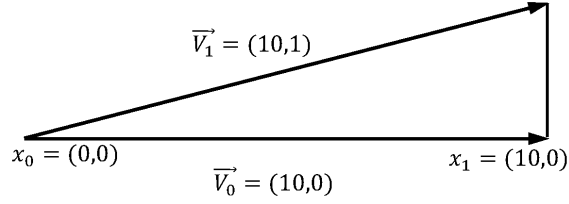


Figure 2.6: Anisotropic target element

$$W_{aniso} = \begin{pmatrix} 10 & 10 \\ 0 & 1 \end{pmatrix} \quad (2.27)$$

The Target Matrix Optimization Paradigm compares an actual element to the ideal element using the weighted Jacobian matrix  $T$  which is a non-dimensional matrix which provides a convenient scaling of  $A$ . Because the target matrix  $W$  is an element Jacobian,  $\det(W) \neq 0$  and thus  $W^{-1}$  exists.

$$T = AW^{-1} \quad (2.28)$$

A local quality function  $\mu = \mu(T)$  measures the relationship between  $A$  and  $W$ .

By applying QR-factorization<sup>4</sup>, separating out quality into three matrix factors describing size, orientation (angle), shape (aspect ratio) is possible [16].

Some relationships between  $A$  and  $W$  which are of interest are discussed here:

- If an element has the ideal size, shape and orientation, then  $A = W$  and  $T = I$  (the identity matrix). An example of a local quality function which attempts to enforce this relationship is <sup>5</sup>

$$\mu = |T - I|_F^2$$

in which  $\mu = 0$  is the ideal value, achieved if and only if  $T = I$ .

- If an element has the ideal size and shape but not the ideal orientation, then  $A = RW$  and  $T = R$  where  $R$  is a rotation matrix. An example of a local quality function which attempts to enforce this relationship is

$$\mu = |T^t T - I|_F^2$$

in which  $\mu = 0$  is the ideal value achieved if and only if  $T = R$ .

- If an element has the ideal shape but not the ideal size and orientation, then  $A = sRW$  and  $T = sR$  where  $s$  is a positive scalar. An example of a local quality function which attempts to enforce this relationship is

$$\mu = |T|_F |T^{-1}|_F$$

in which  $\mu = 2$  is the ideal value (for a 3D mesh, however  $\mu = 3$ ). This relationship is satisfied if and only if  $T = sR$ .

Quality functions used in TMOP can be determined by one of the main functions, shape, size+shape, size+shape+orientation functions or a combination of these quality functions.

Putting this all together to form a quality function for the entire mesh or a subset of it, an objective function to be optimized to improve mesh quality is

$$F = \frac{1}{N} \sum_{n=1}^N \mu(T_n) \quad (2.29)$$

---

<sup>4</sup>The QR-factorization of a matrix  $A$  is a decomposition of that matrix into a product  $A = QR$  of an orthogonal matrix  $Q$  and an upper triangle matrix  $R$ .

<sup>5</sup>The Frobenius matrix norm is defined as  $|A|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}$  for  $A_{m \times n}$ .

is obtained in which  $n$  is the sample element index and  $N$  is the total number of elements [22]. The objective function, which is nonlinear, should be minimized to find the optimal mesh. Both local and global optimization can be applied by defining the objective function as shown by changing the sum within a local patch or all elements in the global mesh.

TMOP has several advantages [33, 32] compared with other methods for mesh optimization. Dealing with a wide variety of mesh and element types is much easier requiring only a change in target matrix. By using target matrices, it is possible to take advantage of the extensive mathematical and numerical theories of matrices, furthermore addressing more than one mesh quality need less effort in comparison with other methods. While several objective functions must be combined together to achieve simultaneous control of length, area, and angles in other methods, TMOP can do this using only one objective function as a result of controlling all these properties via a single target-matrix.

Knupp [35] has shown the robustness of TMOP by applying it to different geometries, element types and mesh sizes. An initial mesh on the surface of a sphere with different triangles was improved to a mesh with equilateral triangles by using a shape quality function and constructing the target matrix as in Eq. 2.26. Another example in that paper is a problem involving a viscous boundary layer which has good quality in the boundary layer but contains sliver elements in the far-field. Improving the shape of the sliver elements while preserving the boundary layer mesh and retaining the good elements in the size-transition region between the boundary layer and far-field was the goal of this problem. Two different target matrices were defined in this problem, one to preserve those parts of the initial mesh in the boundary layer and one to create better-shaped elements elsewhere. Hence, the objective function is the sum of two different objectives based on different quality functions. The flexibility of TMOP allows trying out a variety of combination of objective functions fairly quickly to find a suitable one. There are many other examples show the robustness and effectiveness of the Mesquite as a mesh improvement toolkit for mesh generation including [66, 31, 30, 32, 27].

Because of the flexibility of the TMOP for controlling element size, shape and orientation, it is a very good choice for anisotropic mesh adaptation with the aim of generating aligned quasi-structured meshes, and will be used in this thesis.

Mesquite's TMOP implementation is used for node movement in this research. The conjugate gradient and trust region methods are used as the optimization methods.

# Chapter 3

## Methodology

Mesh quality, which is defined as geometric properties of a mesh, can adversely affect solution accuracy or computational efficiency of numerical solutions. Several geometric mesh properties are important to achieve good quality meshes. These include mesh smoothness, local mesh size, local mesh angle, aspect ratio and orientation. This chapter is devoted to the discussion of different techniques applied to improve mesh quality. After introducing the quality measure that is used in this thesis in the first section, the four principal operations to improve the quality of the mesh are described. These operations are face swapping, vertex insertion, vertex removal and vertex movement which are addressed in Section 3.2. Finally, the general algorithm applied to produce quality quasi-structured anisotropic meshes is introduced.

### 3.1 Mesh Quality Measures

Mesh quality refers to geometric properties of a mesh, such as local mesh size, orientation and local shape including minimum/maximum angle and aspect ratio of the element. Badly shaped elements can adversely influence the solution and lead to inaccurate, inefficient, or even unstable approximations. Several measures of triangle quality have been proposed based on dimensionless ratio of various geometric parameters including minimum angle, maximum angle, the ratio of inscribed circle to circumcircle radius, edge ratio which is the ratio of largest edge to shortest edge, matrix norm, the ratio of area of the element to perimeter squared and shortest edge to circumradius [57]. Since the final goal of this thesis is to produce highly anisotropic quasi-structured meshes, both aspect ratio and orientation are important. Accordingly, a quality measure based on norm of the triangle matrix has been chosen which contains information about both aspect ratio and orientation of the mesh.

Given a mesh element, an element quality metric is defined to measure the relationship between the Jacobian matrix  $A$  and the Target matrix  $W$  as discussed in Section 2.4.1.

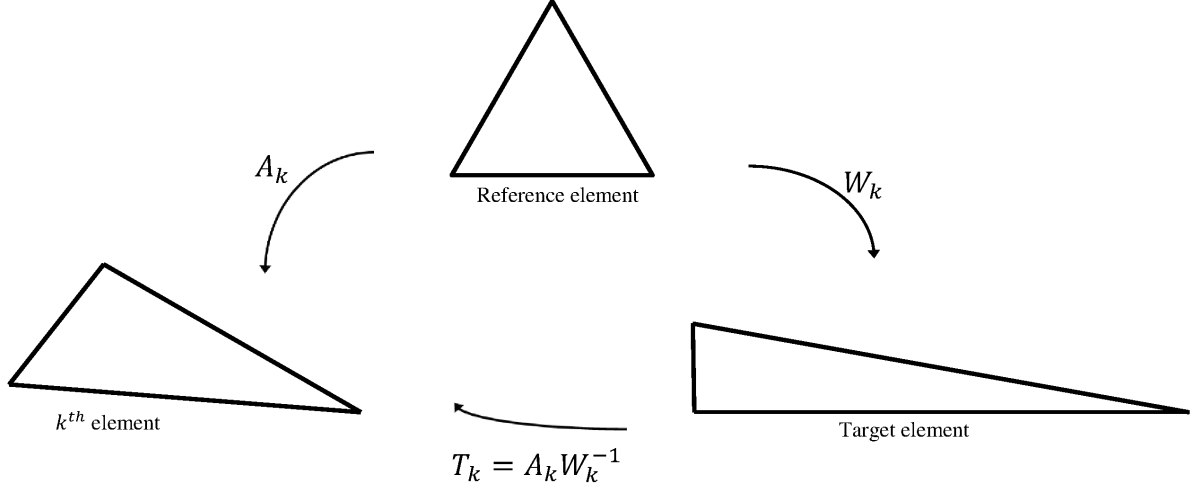


Figure 3.1: Weighted Jacobian matrix definition

As highly stretched right triangles are used as the desired element in anisotropic quasi-structured mesh adaptation, the target triangle in Fig. 3.1 is a right triangle which is stretched in one direction. The reference element shown in this figure is constructed by placing one node at the origin and the second node at unit length along one Cartesian axis and then placing the third one such that an equilateral triangle is constructed. The weighted Jacobian matrix  $T$  maps  $W$  into  $A$  and provides a convenient scaling of  $A$ . Since both  $A$  and  $W$  have the same units (length),  $T$  is a non-dimensional matrix. Matrix-norm quality measures use the weighted Jacobian matrix  $T$  to define element quality measure [34]. This matrix can be decomposed into geometrically meaningful matrices representing size, orientations and shape as it discussed earlier. Based on the precise requirements for the shape, size and orientation of the final mesh, a combination of decomposed matrices are chosen.

Having computed the weighted Jacobian matrix, a scalar quantity should be defined to measure mesh quality. Converting matrices to a scalar quantity is possible by taking the determinant, trace<sup>6</sup> or norm of the matrix.

The quality criterion used for vertex movement, one of the four principal operations for mesh quality improvement, is a matrix-norm quality measure. For the other three

---

<sup>6</sup> $trace(T) = \sum_i T_{ii}$

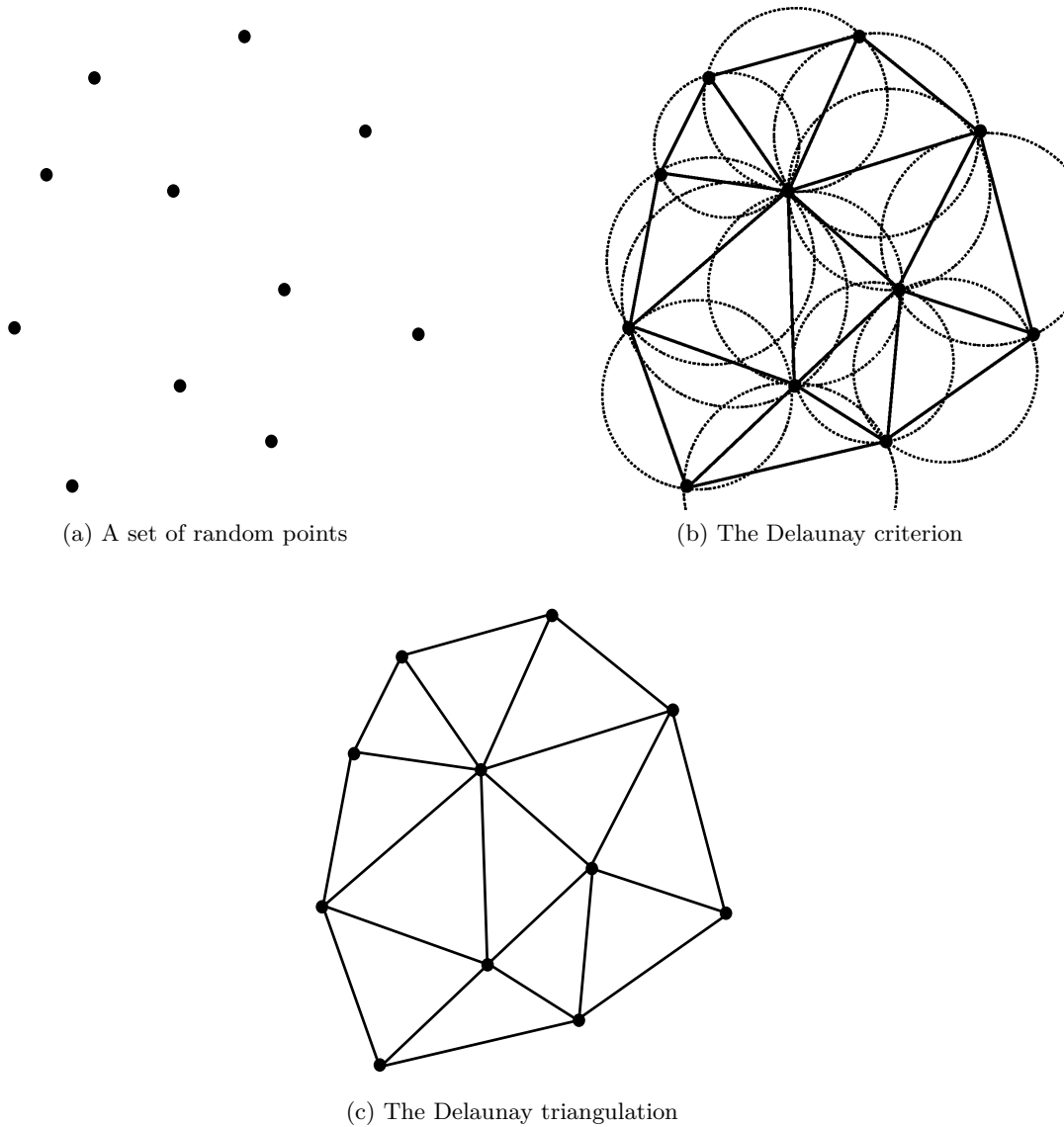


Figure 3.2: Delaunay triangulation of a set of random points

operations, Delaunay criteria is used. The Delaunay triangulation for a set of points in a plane is a triangulation such that no point is inside the circumcircle of any triangles; however, points on the circumcircles are allowed. As depicted in Fig. 3.2 there is no point inside the circumcircle of any triangle, hence the elements are Delaunay triangles.

Delaunay triangulations maximize the minimum angle among all possible triangulations

of a given point set [25]. The most elegant measure for analyzing Delaunay refinement is the ratio of circumradius  $R$  to shortest edge  $l_{min}$  according to Miller *et.al* [44]. For combined refinement and de-refinement, a range of acceptable triangle qualities are defined such as

$$\begin{aligned} \frac{l_{min}}{\sqrt{A_{des}}} &< B_l + \tau \\ \frac{R}{\sqrt{A_{des}}} &> B_R - \tau \end{aligned} \quad (3.1)$$

based on the desired element[55].

$\sqrt{A_{des}}$  is the ideal area of the element and shortest edge and circumradius of the the element is normalized by that, so the quality measure parameters will be dimensionless. Triangles with  $\frac{l_{min}}{\sqrt{A_{des}}} < B_l + \tau$  are considered large triangles for which a new vertex is inserted on the circumcenter of that element and triangles with  $\frac{R}{\sqrt{A_{des}}} > B_R - \tau$  are considered small elements for which one of the vertices on the shortest edge is removed. The values of  $B_l$  and  $B_R$  are chosen base on the desired triangle. For example, for an equilateral triangle which is the desired element shape for isotropic physical behavior, it can be shown that the value of  $B_l$  and  $B_R$  are:

$$\begin{aligned} \frac{l_{min}}{\sqrt{A_{des}}} &= \frac{2}{3^{\frac{1}{4}}} \\ \frac{R}{\sqrt{A_{des}}} &= \frac{2}{3^{\frac{3}{4}}} \end{aligned}$$

The range of acceptable triangle qualities are defined based on the value of  $\tau$ . To ensure shape quality, the value of  $\tau$  is restricted to ensure shape quality so that a lower bound of the minimum angle of the triangle is obtained and it can be written as  $\frac{l_{min}}{R} = 2 \sin \theta_{min}$ .

## 3.2 Mesh Operations

There are four principal operations that are used in our algorithm to improve the quality distribution of triangles in the mesh. These methods are face swapping, vertex insertion, vertex removal and vertex movement. The last operation, vertex movement, seeks to reposition existing mesh vertices to achieve better quality, and is implemented by Mesquite which was discussed in the previous chapter. Other three operations, face swapping, vertex insertion and vertex removal are implemented using GRUMMP<sup>7</sup>. GRUMMP is a set of libraries, written in C++ which support unstructured mesh creation and modification [50]. The goal of the GRUMMP project is to develop automatic mesh generation software for

---

<sup>7</sup>Generation and Refinement of Unstructured Mixed-element Meshes in Parallel

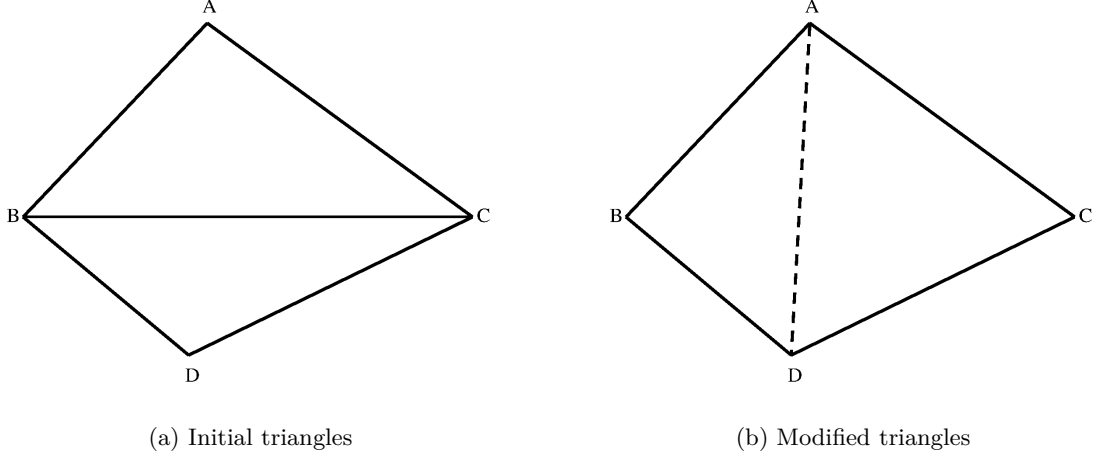


Figure 3.3: Face swapping

unstructured meshes with mixed element types. This package can produce high-quality meshes that meet user-defined mesh density requirements.

### 3.2.1 Face Swapping

The simplest operation is face swapping which chooses the best diagonal for the quadrilateral formed by two neighboring triangles. Fig. 3.3 illustrates face swapping for which the initial triangles are  $ABC$  and  $BCD$ . The common face  $BC$  is swapped to  $AD$  and new triangles  $ABD$  and  $ADC$  are created. Face swapping chooses either  $BC$  or  $AD$  to maximize the quality of the two elements, i.e, the diagonal which maximizes the minimum quality is chosen. These two triangles which have better quality, i.e. those which are closer to the desired elements, are chosen. Specifically, face swapping is done if

$$Quality(ABC) + Quality(BCD) < Quality(ABD) + Quality(ADC) \quad (3.2)$$

Otherwise, swapping is not applied.

By applying this operation, the number of mesh points and the number of mesh elements does not change. However, the connectivity of points are changed.

In the case of an anisotropic mesh, a mapped triangle from edge lengths is constructed.



Edge lengths are computed in the metric space, for example for edge AB in Fig. 3.3 as:

$$|\overrightarrow{BA}|_{mapped} = \left( \begin{bmatrix} X_B - X_A \\ Y_B - Y_A \end{bmatrix}^T M \begin{bmatrix} X_B - X_A \\ Y_B - Y_A \end{bmatrix} \right)^{\frac{1}{2}} \quad (3.3)$$

The circumradius of the mapped triangle is calculated based on anisotropic length. The mapped triangles are different depending on the diagonal which has been chosen. Face swapping is applied if the average circumradius corresponding to the mapped triangles of  $ABD$  and  $ADC$  is smaller than the average circumradius corresponding to the mapped triangles of  $ABC$  and  $BCD$ . This is a quasi-Delaunay extension, since the Delaunay criterion is applicable only for isotropic metrics [55].

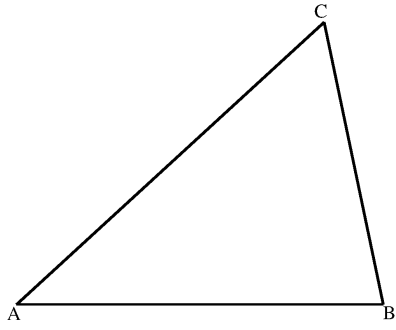
Since the quality function is dependent on vertex connectivity, swapping an edge changes the metric and for new elements, new metrics should be defined based on their vertices.

It should be noted that the diagonal of the quadrilateral formed by two adjacent triangles can be only swapped if the quadrilateral is convex.

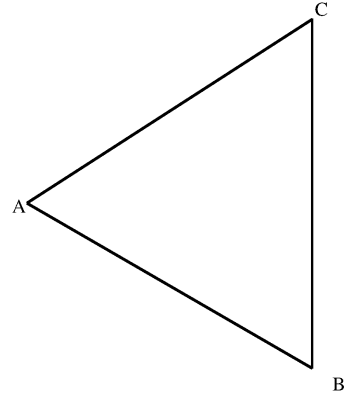
### 3.2.2 Vertex Insertion

The second mesh modification operation is vertex insertion for triangles with large circumradii or poor shape. For interior cells of the mesh, the new vertex is inserted at the actual circumcenter of an isotropic triangle [55]. In the case of an anisotropic mesh, the circumcenter of the triangle is calculated based on anisotropic length, i.e. the metric-based length. The initial triangle is transformed to one with lengths as described by the metric and then the circumcenter of this mapped triangle is calculated. The following example shows how to calculate the location of the new vertex. The triangle  $ABC$  shown in Fig. 3.4a is mapped to the triangle with edge lengths that are computed using the metric. Metrics assigned to the vertices of the initial triangle are averaged to obtain the metric associated with the triangle  $M$ . The length of edges of this triangle is measured as Eq. 3.3.

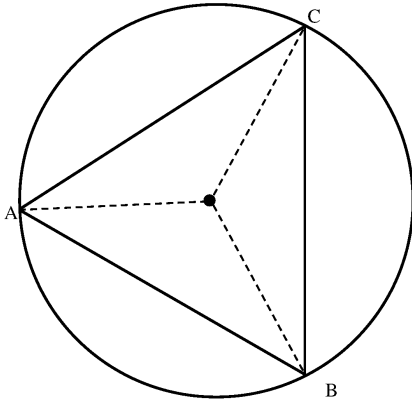
Having found the circumcenter, the triangle is transformed back to the actual mesh, a vertex is inserted at that location, and three triangles are created. The resulting three triangles have smaller circumradius and no edge smaller than the original circumradius is created. By applying vertex insertion, the number of mesh vertices are increased by one and the number of mesh elements increase by two (one triangle is converted to three triangles). A metric should also be assigned to the new vertex based on a linear interpolation from the three vertices of the triangle in which the vertex is inserted.



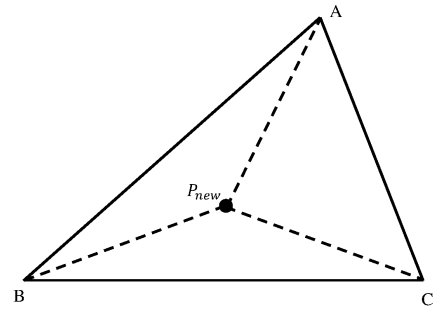
(a) Initial triangle in physical space



(b) Mapped triangle in metric-based mapped space



(c) Circumcenter in mapped triangle in metric-based mapped space



(d) Metric-based circumcenter in physical space

Figure 3.4: Vertex insertion procedure

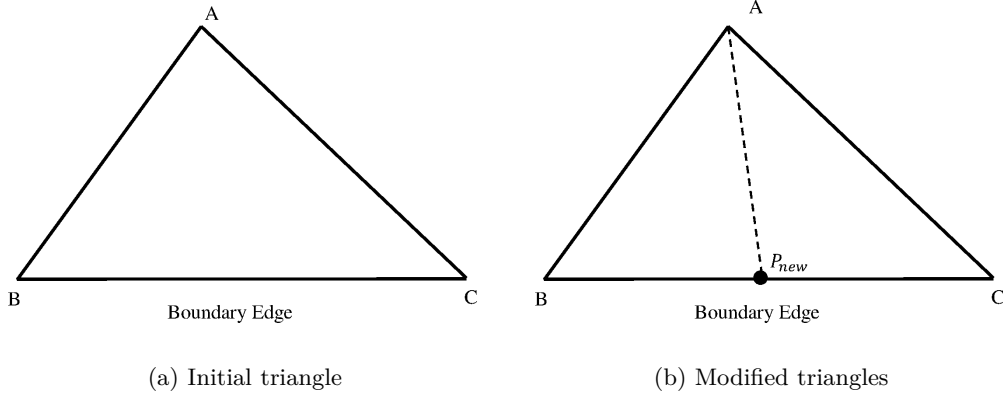


Figure 3.5: Point insertion on a boundary face

If the circumcircle of the mapped triangle being eliminated by insertion is not empty in the metric space, i.e. a vertex of another cell in the mapped mesh is inside it, the longest edge of the original triangle is split instead. This is the same approach applied for cases where the circumcenter of the triangle lies outside the domain or across a boundary. When this happens, the boundary edge is split instead. Fig. 3.5 shows an initial triangle in which boundary edge  $BC$  is split and consequently two new boundary edges  $BP$  and  $PC$  are constructed.

For those cases in which the circumcenter of a triangle lies inside the domain but near the boundary, that boundary segment is split. Previous theoretical results prove the effectiveness of boundary protection for isotropic meshes. When new vertices are forbidden inside a circle whose diameter is a boundary edge, Ruppert [64] showed that the final mesh has minimum angle  $\theta_{min} \approx 20.7^\circ$ . If edges are protected by a diametral lens, found by taking the intersection of the two circles of radius  $\frac{L}{\sqrt{3}}$  with the boundary segment as a chord, Shewchuk [68] showed that  $\theta_{min} \approx 25.7^\circ$  is possible.

The difference between the diametral circle and diametral lens is shown in Fig. 3.6. For an anisotropic mesh, the diametral lens is approximated by transforming any potential boundary triangles using the average metric of the two boundary vertices.

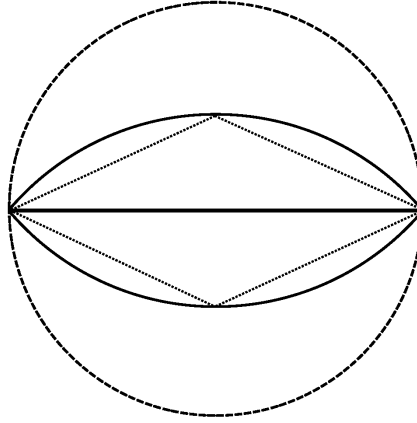


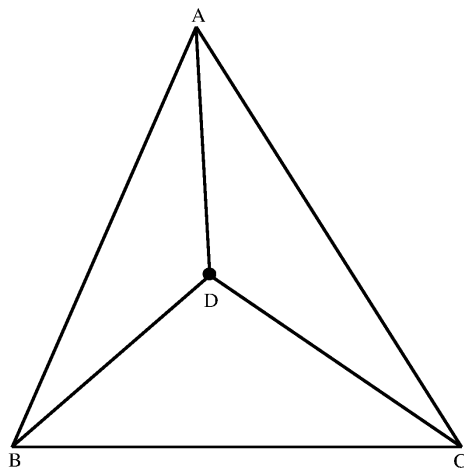
Figure 3.6: Comparison between a diametral circle (dashed) and diametral lenses

### 3.2.3 Vertex Removal

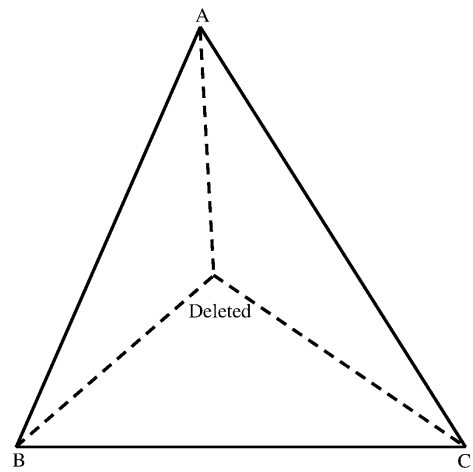
Another operation that is used to improve the quality of the mesh is vertex removal. Removing vertices increases edge lengths in the mesh. The mesh is coarsened with the goal of reducing mesh size while maintaining a good distribution of vertices. To remove a vertex, the edges incident on that vertex are first swapped until there are only three edges incident on that vertex. Then the three triangles incident in that vertex are combined to form one triangle as shown in Fig. 3.7. Three triangles  $ABD$ ,  $ACD$ ,  $BCD$  are combined and the new triangle  $ABC$  is created and the point  $D$  is deleted. The edges are then swapped recursively as described in Section 3.2.1.

If swapping is not allowed for some specific edges and consequently the number of triangles incident on the point which should be deleted is four instead of three, another approach is used. Those four triangles are combined with the aim of creating two triangles and then the edges are swapped recursively. An example is shown in Fig. 3.8. There are four triangles in the initial mesh which are modified and two triangles with larger circumradius are constructed.

When the vertex that should be deleted is located on a boundary edge, a different approach is used which is the reverse of the approach for vertex insertion on a boundary edge discussed in the last section. There are two triangles incident on a boundary vertex; by deleting the vertex  $D$  shown in Fig.3.9 a new triangle is constructed with the edge  $BC$ .

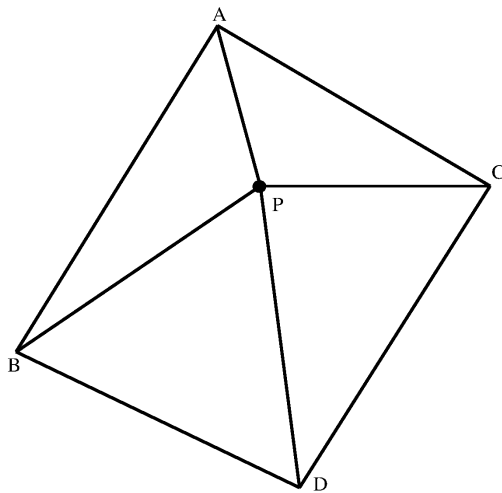


(a) Initial triangles

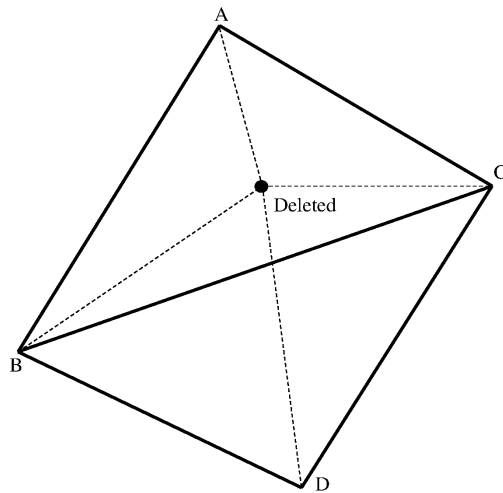


(b) Modified triangle

Figure 3.7: Vertex removal



(a) Initial triangles



(b) Modified triangles

Figure 3.8: Vertex removal for four triangles

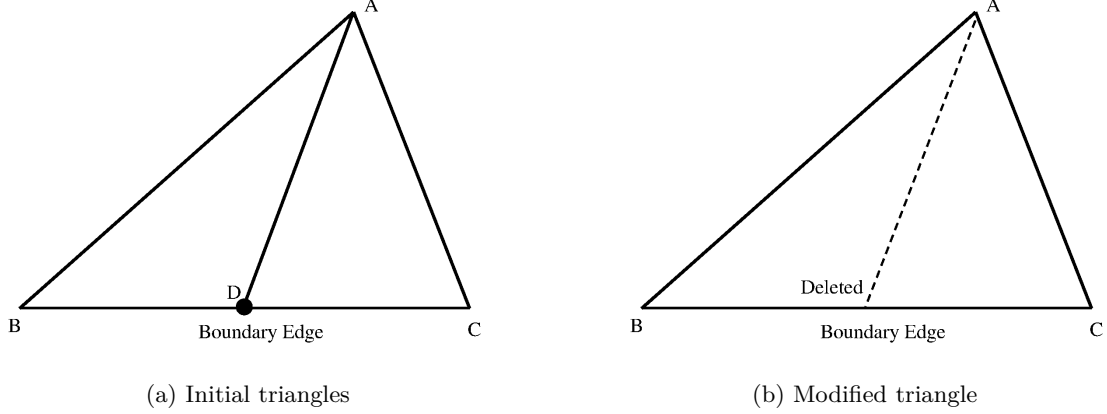


Figure 3.9: Vertex removal for a boundary point

### 3.2.4 Vertex Movement

In vertex movement schemes, existing mesh vertices are repositioned to achieve better quality. Given a metric to measure mesh quality, one can formulate a numerical optimization problem which guides vertex movement to find the optimal mesh and thus improve its quality. As mentioned in the last chapter, the algorithm used for vertex movements in this research is the Target-Matrix Optimization Paradigm (TMOP). The target element for anisotropic meshes is a triangle with aspect ratio and orientation defined by the metric. Since we are looking for a quasi-structured mesh, the target element should be a right triangle. Since every sample element has a different Target-matrix, supplying this information would seem to constitute an enormous burden. Accordingly, the target matrix should be calculated from the mesh metric to specify size, shape and anisotropy of the mesh. The metric is given by:

$$M = \begin{pmatrix} m_{xx} & m_{xy} \\ m_{xy} & m_{yy} \end{pmatrix} \quad (3.4)$$

#### 3.2.4.1 Average Metric

Since the flow solver computes the vertex-based metric, some kind of averaging is needed to calculate the target matrix for a triangle from the metrics assigned at each vertex. If  $P_1 = (x_1, y_1)$ ,  $P_2 = (x_2, y_2)$  and  $P_3 = (x_3, y_3)$  are the vertices of a triangle  $\Delta_{p_1 p_2 p_3}$  and the metrics assigned at each vertex are  $M_1$ ,  $M_2$  and  $M_3$  respectively, the metric  $M$

corresponding to  $\Delta_{p_1 p_2 p_3}$  can be calculated based on the following matrix  $D$ :

$$D = \begin{pmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{pmatrix} \quad (3.5)$$

If  $\det(D) = 0$ , which means that the three vertices are colinear, then the metric corresponding to  $\Delta_{p_1 p_2 p_3}$  is simply the average of three metrics:

$$M = \frac{1}{3} (M_1 + M_2 + M_3) \quad (3.6)$$

Otherwise, the metric-based average is calculated. The center of the mapped triangle with metric-based edge lengths as shown in Fig. 3.4c is considered to be  $(x_c, y_c)$ . By defining  $w$  as:

$$w = D^{-1} \begin{pmatrix} 1 \\ x_c \\ y_c \end{pmatrix} \quad (3.7)$$

The average metric  $M$  is then calculated by:

$$\begin{aligned} M_{xx} &= w \cdot M'_{xx} \\ M_{xy} &= w \cdot M'_{xy} \\ M_{yy} &= w \cdot M'_{yy} \end{aligned} \quad (3.8)$$

where  $M'_{xx}$ ,  $M'_{xy}$ , and  $M'_{yy}$  are defined as:

$$M'_{xx} = \begin{pmatrix} M_{1,xx} \\ M_{2,xx} \\ M_{3,xx} \end{pmatrix}, \quad M'_{xy} = \begin{pmatrix} M_{1,xy} \\ M_{2,xy} \\ M_{3,xy} \end{pmatrix}, \quad M'_{yy} = \begin{pmatrix} M_{1,yy} \\ M_{2,yy} \\ M_{3,yy} \end{pmatrix} \quad (3.9)$$

### 3.2.4.2 Isotropic Elements

Having defined the average value of the metric for each triangle, the target matrix can be calculated. The flow solver computes several isotropic metrics. The metric for an isotropic

element with size  $s$ , is

$$M = s \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

where the target element is an equilateral triangle as shown in Fig. 2.5 and the corresponding target matrix is

$$T = \begin{pmatrix} 1 & 1/2 \\ 0 & \sqrt{3}/2 \end{pmatrix}$$

So, for those metrics in which  $m_{xy}$  is zero and  $m_{xx} = m_{yy}$ , the above target matrix is assigned to the element. It should be noted that we do not take the length scale into account in vertex movement and consequently in the target matrix. Accordingly, regardless of the value of  $m_{xx}$ , as long as  $m_{xx} = m_{yy}$  and  $m_{xy} = 0$ , the target matrix is the same. The tolerance of equality is set as:

$$\frac{|m_{xx}-m_{yy}|}{\sqrt{\det(M)}} < 10^{-6}, -10^{-6} < \frac{m_{xy}}{\sqrt{\det(M)}} < 10^{-6}$$

### 3.2.4.3 Anisotropic Elements

For anisotropic elements the target matrix is calculated based on eigenvalues and eigenvectors of the metric  $M$ . Since the metric is a  $2 \times 2$  matrix for two-dimensional problem, there are two eigenvalues for  $M$ :

$$\lambda = \frac{1}{2} \left( (m_{xx} + m_{yy}) \pm \sqrt{(m_{xx} - m_{yy})^2 + 4m_{xy}^2} \right) \quad (3.10)$$

The larger eigenvalue is assumed to be  $\lambda_1$  and the smaller one is  $\lambda_2$ . The normalized eigenvectors are then calculated such that the equation  $MV = \lambda V$  holds. There are two eigenvectors  $V_1$  and  $V_2$ , one corresponding to each eigenvalue:

$$V_1 = \frac{1}{\sqrt{(m_{xx} - \lambda_1)^2 + (m_{xy})^2}} \begin{pmatrix} m_{xx} - \lambda_1 \\ -m_{xy} \end{pmatrix} \quad (3.11)$$

$$V_2 = \frac{1}{\sqrt{(m_{xx} - \lambda_2)^2 + (m_{xy})^2}} \begin{pmatrix} m_{xx} - \lambda_2 \\ -m_{xy} \end{pmatrix} \quad (3.12)$$

where  $V_1$  and  $V_2$  correspond to the larger and smaller eigenvalues respectively. If  $m_{xy} = 0$ ,



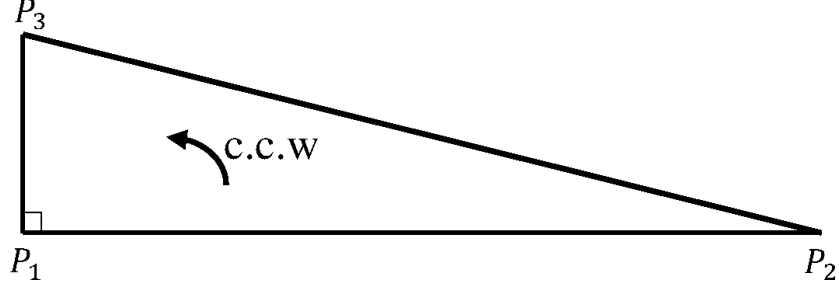


Figure 3.10: Correct order and orientation for smoothing

then the eigenvalues are  $m_{xx}$  and  $m_{yy}$  for which the eigenvectors are  $\frac{0}{0}$ . In this case, the eigenvectors are defined as:

$$V_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, V_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (3.13)$$

The target matrix is then can be calculated based on the eigenvalues and eigenvectors as:

$$T = \begin{pmatrix} V_1 & V_2 \end{pmatrix} \begin{pmatrix} \sqrt{\lambda_2} & 0 \\ 0 & \sqrt{\lambda_1} \end{pmatrix} \quad (3.14)$$

It should be noted that each eigenvector is scaled by the eigenvalue corresponding to the other eigenvalue, i.e.  $V_1$  is scaled by  $\sqrt{\lambda_2}$  and vice versa.

#### 3.2.4.4 Orientation and Order

As mentioned before, the target matrix should be constructed to correspond to the Jacobian matrix of a right-triangle element with the first vertex in the element being the one at the right angle and the other two in counter-clockwise order. Therefore, it is important to construct the target matrix such that the correct orientation is preserved. As we are looking for quasi-structured elements, as shown in Fig. 3.10, each element has a right angle at  $P_1$  and two acute angles  $P_2$  and  $P_3$ . Since the target element is anisotropic,  $\angle P_3 \gg \angle P_2$ ; so it is possible to distinguish these three angles once the initial mesh is read. All elements are stored so that the first vertex is the one with right angle  $P_1$  and then  $P_2$  and  $P_3$  in counter-clockwise order. This is an important issue in constructing target matrix. For

example in Fig. 3.10, the correct target matrix is:

$$T = \begin{pmatrix} 10 & 0 \\ 0 & 1 \end{pmatrix}$$

In which the first column is the vector  $\overrightarrow{P_2P_1}$  and the second column is  $\overrightarrow{P_3P_1}$  in counter-clockwise order. The target matrix calculation may result in clockwise order matrix such as:

$$T' = \begin{pmatrix} 0 & 10 \\ 1 & 0 \end{pmatrix}$$

which should be modified by exchanging columns. By calculating target matrix based on the description in Section 3.2.4.3, the correct orientation can be chosen based on the positive determinant of the target matrix. The target matrix has counter-clockwise order if

$$T_{11}T_{22} > T_{12}T_{21} \quad (3.15)$$

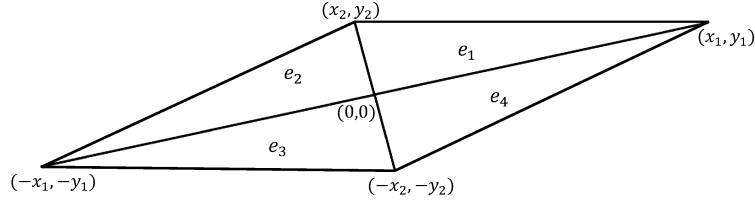
Therefore if the determinant of the calculated target matrix is negative, the columns are exchanged.

All four triangles shown in Fig. 3.11 have the same metric. These triangles occur at the four corners of a high aspect ratio quadrilateral. Elements  $e_1$  and  $e_3$  can form a quadrilateral with the diagonal orientation depicted in the right half of Fig. 3.11b and elements  $e_2$  and  $e_4$  can form a quadrilateral with the other diagonal orientation which is shown in the left half. Since we are looking for quasi-structured meshes, those two diagonal orientations make no difference in the final results. Accordingly, using each of those four target matrices should result in the same final mesh. Experiments have shown that if the combination of  $e_1$  and  $e_3$  is chosen instead of  $e_2$  and  $e_4$ , the final mesh has elements which are closer to the desired target. Therefore if the target matrix is constructed as

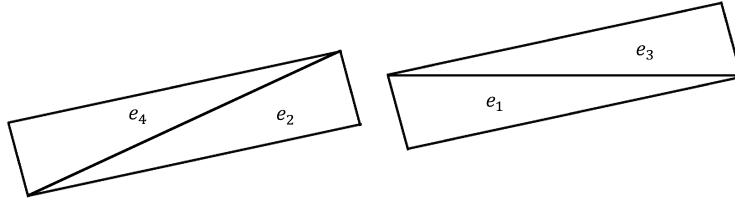
$$T' = \begin{pmatrix} x_2 & -x_1 \\ y_2 & -y_1 \end{pmatrix} \quad (3.16)$$

which is the target matrix for elements  $e_2$  and  $e_4$ , it is replaced by

$$T = \begin{pmatrix} x_1 & x_2 \\ y_1 & y_2 \end{pmatrix} \quad (3.17)$$



(a) Four right triangles incident on a vertex



(b) Equilateral with the same diagonals

Figure 3.11: Decision on target element

which corresponds to elements  $e_1$  and  $e_3$ . By calculating target matrix based on the description in Section 3.2.4.3, the correct configuration can be chosen by the norm of two column of the target matrix. The target matrix is always constructed such that the norm of the first column is larger than the norm of the second column.

### 3.3 General Algorithm

The general algorithm applied to produce quasi-structured anisotropic meshes is described here and illustrated in a flowchart in Fig. 3.12. The initial mesh is modified based on the metric that is obtained through the adaptive process and therefore the final mesh matches the physical properties of the flow dictated by the metric.

- Accordingly, the first step is assigning a metric to each vertex by one of the methods described in Chapter 2. The flow solver that is used in this research computes a vertex-based metric as described in Section 2.2.
- Having set the metric for each vertex, the target matrices should be defined for each element based on the metric at its three vertices. The target matrix is calculated

from the element average metric as described in the previous section and used for mesh optimization by Mesquite. The target matrix is calculated for both anisotropic triangles in regions with anisotropic physical behavior and isotropic triangles for the rest of the domain.

- Edges are then swapped to obtain the maximum possible mesh quality with the current vertices. The modified mesh after swapping matches the target better and hence has a higher quality.
- For large or badly-shaped triangles, new vertices are inserted either at the circumcenter for interior elements or at the midpoint of a boundary face until there are no more large/badly-shaped triangles or the maximum number of vertices is reached. Based on a linear interpolation from the three vertices of the triangle in which the new point is inserted, a new metric is also assigned to the new vertex. After vertex insertion, there are new edges that should be recursively swapped to match the target mesh better.
- For small edges, vertices are removed until there are no more small edges or the maximum number of deleted vertices is reached. The maximum allowed number of deleted vertices is a fraction of the number of vertices inserted in the previous refinement step. Reduction of the overlap between vertex insertion and deletion can be obtained by limiting the number of removed vertices and consequently the process is always terminated regardless of the number of iterations. Since, after vertex removal, there are again new edges, another loop of edge swapping is required.
- The process is repeated with a new metric based on the refined mesh until all triangles are within the acceptable tolerance. Construction of large triangles with low quality is possible if the final process is vertex removal, therefore the process of vertex insertion and deletion always is terminated by vertex insertion.
- The final mesh optimization (smoothing) step is done followed by face swapping.

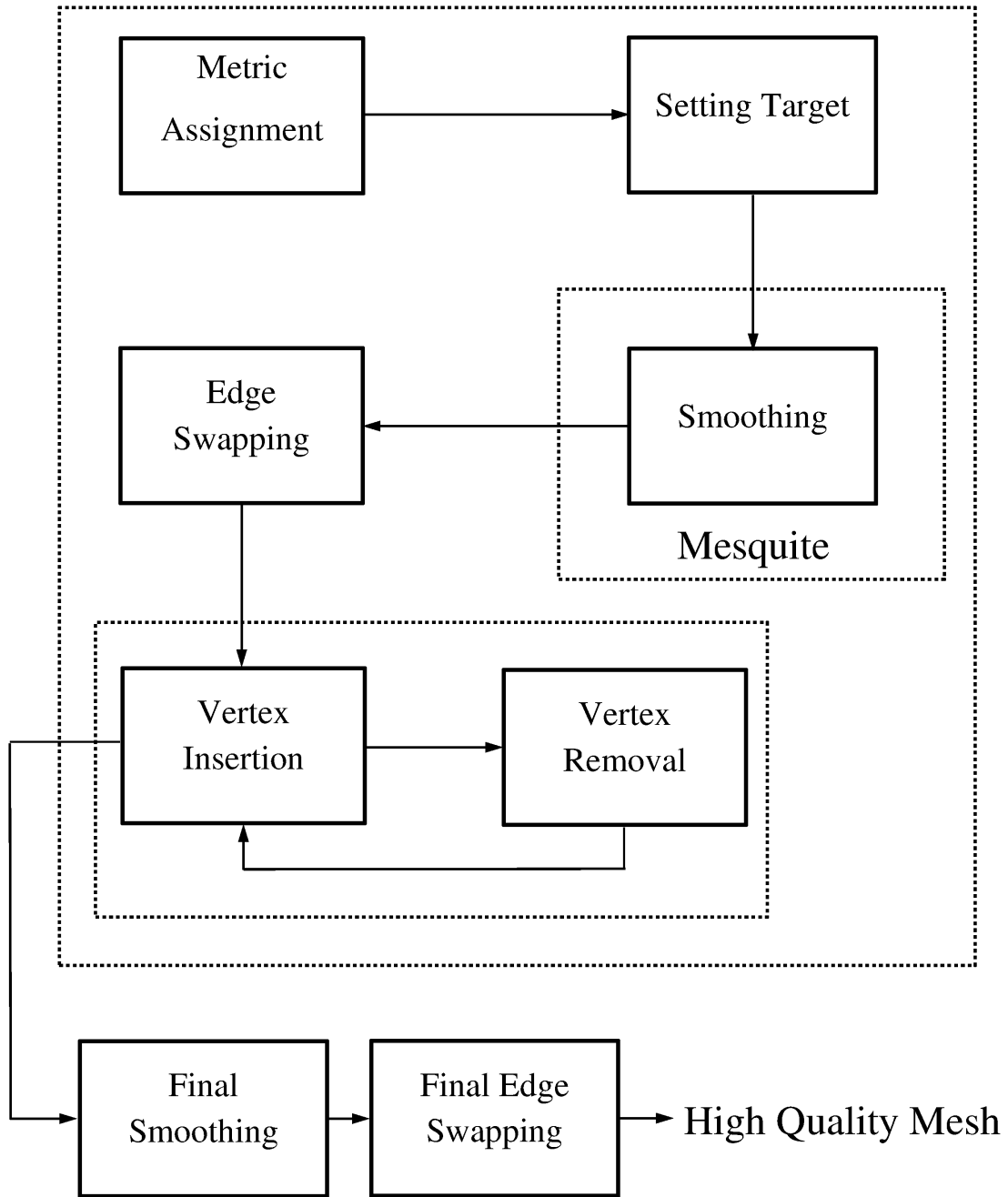


Figure 3.12: General algorithm

# Chapter 4

## Results

This section presents meshes generated by applying the proposed algorithm. The effect of each of four principal operations introduced in the last chapter are investigated by some preliminary tests. These examples show how each mesh modification techniques improves the mesh quality. The general algorithm is then applied to a domain with curve boundaries to show how the mesh is well suited to the desired target. A verification case is then introduced in the next section to demonstrate how the modified mesh is well suited to the desired target. Finally two test cases, subsonic viscous flow and transonic inviscid flow, are presented to show the robustness of our smoothing/adaptation algorithm.

### 4.1 Preliminary Tests

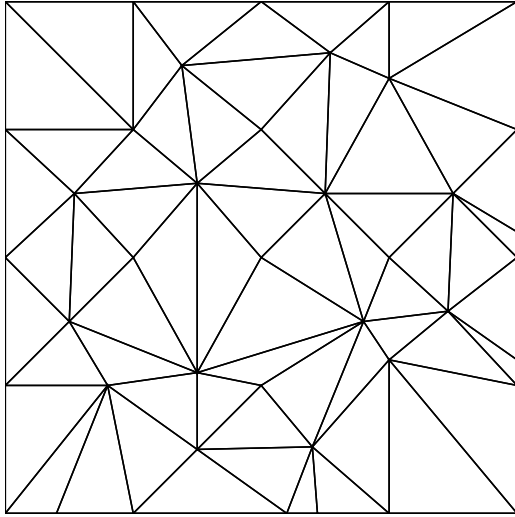
As mentioned in the previous chapter, there are four mesh modification techniques to improve the quality of the mesh. By face swapping, an initial mesh shown in Fig. 4.1a on a  $1 \times 1$  square domain is modified and the grid depicted in Fig. 4.1b is obtained. As shown here the faces of the initial isotropic grid are swapped to create the best possible anisotropic mesh for the same vertices from the metric

$$M = \begin{pmatrix} 100 & 0 \\ 0 & 1 \end{pmatrix}$$

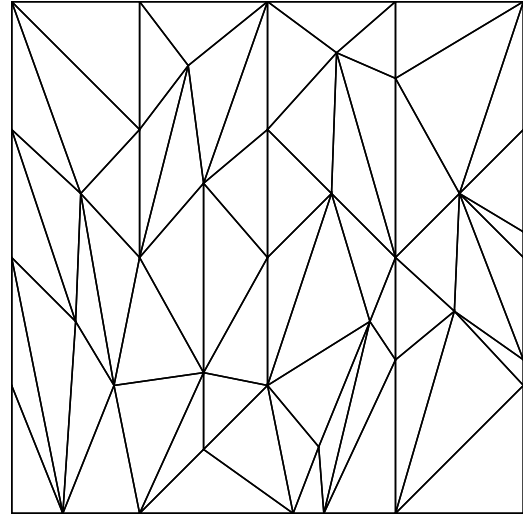
which corresponds to the triangles with target matrix

$$T = \begin{pmatrix} 1 & 0 \\ 0 & 10 \end{pmatrix}$$

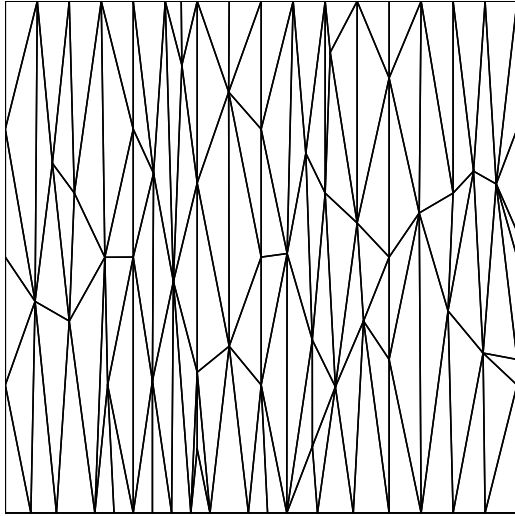
Therefore the final mesh should contain triangles stretched by a factor of ten in the  $y$ -direction. As depicted in this figure, the modified mesh contains stretched triangles. As the vertices are fixed, i.e. number of vertices and their locations are fixed, the final mesh contains elements with aspect ratios less than ten but as close as possible to the desired



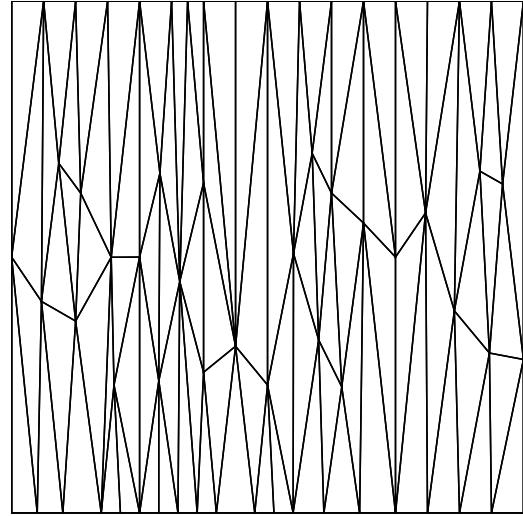
(a) Initial mesh



(b) The effect of swapping



(c) The effect of vertex insertion



(d) The effect of vertex removal

Figure 4.1: The effect of mesh modifying operations

target. Although by face swapping alone it is not possible to get the desired mesh, the modified mesh has higher quality in the sense of matching the metric.

This swapped mesh is used to demonstrate the effect of vertex insertion. The target metric is the same as the last example and point insertion continues until there are no more large triangles. The modified mesh has elements with high aspect ratio in the y-direction

<i>Operation</i>	<i>Mesh Quality</i>
Initial mesh	1.64829
Face Swapping	1.63627
Vertex Insertion	1.43321
Vertex Removal	1.42056

Table 4.1: Quality improvement by mesh modifying operation

as shown in Fig. 4.1c.

This modified mesh after point insertion is used as the initial mesh to depict the effect of vertex removal. The desired elements have the same metric as the last example

$$M = \begin{pmatrix} 100 & 0 \\ 0 & 1 \end{pmatrix}$$

and the final mesh after coarsening has elements which are closer to this target in comparison with the initial mesh as depicted in Fig. 4.1d. As shown in this figure, the modified mesh contains elements with higher aspect ratios.

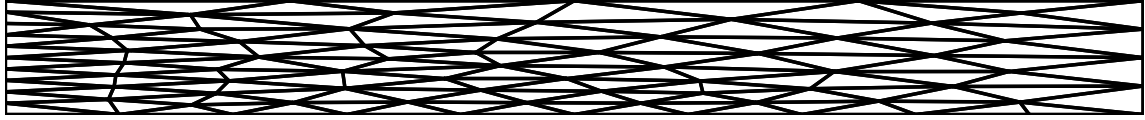
Table 4.1 shows how each of these operations improve mesh quality. The mesh quality is defined as the Frobenius norm of the weighted Jacobian matrix  $T$ . As discussed in Section 2.4.1, the weighted Jacobian matrix is defined as  $T = AW^{-1}$ . Knowing the target matrix and the Jacobian matrix for each element,  $T$  can be obtained and the average of the Frobenius norm of  $T$  is calculated. As we are looking for the size, shape, and orientation matrix, the ideal  $T$  is the identity matrix and the ideal Frobenius norm is  $\sqrt{2}$ . As demonstrated in Table 4.1, by applying each operation the mesh quality improves in the sense that it becomes closer to the ideal value  $\sqrt{2}$ .

To show how smoothing improves the mesh quality, a rectangular domain of size  $1 \times 10$  is considered. As shown in Fig. 4.2a, the initial mesh has elements that are anisotropic but not quasi-structured. The modified mesh after smoothing with the target matrix as

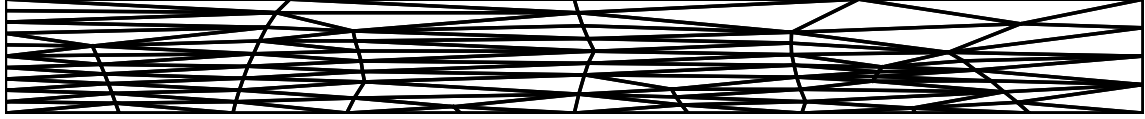
$$T = \begin{pmatrix} 10 & 0 \\ 0 & 1 \end{pmatrix}$$

has elements which are stretched in the  $x$ -direction and aligned, i.e. quasi-structured. There are some elements in the modified mesh that are not aligned properly since the number of mesh points is fixed and consequently, aligning all points is not possible. However, this example shows how vertex movements can improve the mesh quality especially when





(a) Initial mesh



(b) Modified mesh

Figure 4.2: The effect of vertex movement

	<i>Mesh Quality</i>
Initial Mesh	3.99734
Smoothed Mesh	2.03629

Table 4.2: Quality improvement by mesh modifying operation

quasi-structured meshes are desired. The same mesh quality measure as used for other operations is applied. Table 4.2 show that smoothing improves mesh quality significantly.

All these four test cases were applied for square or rectangular domain with straight lines as the boundary. Real test cases, however contains curved boundaries for which point insertion and deletion processes are more difficult to apply. The initial mesh on a circular domain centered at the origin and with radius of one (see Fig. 4.3a) has been chosen to show the robustness of the proposed algorithm. The following metric is used to modify this mesh:

$$M = \begin{pmatrix} 1 & 0 \\ 0 & m \end{pmatrix} \quad m = \begin{cases} 100 & -0.1 < y < 0.1 \\ 1 & \text{Elsewhere} \end{cases}$$

The Frobenius norm of the weighted Jacobian matrix is calculated in this case as a mesh quality measure. Table 4.3 illustrates how this adaptation/smoothing algorithm improves mesh quality.

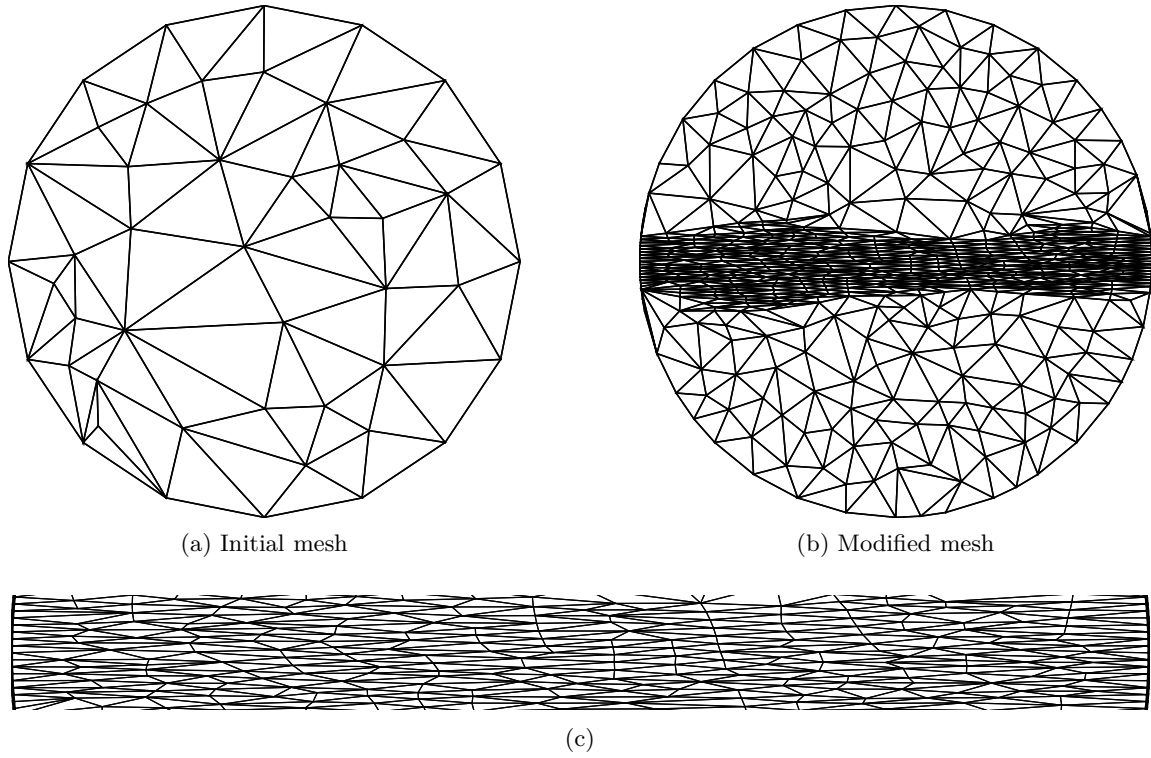


Figure 4.3: Anisotropic quasi-structured mesh on a curved boundary

	<i>Mesh Quality</i>
Initial Mesh	4.05429
Smoothed Mesh	1.62481

Table 4.3: Quality improvement by mesh modifying operation

## 4.2 Verification Case

As a verification case, the algorithm has been applied to a square domain containing two different anisotropies. The first one is a fake boundary layer which is highly anisotropic in one direction and the second one is a fake shock wave highly anisotropic in the perpendicular direction. Since the expected mesh is totally predefined, by comparing the desired mesh and the mesh that will be obtained by the proposed algorithm the robustness of the algorithm

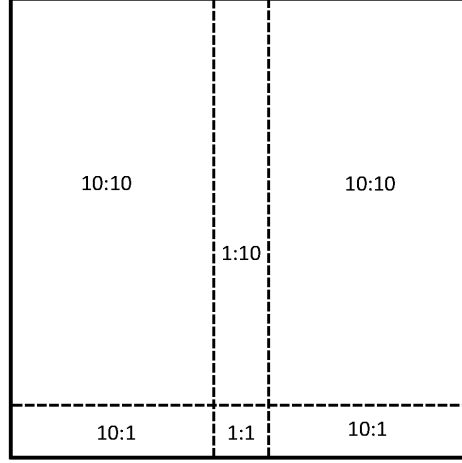


Figure 4.4: Expected triangle aspect ratio

may be demonstrated. The metric is defined by

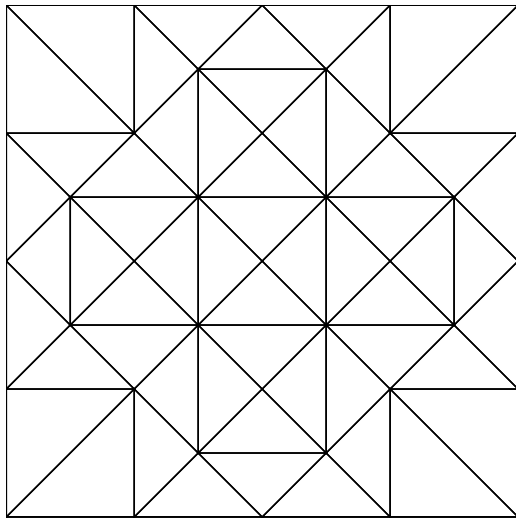
$$M = \begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix} \quad (4.1)$$

where  $a$  and  $b$  are as follows:

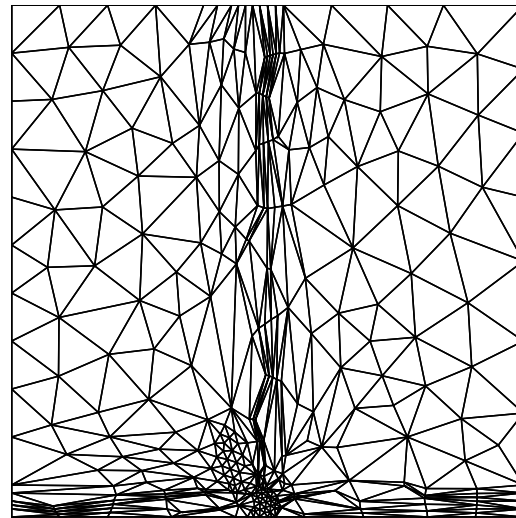
$$a = \begin{cases} 1 \times 10^2 & 0.475 < x < 0.525 \\ 1 & \text{Elsewhere} \end{cases} \quad (4.2)$$

$$b = \begin{cases} 1 \times 10^2 & 0 \leq y < 0.05 \\ 1 & \text{Elsewhere} \end{cases}$$

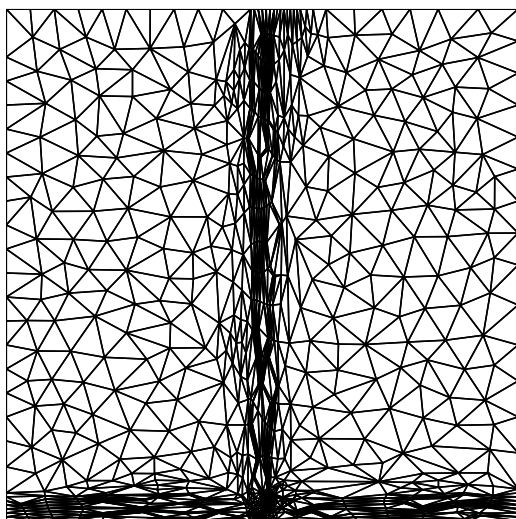
which means that the triangles should be stretched in the  $x$ -direction by a factor of ten for  $0 \leq y < 0.05$  and stretched by a factor of ten in the  $y$ -direction for  $0.475 < x < 0.525$  as shown in Fig. 4.4. The proposed algorithm is applied to the initial mesh on a  $1 \times 1$  square, Fig. 4.1a, and the process of vertex insertion/deletion and vertex movements are repeated six times. The meshes produced after second, fourth and sixth refinements are shown in Fig. 4.5. A closer look at the final mesh is depicted in Fig. 4.6 showing that the mesh achieved the desired anisotropy and structure. As demonstrated in Fig. 4.6a,



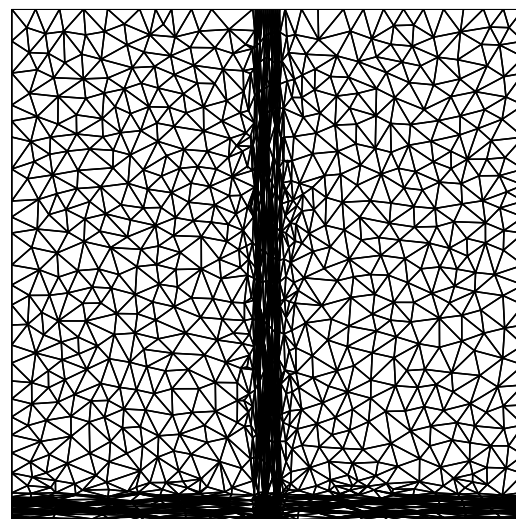
(a) Initial isotropic mesh (37 vertices)



(b) Second refinement (366 vertices)

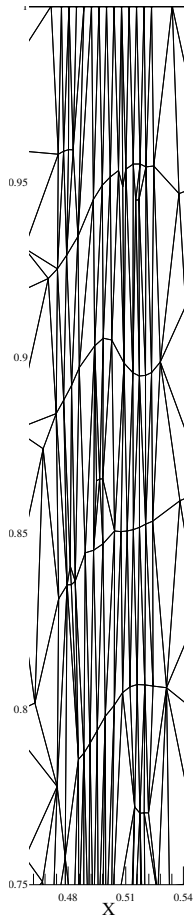


(c) Fourth refinement (883 vertices)

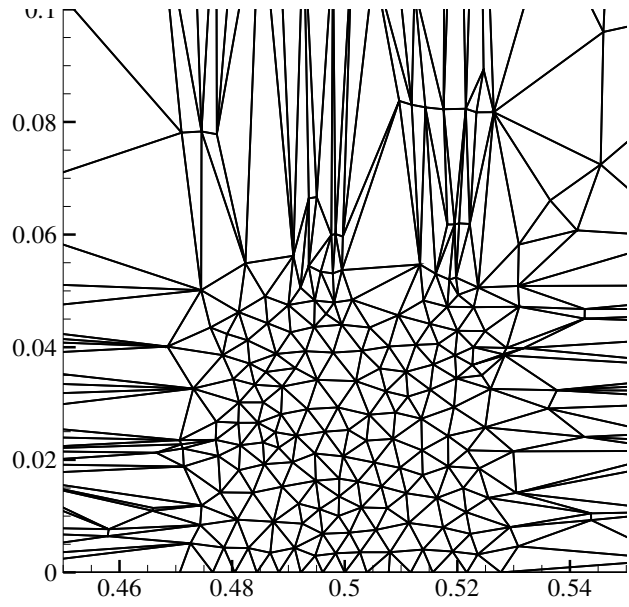


(d) Sixth refinement (1537 vertices)

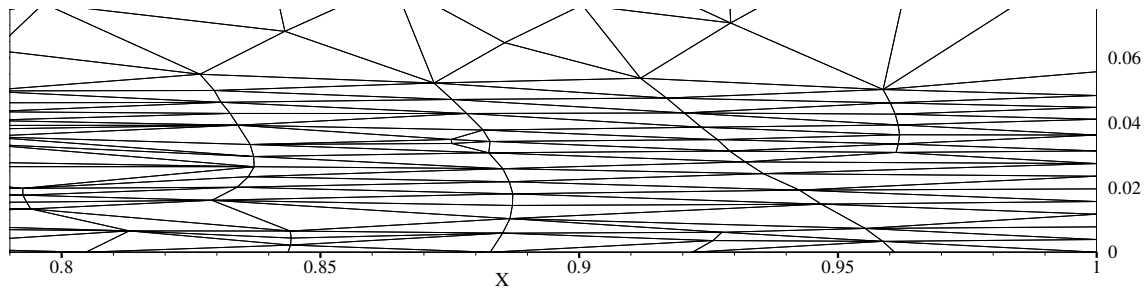
Figure 4.5: A mesh on a  $1 \times 1$  square being recursively refined



(a) Stretched mesh in the y-direction for  $0.475 < x < 0.525$



(b) Isotropic mesh for  $0.475 < x < 0.525$  and  $0 < y < 0.05$



(c) Stretched mesh in the x-direction for  $0 < y < 0.05$

Figure 4.6: Closeup view of the mesh in Fig. 4.5d

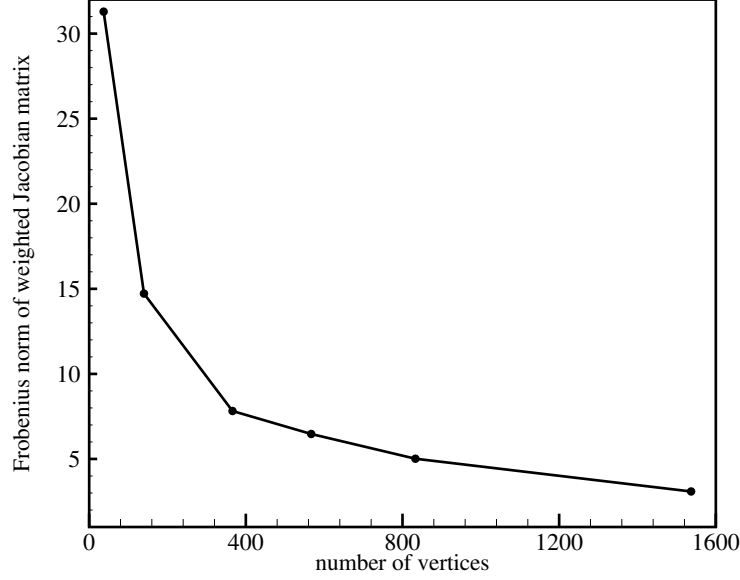


Figure 4.7: Quality improvement for the mesh on a  $1 \times 1$  square being recursively refined

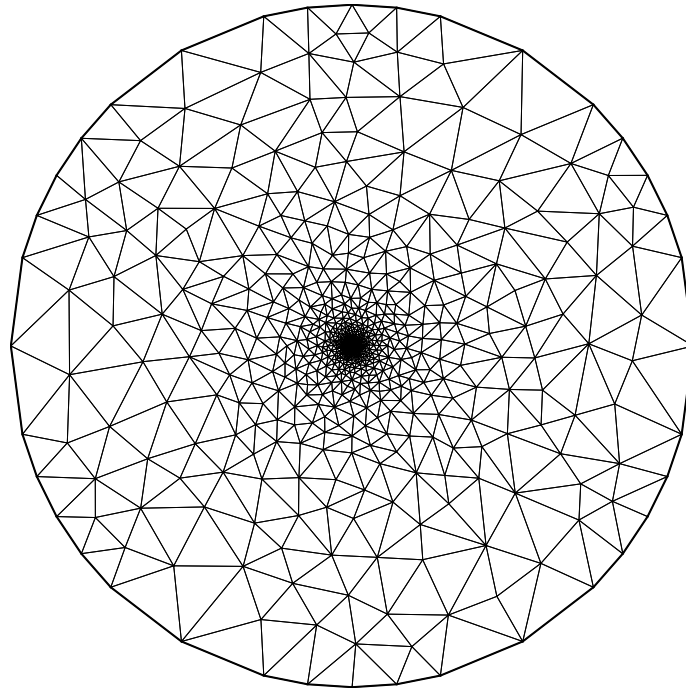
the elements in  $0.475 \leq x < 0.525$  are aligned (quasi-structured) and stretched in the  $y$ -direction (anisotropic). Also mesh elements in  $0 < y < 0.05$  are aligned and stretched in the  $x$ -direction as shown in Fig. 4.6c; moreover, mesh elements near the point  $(0.5, 0.05)$  are isotropic as shown in Fig. 4.6b.

The same mesh quality measure as defined in Section 4.1 is used to show the quality improvements by applying the adaptation/smoothing algorithm. The average Frobenius norm of the weighted Jacobian matrix over all elements are shown in Fig. 4.7. As depicted in this figure, the mesh quality asymptotically goes to  $\sqrt{2}$ , the ideal value.

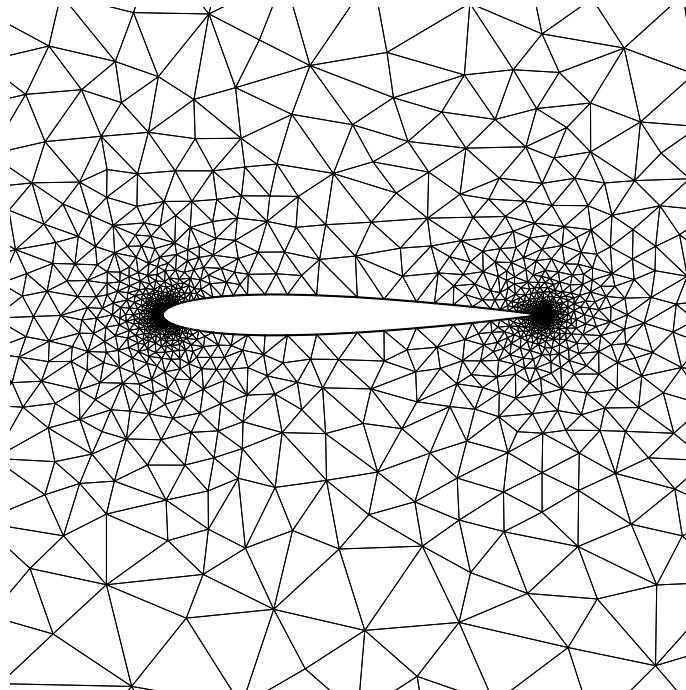
This example demonstrates the ability of the proposed algorithm to create anisotropic quasi-structured meshes from a predefined metric.

### 4.3 Test Cases

Two test case are presented here to demonstrate the effect of using the proposed algorithm. All solutions are computed using a second order vertex-centered finite volume solver described in [51]. This solver uses least square  $L_2$  reconstruction [52], Roe's scheme [62], and



(a) Farfield view



(b) Closeup view

Figure 4.8: Initial mesh with 2543 vertices around the NACA 0012 airfoil

Newton-GMRES [43, 45] for rapid convergence. Viscous terms are discretized as described in [52]. The initial mesh is constructed using the isotropic mesh generator in GRUMMP [49] and is refined by using anisotropic adaptation based on metrics [54]. To calculate the metric, the reconstruction error metric proposed in [56] is used.

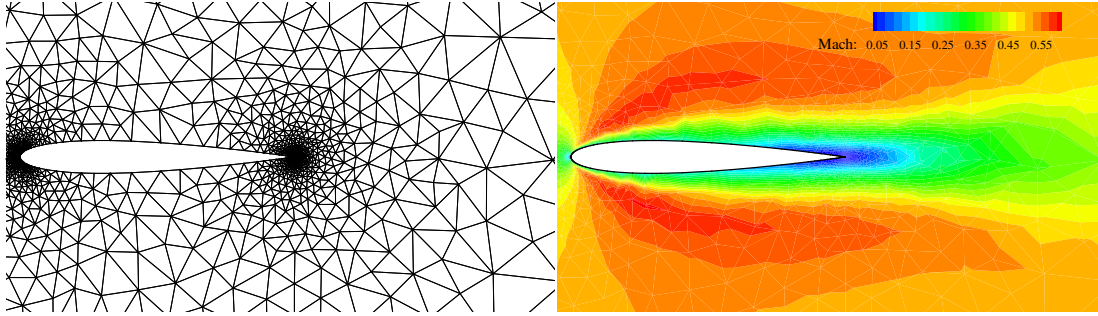
For all cases, the mesh shown in Fig. 4.8a is used as the initial mesh. This mesh is the Delaunay triangulation produced by GRUMMP and the refinement is done based only on boundary curvature. A closeup view of the initial mesh around the airfoil is depicted in 4.8b which shows that this mesh is ill-suited to compute viscous flow or transonic inviscid flow around the airfoil. The farfield boundary is a circle of radius 500 chords centered at the leading edge of the the airfoil and consequently as a result of large boundary, the errors arising from boundary placement can be neglected.

These two test cases, subsonic viscous flow with boundary layer and transonic inviscid flow with shocks, are classical problems in aerodynamics that can be used to show how successful an algorithm is at generating the desired mesh, as these two flows demonstrate two different physical behaviors that commonly occur in aerodynamics: a boundary layer, which is an anisotropic physical behavior parallel to the flow, and a shock, which is an anisotropic behavior at a large angle to the flow direction. If the method is successful in generating quasi-structured meshes match these two anisotropies, it should also be successful in other test cases.

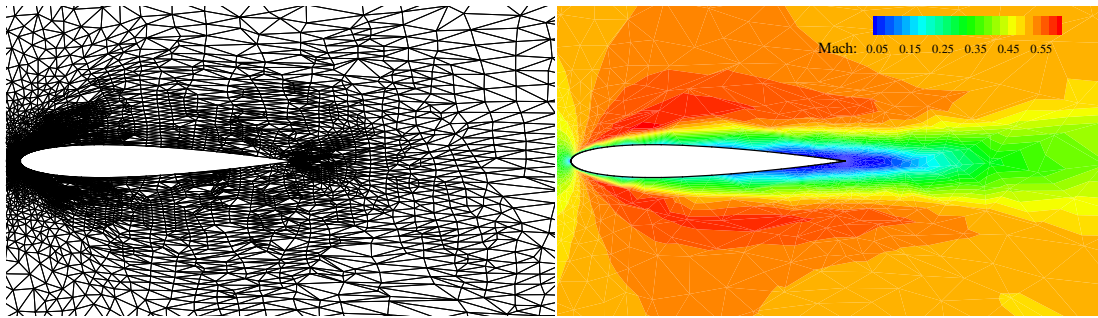
### 4.3.1 Subsonic viscous flow around an airfoil

The first test case is subsonic viscous flow over the NACA-0012 airfoil to demonstrate the advantages of using anisotropic quasi-structured mesh. The subsonic flow  $Ma = 0.5$  over the airfoil with  $Re = 5000$  and an angle of attack of  $\alpha = 0^\circ$  is examined. The resulting solution on the initial mesh is depicted in Fig. 4.9a and shows how poor the mesh is for viscous flow. A metric is calculated based on this solution and used for creating a new mesh. The solution accuracy improvement by repeating this process, using this metric after adaptation and smoothing as shown in Fig. 4.9b and 4.9c for the intermediate solutions. The solution on the final mesh, Fig. 4.9d, indicates the success of our algorithm. The resolution of the boundary layer and wake has been improved on the final mesh in Fig. 4.10 and illustrates how adaptation and smoothing improve the solution accuracy. Small isotropic elements at the leading edge of the airfoil, Fig. 4.10b, quickly become aligned, and highly anisotropic elements appear as the boundary layer develops as shown in Fig. 4.10d. The aligned elements shown in this figure, in comparison with the non-smoothed mesh show

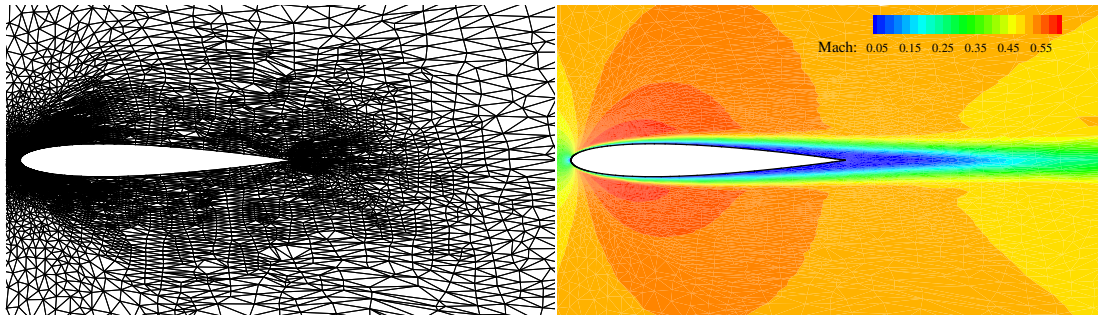




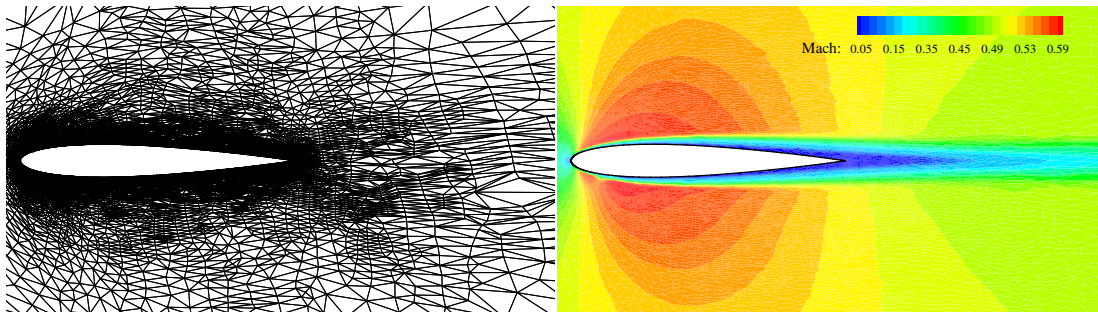
(a) Solution on the initial mesh (2543 vertices)



(b) Solution after first refinement (3899 vertices)



(c) Solution after second refinement (9140 vertices)



(d) Solution on the final mesh (19964 vertices)

Figure 4.9: Meshes and solutions for subsonic viscous flow over NACA 0012,  $Ma = 0.5$ ,  $Re = 5000$ ,  $\alpha = 0^\circ$

	$C_{D,P}$	$C_{D,v}$	$C_L$	$\frac{x_{Seperation}}{x_{chord}}$
Initial Mesh (2543 vertices)	0.610815	0.292588	0.015074	—
Smoothed Mesh, 3899 vertices	0.557115	0.206715	0.009244	0.823659
Smoothed Mesh, 9140 vertices	0.022429	0.034126	0.000203	0.817633
Smoothed Mesh, 19964 vertices	0.021383	0.033847	0.000108	0.814173
Non-smoothed Mesh, 3885 vertices	0.485876	0.172378	0.005786	0.850845
Non-smoothed Mesh, 7645 vertices	0.129702	0.037979	0.021304	0.848619
Non-smoothed Mesh, 17386 vertices	0.056125	0.036432	0.006265	0.844942
ARC2D (320×128)	0.0221	0.0321	—	0.824
Mavriplis (320×64) [42]	0.0229	0.0332	—	0.814
Radespiel (512×128)[59]	0.0224	0.0330	—	0.814
4 <sup>th</sup> order Spectral Volume [29]	0.022267	0.032405	—	0.813

Table 4.4: Comparison of drag and lift for NACA 0012 airfoil:  $Ma = 0.5$ ,  $Re = 5000$ ,  $\alpha = 0^\circ$

that smoothed mesh is quasi-structured. The non-smoothed mesh and the corresponding results for it, shown here as the comparison, are obtained by using the anisotropic mesh adaptation scheme developed by Pagnutti[54].

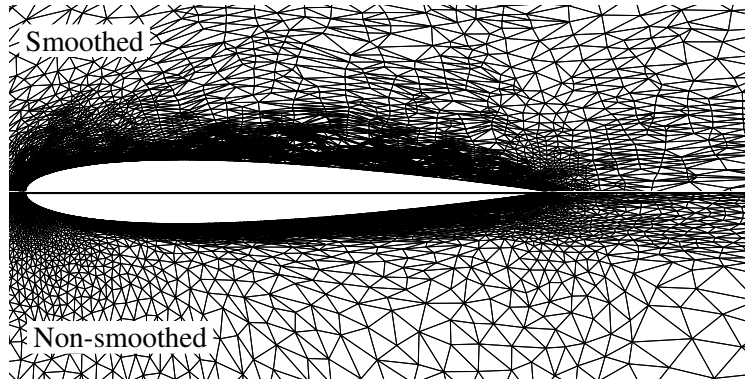
The drag and lift convergence plots are depicted in Fig. 4.11 and Fig. 4.12 respectively, illustrate that the solution is converging faster with smoothed mesh rather than simple anisotropic refinement, non-smoothed mesh. The solution is converging to a single value, for both pressure and viscous drag and also lift coefficient.

As this is a very common problem, there are many published results for this test case. The adapted mesh obtained by Yano and Darmofal [41] is shown in Fig. 4.13 for the same Mach number and Reynolds number.

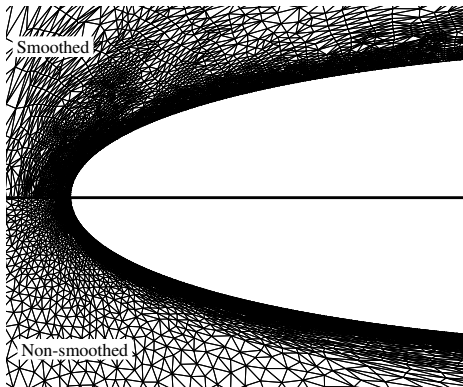
Table 4.4 is the summary of some of these published results and as shown in this table, the solution on the finest smoothed mesh has drag coefficients, both viscous and pressure drag, and separation point which lie within the acceptable range of these published results.

Pressure distribution on upper and lower surfaces of the airfoil for both smoothed and non-smoothed meshes are depicted in the left side of Fig. 4.14. Ideally, for zero angle of attack, the pressure distribution diagram should be symmetric along the airfoil, i.e. the pressure coefficient  $C_P$  is the same on these surfaces; however as a result of non-symmetric mesh, numerical results are not expected to be the same for these edges. Closer look at the pressure distribution on the right side of Fig. 4.14 illustrates that the solution is more symmetrical on a smoothed mesh in comparison with non-smoothed mesh.

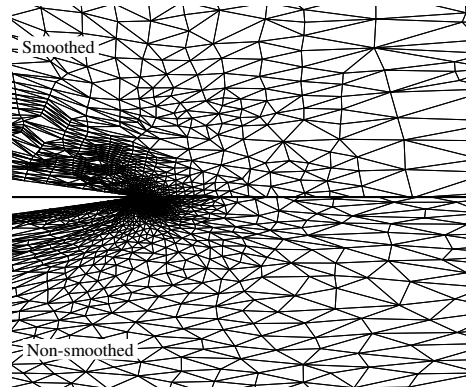
The wall shear stress has the same behavior for the top and bottom edges of the airfoil.



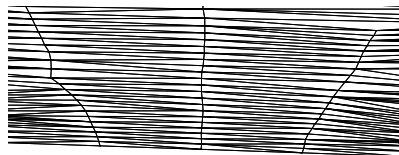
(a) Mesh plot (top half: smoothed; bottom half: non-smoothed)



(b) Leading edge (top half: smoothed; bottom half: non-smoothed)



(c) Trailing edge (top half: smoothed; bottom half: non-smoothed)



(d) Boundary layer (top half: smoothed; bottom half: non-smoothed)

Figure 4.10: Comparison between smoothed and non-smoothed mesh, viscous subsonic flow around NACA 0012

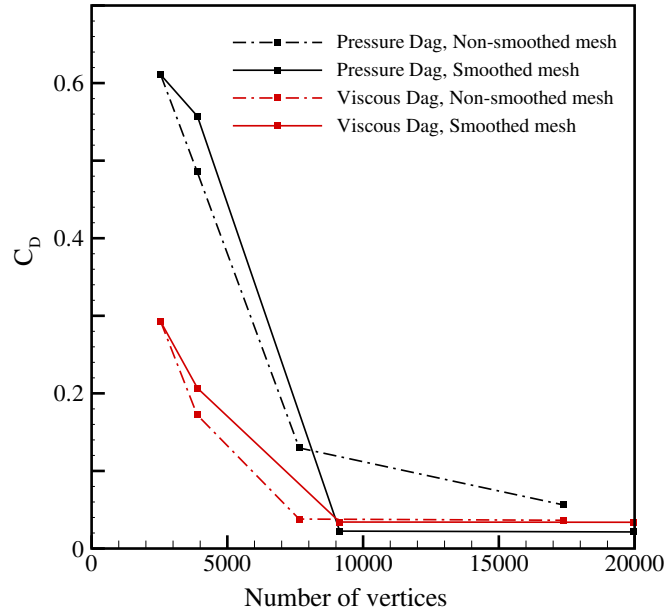


Figure 4.11: Drag convergence for NACA 0012,  $Ma = 0.5$ ,  $Re = 5000$ ,  $\alpha = 0^\circ$

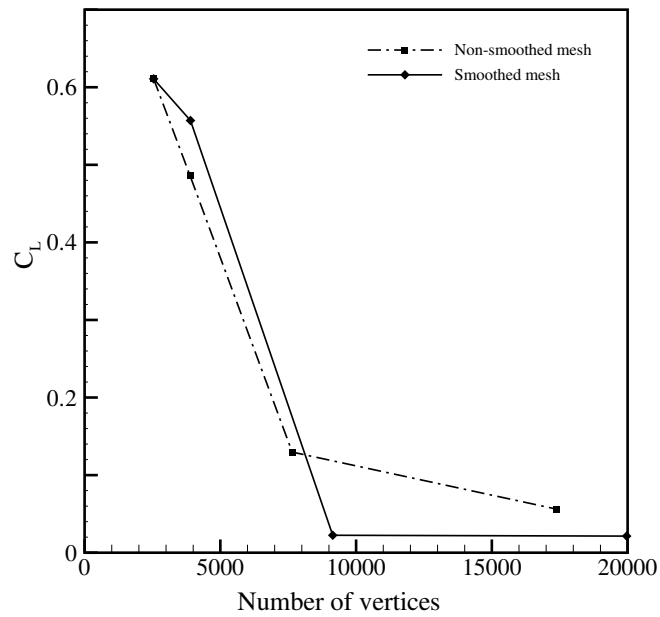


Figure 4.12: Lift convergence for NACA 0012,  $Ma = 0.5$ ,  $Re = 5000$ ,  $\alpha = 0^\circ$

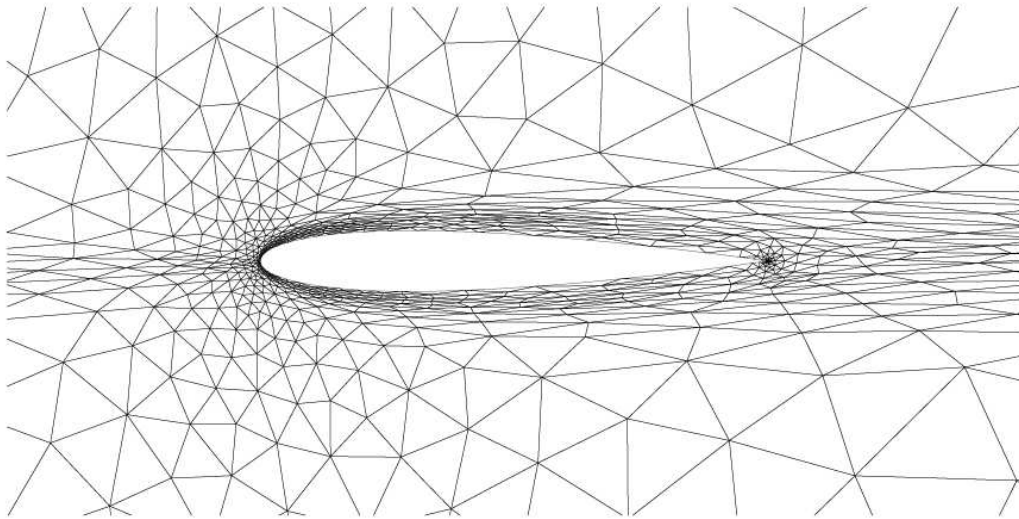


Figure 4.13: Selected adapted mesh for the subsonic laminar problem. Figure courtesy of Yano and Darmofal [41].

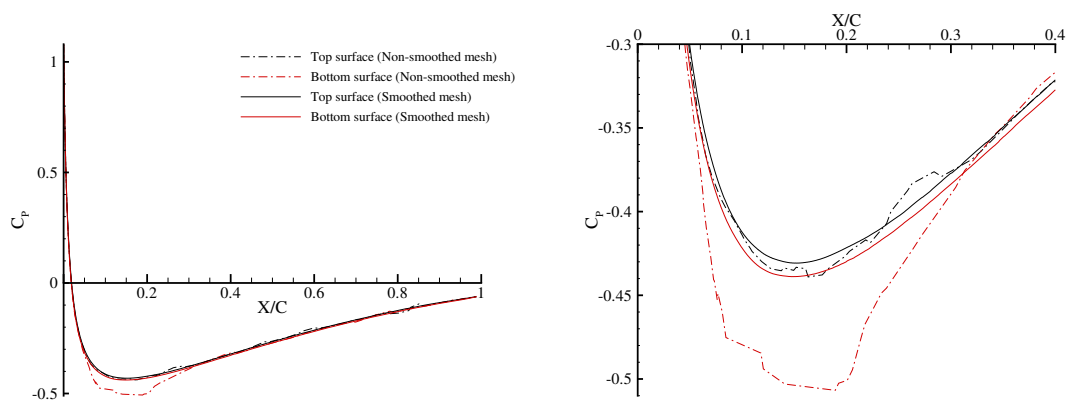


Figure 4.14: Pressure coefficient distribution for NACA 0012,  $Ma = 0.5$ ,  $Re = 5000$ ,  $\alpha = 0^\circ$

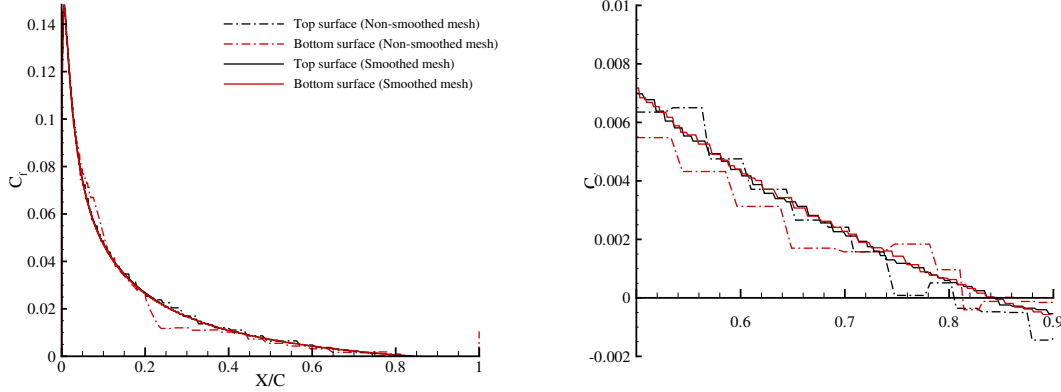


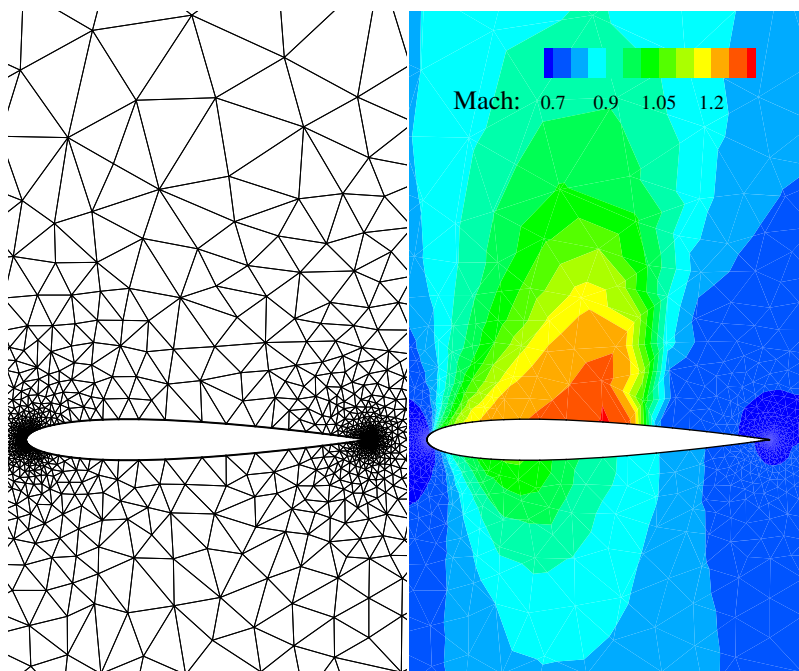
Figure 4.15: Wall shear stress distribution for NACA 0012,  $Ma = 0.5$ ,  $Re = 5000$ ,  $\alpha = 0^\circ$

As depicted in Fig. 4.15, skin friction coefficient  $C_f$  obtained for the smoothed mesh are closer for upper and lower surfaces in comparison with the non-smoothed mesh.

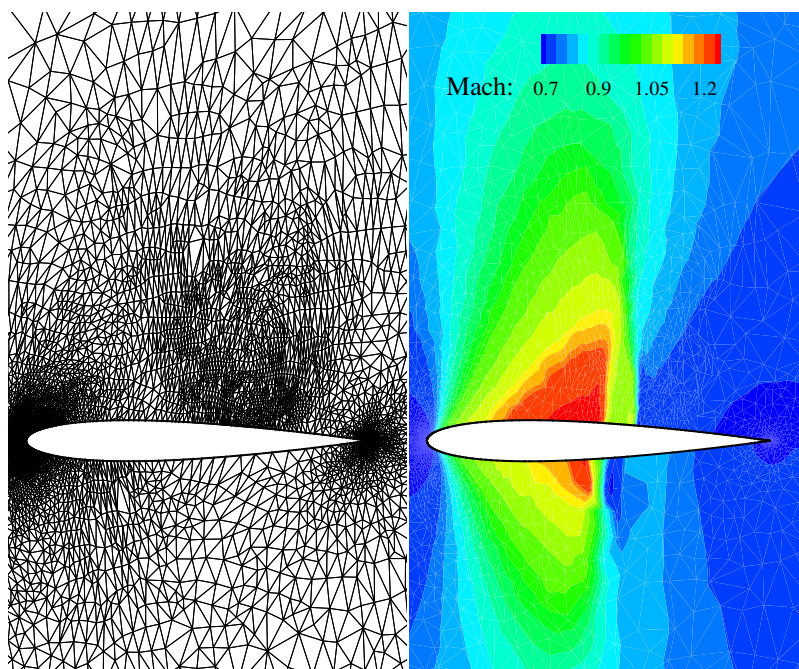
#### 4.3.2 Transonic inviscid flow around an airfoil

The second test case is transonic inviscid flow over the NACA-0012 airfoil where discontinuities (shocks) occur. This test case demonstrates the advantages of using an anisotropic quasi-structured mesh for another flow with anisotropic behavior. The transonic flow at  $Ma = 0.8$  over the airfoil with an angle of attack of  $\alpha = 1.25^\circ$  is examined. As a result of non-symmetric flow, we expect a strong shock on the suction (upper) side of the airfoil and a much weaker shock on the pressure (lower) side. The resulting solution on the initial mesh is depicted in Fig. 4.16a and shows how poor the mesh is for transonic flow. A metric is calculated based on this solution and used for creating a new mesh. The solution accuracy improves by repeating this process, using this metric after adaptation and smoothing as shown in Fig. 4.16b and 4.16c for the intermediate solutions. The solution on the final mesh, Fig. 4.16d, indicates the robustness of our algorithm. The resolution of shock and contact region has been improved on the final mesh in Fig. 4.17 and illustrates how adaptation and smoothing improve the solution accuracy. The sharp shock shown in Fig. 4.16d is taken by using the aligned elements in shock region demonstrated in Fig. 4.17b for the strong shock at the top surface and and Fig. 4.17c for the weak shock at the bottom surface of the airfoil. As demonstrated in these figures, the smoothed mesh is adapted such that the elements in these regions are more anisotropic in comparison with the non-smoothed

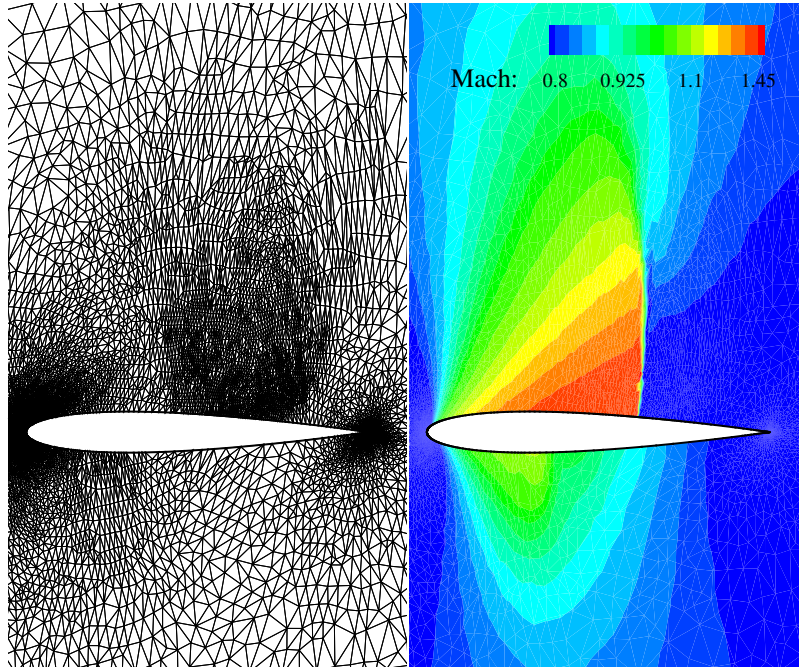




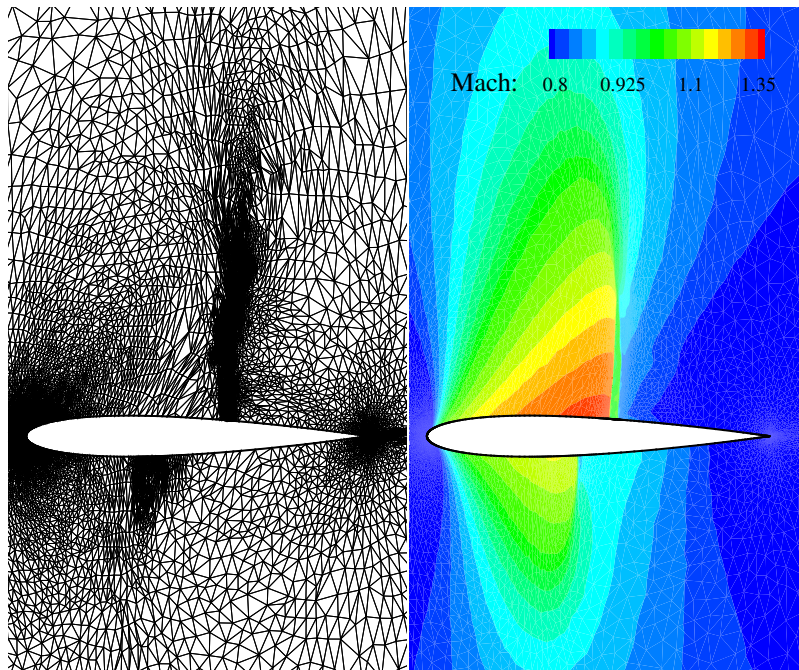
(a) Solution on the initial mesh (2543 vertices)



(b) Solution on first refinement (3816 vertices)



(c) Solution on fifth iteration (13348 vertices)



(d) Solution on the final mesh (19381 vertices)

Figure 4.16: Meshes and solutions for transonic inviscid flow over NACA 0012,  $Ma = 0.8$ ,  $\alpha = 1.25^\circ$



	$C_D$	$C_L$
Initial Mesh (2543 vertices)	0.0234527	0.255206
Smoothed Mesh, 3816 vertices	0.0233181	0.295698
Smoothed Mesh, 8751 vertices	0.0229092	0.318908
Smoothed Mesh, 19381 vertices	0.0224820	0.325311
Non-smoothed Mesh, 3842 vertices	0.0217434	0.239249
Non-smoothed Mesh, 10232 vertices	0.0161220	0.253453
Non-smoothed Mesh, 19995 vertices	0.0160930	0.279395
Yano and Darmofal model [41]	0.022628	0.35169
Barter model [7]	–	0.3418
AGARD-211 [1]	0.0221	0.3474

Table 4.5: Comparison of drag and lift for NACA 0012 airfoil:  $Ma = 0.8$ ,  $\alpha = 1.25^\circ$

mesh and consequently can predict the discontinuities with higher resolution. It should be noted that, for approximately the same number of mesh points, 19381 vertices for the smoothed mesh and 19995 vertices for the non-smoothed one, and four steps of refinement, the shock region in the smoothed mesh is sharper than the non-smoothed one significantly. Similar to the previous test case, the non-smoothed mesh and the corresponding results shown here are obtained by applying the anisotropic mesh adaptation scheme developed by Pagnutti.

Moreover, the contact region right after the trailing edge has more anisotropic elements in the smoothed mesh rather than the non-smoothed mesh as shown in Fig. 4.17d.

The computed shock on the smoothed mesh has a little overshoot and is sharper in comparison with non-smoothed mesh as shown in Fig. 4.18 and 4.19 for pressure distribution and Mach number, respectively on the top surface of the airfoil.

The pressure drag and lift convergence plots are depicted in Fig. 4.20 and Fig. 4.21 respectively, illustrate that the solution convergence is faster with smoothed mesh rather than anisotropic refinement and the solution is converging to a single value, for both drag and also lift coefficients. As mentioned before, this is a classic problem and was solved several times as shown by different models in Table. 4.5. The mesh obtained by one of these methods, Yano and Darmofal model, is shown in Fig. 4.22 for the adapted mesh for transonic inviscid problem with the same Mach number and angle of attack.

Comparison between drag and lift coefficients for different mesh sizes and published results are depicted in this table and our results for the finest smoothed mesh lie within the acceptable range of these published results.

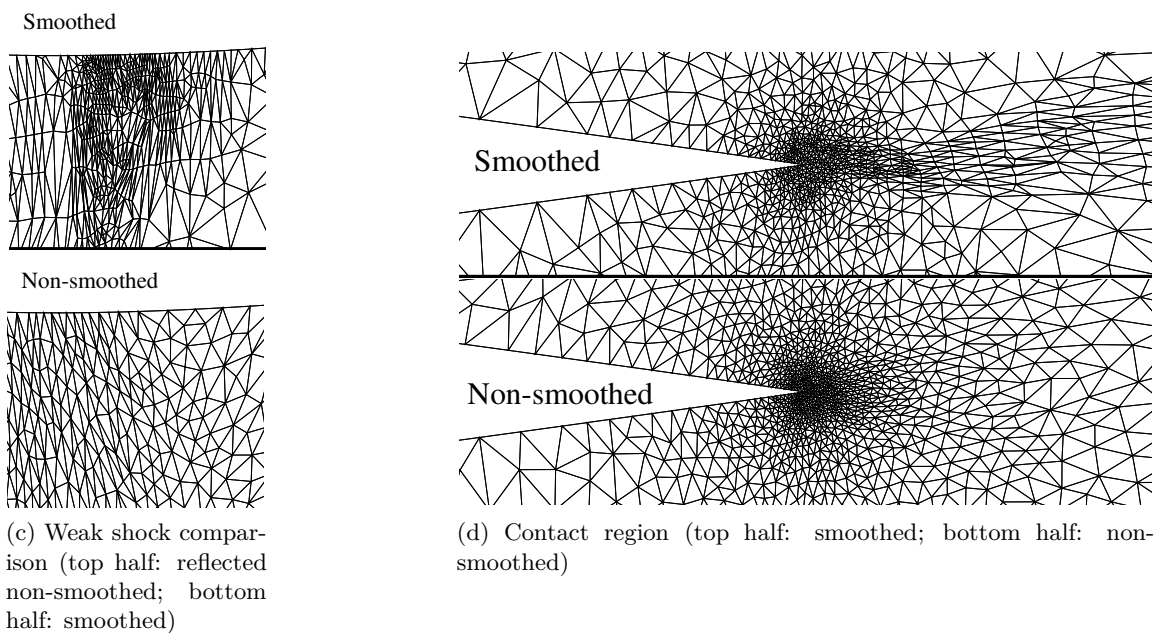
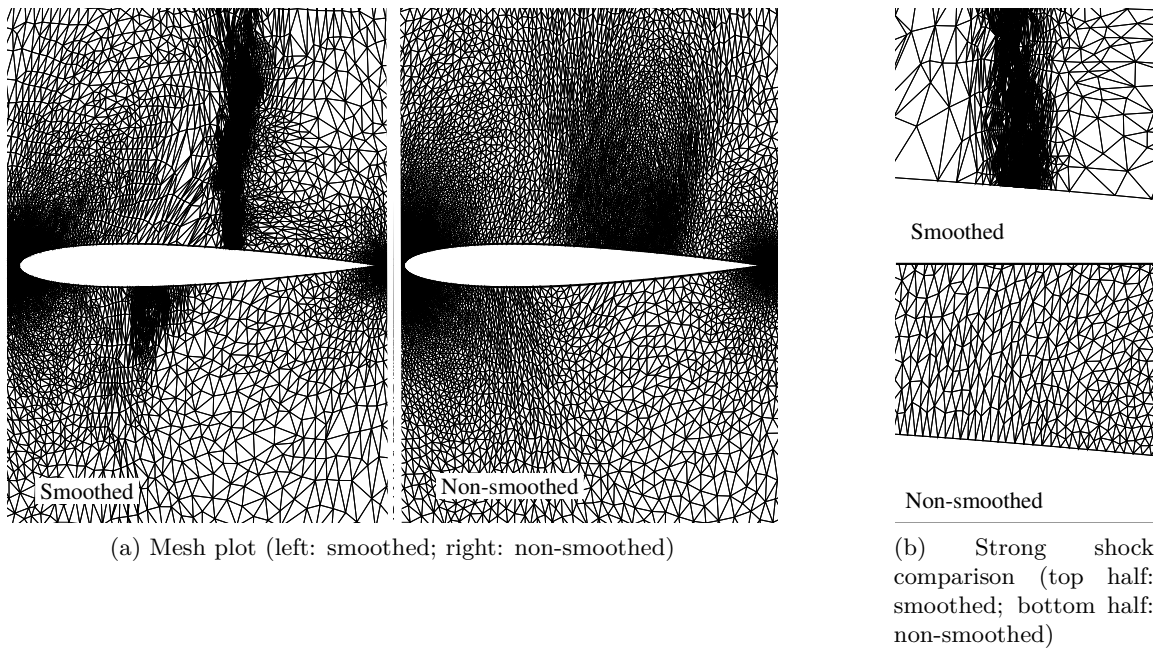


Figure 4.17: Comparison between smoothed and non-smoothed mesh, inviscid transonic flow around NACA 0012

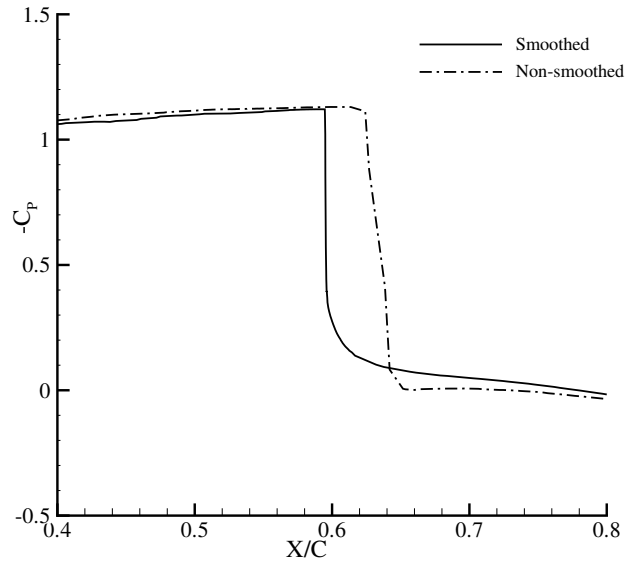


Figure 4.18: Pressure coefficient distribution for NACA 0012,  $Ma = 0.8$ ,  $\alpha = 1.25^\circ$

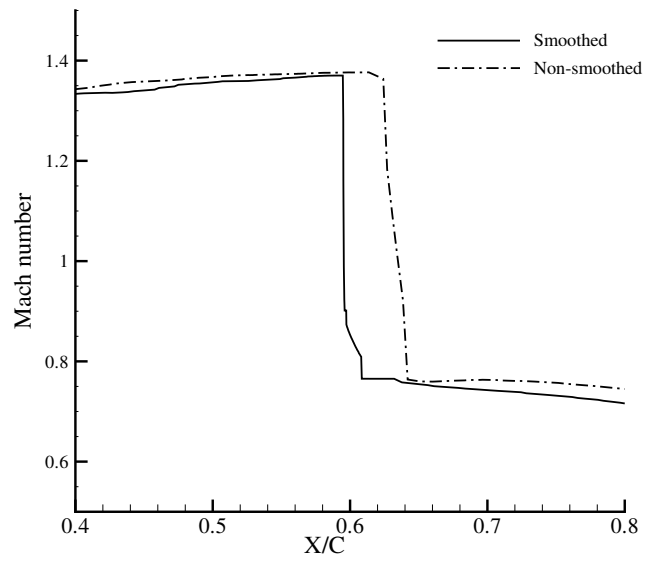


Figure 4.19: Mach number distribution for NACA 0012,  $Ma = 0.8$ ,  $\alpha = 1.25^\circ$

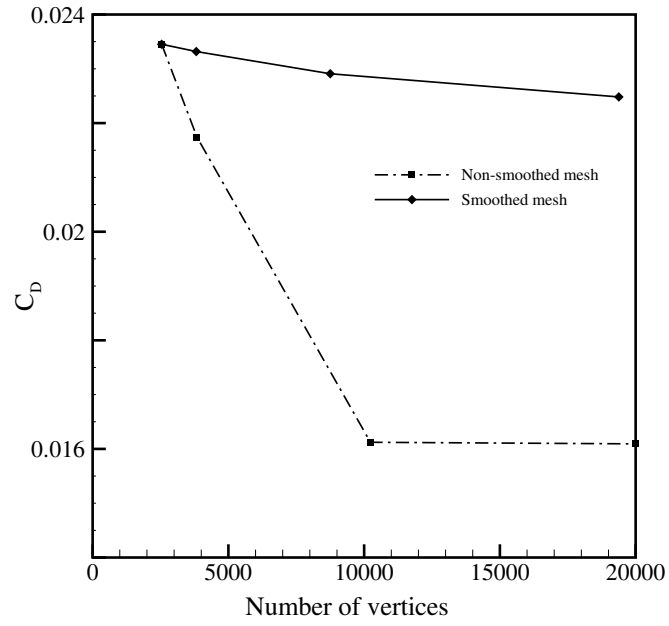


Figure 4.20: Drag convergence for NACA 0012,  $Ma = 0.8$ ,  $\alpha = 1.25^\circ$

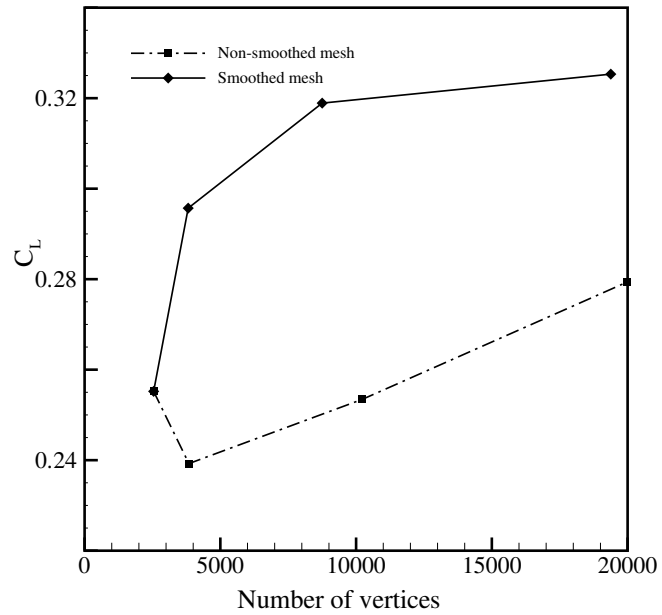


Figure 4.21: Lift convergence for NACA 0012,  $Ma = 0.8$ ,  $\alpha = 1.25^\circ$

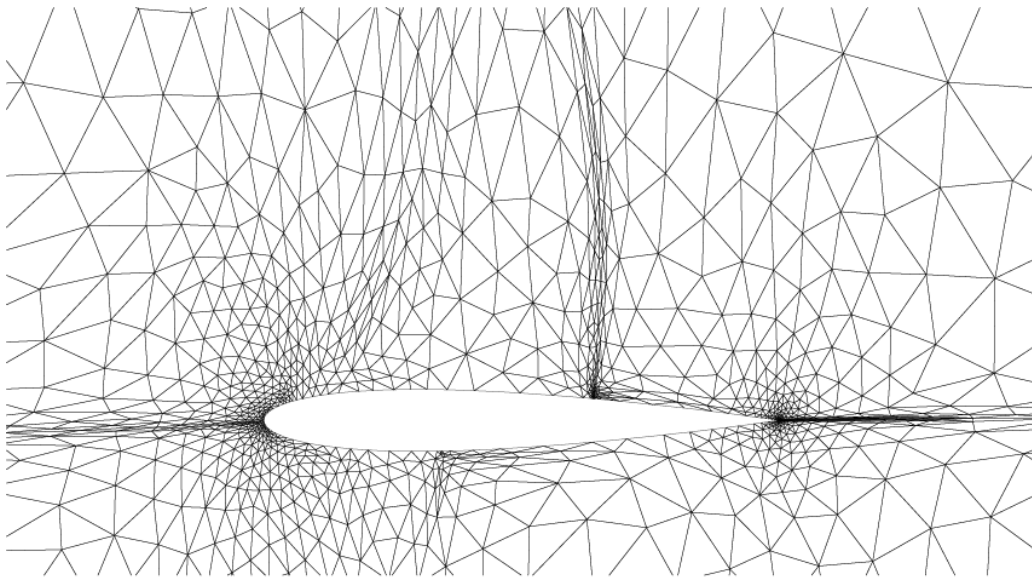


Figure 4.22: Selected adapted mesh for the transonic inviscid problem. Figure courtesy of Yano and Darmofal [41].

## Chapter 5

# Concluding Remarks

### 5.1 Summary

An existing anisotropic adaptation scheme, which produces meshes without any regular structure [54], has been modified in this research to produce quasi-structured meshes where they are needed the most. Anisotropic meshes are those meshes containing long and thin elements that have been stretched according to a preferred direction for which the numerical gradients are smaller. A previous mesh adaptation scheme capable of producing highly anisotropic meshes, which matches the physical behavior of the flow, but without any regular structure, has been modified so that aligned, quasi-structured meshes are constructed which means that both the accuracy and efficiency of the flow solution are improved. These final smoothed meshes have quasi-structured elements with regular structure in the anisotropic parts of the domain such that mesh elements have structure and are almost parallel to the body surface in boundary layers in viscous flows and parallel to the shocks in transonic flows. By using this technique, we can get the advantages of flexibility of unstructured meshes for complex geometries and accuracy of the high directional qualities of the structured meshes simultaneously. In mesh adaptation in which the mesh is locally fit to the particular features of the flow, the solution is computed on an initial coarse mesh and then the mesh is successively refined so that the optimal mesh for the solution being computed is produced.

A solution-adaptive approach needs a formulation of the cell-modification parameters from local error estimates and the solution approximation error estimation method has been used in this thesis. The cell modification parameters which define the desired anisotropy are communicated to the meshing program through the metrics. For two-dimensional problems, the metric assigned to each vertex of the mesh is represented by a  $2 \times 2$  matrix. The geometrical interpretation of the metric is the distance and angles measured from that point.

An adaptive meshing algorithm based on Pagnutti's thesis [54] uses the metric to con-

struct an improved anisotropic mesh with higher quality. Mesh quality refers to geometric properties of a mesh, such as local mesh size, orientation and local shape. Among several quality measures that have been proposed based on dimensionless ratio of various geometric parameters, the norm of the triangle matrix, which contains information about both aspect ratio and orientation of the mesh, has been chosen since the final goal of this research is to produce highly anisotropic quasi-structured meshes.

Four mesh improvement operations are applied to the initial coarse mesh with the aim of improving mesh quality: 1) face swapping which chooses the best diagonal for the quadrilateral formed by two neighboring triangles, 2) vertex insertion for large elements, 3) vertex removal for small edges and 4) vertex movements for optimizing the vertex location. For the optimization process, the Mesquite package has been used so that the final mesh is the aligned, quasi-structured mesh. The target mesh element is defined based on the solution-based metric and is used for mesh optimization.

These mesh improvement techniques are applied to the initial coarse mesh with the following sequence: initial smoothing and swapping, repeated vertex insertion and deletion, and final smoothing and swapping step.

## 5.2 Conclusion

The ability of the proposed algorithm to create quasi-structured anisotropic meshes has been shown by creating a mesh for a pre-defined metric on a square domain. The final mesh obtained by applying our algorithm to the initial isotropic mesh contains elements which are properly aligned, quasi-structured and shows the robustness of our algorithm.

Two test case have been examined to demonstrate the benefits of the anisotropic mesh adaptation with the aim of producing quasi-structured meshes. For both test cases, subsonic viscous and transonic inviscid flow around the NACA-0012 airfoil, the flow has anisotropic physical behavior. The presence of boundary layers in the viscous flow and shock waves in the transonic flow, leads to highly anisotropic behavior which can be captured more accurately by using anisotropic meshes. Smoothed meshes are obtained by using our algorithm which contains quasi-structured meshes parallel to the airfoil boundary, stretched in the direction of the boundary in the viscous flow, and quasi-structured meshes parallel to the shock wave, stretched in the direction of the discontinuity in the case of the transonic inviscid flow.

Comparison of the non-smoothed meshes with smoothed, quasi-structured meshes, shows

that using smoothed elements can dramatically increase solution efficiency and accuracy for approximately the same number of mesh point, even more points for the case of non-smoothed mesh. In all the cases examined, for the finest non-smoothed meshes that we used, some of the flow features are insufficiently resolved.

For every mesh that we adapted, the proposed algorithm was able to create high quality quasi-structured anisotropic mesh and consequently, different flow features such as boundary layer and shock waves are captured more accurately.

### 5.3 Recommendations

The proposed method has been developed for anisotropic adaptation with the aim of producing quasi-structured meshes that can be applied to any Taylor-based CFD algorithm, however, this approach was tested for two-dimensional finite volume, compressible flow around the NACA-0012 airfoil. By modifying quality improvement operations for 3-D meshes, this approach can be extended to three dimensional problems. Although there are some limitations for three-dimensional anisotropic meshing, examining the effect of our approach for higher-order 3-D solutions is an item for future work.

Another potential improvement is using adjoint-based error estimation method instead of solution approximation method. Venditti and Darmofal [70] demonstrated that adjoint-based methods provide greater accuracy. Combining adjoint-based adaptation with our quasi-structured mesh recovery seems an intuitive extension.

One problem with our method is that there are still some poor-shaped elements in regions where the flow behavior is highly anisotropic. These low quality elements include some isotropic ones for which vertex removal is required and some elements with long edges for which vertex insertion is required. Those bizarre spots should be fixed ultimately.

Another problem that should be fixed is that although our method predict drag and lift coefficients of the transonic case accurately and a very sharp shock is captured, the location where the shock is captured is a bit different from the shock location in published results. Figuring out the reason why this is not predicted accurately is another potential question.

The goal of this work is to improve solution accuracy by generating a higher-quality mesh with a minimum amount of input from the user. While more accurate solutions are obtained by our approach, it is not clear whether or not this method is faster. One obvious point is that the optimization part for vertex movements takes the most time of refinement for which time saving would improve the total time of refinement significantly.



Hopefully, as further research is done into improving high-order CFD solvers and anisotropic, quasi-structured mesh generators, the advantages of combining the two will significantly increase.

# Bibliography

- [1] Advisory Group for Aerospace Research and Development. Technical report, Technical Report AR-211, Advisory Group for Aerospace Research and Development (AGARD), Technical Report AR-211, NATO., 1985.
- [2] D. Ait-Ali-Yahia, G. Baruzzi, W. G. Habashi, M. Fortin, J. Dompierre, and M. G. Vallet. Anisotropic mesh adaptation: Towards user-independent, mesh-independent and solver-independent CFD. Part II: Structured grids. *International Journal for Numerical Methods in Fluids*, 39(8):657–673, July 2002.
- [3] Thomas Apel. *Anisotropic Finite Elements: Local Estimates and Applications*. 1999.
- [4] R. Auby and R. Lohner. Generation of viscous grids at ridges and corners. *International Journal for Numerical Methods in Engineering*, 77:1247–1289, 2009.
- [5] Y. Liu B. Levy. Lp Centroidal Voronoi Tessellation and its Applications. *ACM Transactions on Graphics*, 29,4, 2010.
- [6] R. Balasubramanian and J.C. Newman III. Comparison of adjoint-based and feature-based grid adaptation for functional outputs. *International Journal for Numerical Methods in Fluids*, 53:1541–1569, 2007.
- [7] Garrett Ehud Barter. *Shock Capturing with PDE-Based Artificial Viscosity for an Adaptive, Higher-Order Discontinuous Galerkin Finite Element Method*. PhD thesis, Department of Aeronautics and Astronautics, MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 2008.
- [8] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica*, 2001:1–102, 2000.
- [9] Houman Borouchaki, Paul Louis George, and Bijan Mohammadi. Delaunay mesh generation governed by metric specifications. Part 2: Applications. *Finite Element Analysis and Design*, 25:85–109, 1997.

- [10] Frank J. Bossen and Paul S. Heckbert. A pliant method for anisotropic mesh generation. In *Proceedings of the Fifth International Meshing Roundtable*, pages 63–74. Sandia National Laboratories, October 1996.
- [11] C. L. Bottasso. Anisotropic mesh adaption by metric-driven optimization. *International Journal for Numerical Methods in Engineering*, 60, 2004.
- [12] Weiming Cao. An interpolation error estimate based on the anisotropic measures of higher order derivatives. *SIAM Journal of Scientific Computing Society for Industrial and Applied Mathematics*, 29, No. 2:756–781, 2007.
- [13] M.J. Castro-Diaz, F. Hecht, B. Mohammadi, and O. Pironneau. Anisotropic unstructured mesh adaptation for flow simulations. *International Journal for Numerical Methods in Fluids*, 25:475–291, 1997.
- [14] DESBRUN D. COHEN-STEINER, P. ALLIEZ. Variational shape approximation. *ACM Transactions on Graphics*, 23:905–914, 2004.
- [15] Eduardo F. D’Azevedo and R. Bruce Simpson. An optimal triangular meshes for minimizing the gradient error. *Numerische Mathematik*, 59:321–348, 1991.
- [16] Lori Freitag Diachin, Patrick Knupp, Todd Munson, and Suzanne Shontz. A comparison of two optimization methods for mesh quality improvement. *Engineering with Computers*, 22(2):61–74, 2006.
- [17] J. Dompierre, M. G. Vallet, Y. Bourgault, M. Fortin, and W. G. Habashi. Anisotropic mesh adaptation: Towards user-independent, mesh-independent and solver-independent CFD. Part III. Unstructured meshes. *International Journal for Numerical Methods in Fluids*, 39(8):675–702, July 2002.
- [18] A. Dvinsky. Adaptive grid generation from harmonic maps on Riemannian manifolds. *Journal of Computational Physics*, 95:450–476, 1991.
- [19] R.P. Dwight. Heuristic a posteriori estimation of error due to dissipation in finite volume schemes and application to mesh adaptation. *Journal of Computational Physics*, 227:2845–2863, 2008.
- [20] Sebastien Legrand Eric Delhez, Vincent Legat and Eric Deleersnijder. Unstructured, anisotropic mesh generation for the northwestern european continental shelf, the con-

- tinental slope and the neighbouring ocean. *Continental Shelf Research*, 27:1344–1356, 2007.
- [21] F. Mavriplis. CFD in aerospace in the new millenium. *Canadian Aeronautics and Space Journal*, 46(4):167–179, 2000.
- [22] L. Freitag, P. Knupp, T. Munson, and S. Shontz. A comparison of optimization software for mesh shape-quality improvement problems. *Proceedings of the Eleventh International Meshing Roundtable*, 2002.
- [23] Lori Freitag-Diachin, Patrick Knupp, and Jason Kraftcheck. Mesquite: Mesh Quality Improvement Toolkit User’s Guide. Technical report, Center for Computation, Computers, Information, and Mathematics, 2009.
- [24] M. B. Giles and N. A. Pierce. An introduction to the adjoint approach to design. *Flow, Turbulence, and Combustion*, 65(3–4):393–415, 2000.
- [25] Serge Gosselin. *Delaunay refinement mesh generation of curve-bounded domains*. PhD thesis, The University of British Columbia, Department of Mechanical Engineering, 2009.
- [26] W. G. Habashi, J. Dompierre, Y. Bourgault, D. Ait-Ali-Yahia, M. Fortin, and M. G. Vallet. Anisotropic mesh adaptation: Towards user-independent, mesh-independent and solver-independent CFD. Part I: General principles. *International Journal for Numerical Methods in Fluids*, 32(6):725–744, March 2000.
- [27] U. Hetmaniuk and P. Knupp. A mesh optimization algorithm to decrease the maximum error in finite element computations. *Proceedings of the Seventeenth International Meshing Roundtable*, pages 533–550, 2008.
- [28] C.R. Illingworth. Some solutions of the equations of flow of a viscous compressible fluid. *Proceedings of the Cambridge Philosophical Society*, 46:469–478, 1950.
- [29] Ravi Kannan and Zhijian Wang. Improving the High Order Spectral Volume Formulation Using a Diffusion Regulator. *Commun. Comput. Phys*, 12, No. 1:247–260, 2012.
- [30] P. Knupp. Achieving finite element mesh quality via optimization of the jacobian matrix norm and associated quantities. Part II- a framework for volume mesh opti-

- mization and the condition number of the jacobian matrix. *International Journal for Numerical Methods in Engineering*, 48:1165–1185, 2000.
- [31] P. Knupp. Mesh quality improvement for SciDAC applications. 46:458–462, 2002.
  - [32] P. Knupp and N. Robidoux. A framework for variational grid generation: Conditioning the jacobian matrix with matrix norms. 21, No.6:2029–2047, 2000.
  - [33] Patrick Knupp. Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part I — A framework for surface mesh optimization. *International Journal for Numerical Methods in Engineering*, 48:1165–1185, 2000.
  - [34] Patrick Knupp. Algebraic mesh quality metrics. *SIAM Journal of Scientific Computing*, 23(1):133–149, 2001.
  - [35] Patrick Knupp. Introducing the target-matrix paradigm for mesh optimization via node-movement. *19th International Meshing Roundtable, Springer-Verlag*, pages 67–84, 2010.
  - [36] François Labelle and Jonathan Richard Shewchuk. Anisotropic Voronoi diagrams and guaranteed-quality anisotropic mesh generation. In *Proceedings of the Nineteenth Annual Symposium on Computational Geometry*, pages 191–200. Association for Computing Machinery, 2003.
  - [37] L.Chen and J.Xu. Optimal Delaunay triangulations. *Journal of Computational Mathematics*, 22:299–308, 2004.
  - [38] V. Liseikin. Computational differential geometry approach to grid generation. *Springer, Heidelberg*, 2004.
  - [39] Adrien Loseille and Rainald Lohner. Anisotropic adaptive simulations in aerodynamics. In *48th AIAA Aerospace Sciences Meeting*, 2010,.
  - [40] D. L. Marcum. Adaptive unstructured grid generation for viscous flow applications. *AIAA Journal*, 34(8):2440–2443, 1996.
  - [41] David L. Darmofal Masayuki Yano. Flow Over the NACA 0012 Airfoil: Subsonic Inviscid, Transonic Inviscid, and Subsonic Laminar Flows.

- [42] Dimitri Mavriplis and Antony Jameson. Multigrid solution of the Euler equations on unstructured and adaptive meshes. ICASE Report No. 87-53, NASA Langley Research Center, 1987.
- [43] Christopher Michalak and Carl Ollivier-Gooch. Matrix-explicit GMRES for a higher-order accurate inviscid compressible flow solver. In *Proceedings of the Eighteenth AIAA Computational Fluid Dynamics Conference*, 2007.
- [44] Gary L. Miller, Dafna Talmor, and Shang-Hua Teng. Optimal coarsening of unstructured meshes. *Journal of Algorithms*, 31(1):29–65, 1999.
- [45] Amir Nejat. *A Higher-Order Accurate Unstructured Finite Volume Newton-Krylov Algorithm for Inviscid Compressible Flows*. PhD thesis, The University of British Columbia, Department of Mechanical Engineering, 2007.
- [46] Marian Nemec, Michael J. Aftosmis, and Mathias Wintzer. Adjoint-based adaptive mesh refinement for complex geometries. In *Forty-sixth AIAA Aerospace Sciences Meeting*, 2008. AIAA 2008-725.
- [47] HOA NGUYEN. *CENTROIDAL VORONOI TESSELLATIONS FOR MESH GENERATION: FROM UNIFORM TO ANISOTROPIC ADAPTIVE TRIANGULATIONS*. PhD thesis, FLORIDA STATE UNIVERSITY, COLLEGE OF ARTS AND SCIENCES, 2008.
- [48] Carl Ollivier-Gooch. Grummp version 0.5.0 user’s guide. Technical report, Department of Mechanical Engineering, The University of British Columbia, 2010.
- [49] Carl F. Ollivier-Gooch. An unstructured mesh improvement toolkit with application to mesh improvement, generation and (de-)refinement. AIAA 98-0218, January 1998.
- [50] Carl F. Ollivier-Gooch. GRUMMP — Generation and Refinement of Unstructured, Mixed-element Meshes in Parallel. <http://tetra.mech.ubc.ca/GRUMMP>, 1998–2010.
- [51] Carl F. Ollivier-Gooch. ANSLib: A scientific computing toolkit supporting rapid numerical solution of practically any PDE. In *Proceedings of the Eighth Annual Conference of the Computational Fluid Dynamics Society of Canada*, pages 21–28. Société canadienne de CFD / CFD Society of Canada, June 2000.

- [52] Carl F. Ollivier-Gooch and Michael Van Altena. A high-order accurate unstructured mesh finite-volume scheme for the advection-diffusion equation. *Journal of Computational Physics*, 181(2):729–752, 2002.
- [53] O. DEVILLERS B. LEVY DESBRUN P. ALLIEZ, D.COHEN-STEINER. Anisotropic polygonal remeshing. *ACM Transactions on Graphics*, 22, 3:485–493, 2003.
- [54] Doug Pagnutti. Anisotropic adaptation: Metrics and meshes. Master’s thesis, The University of British Columbia, Department of Mechanical Engineering, 2008.
- [55] Doug Pagnutti and Carl Ollivier-Gooch. Delaunay-based anisotropic mesh refinement. In *Proceedings of the Seventeenth International Meshing Roundtable*, 2008.
- [56] Doug Pagnutti and Carl Ollivier-Gooch. A generalized framework for high order anisotropic mesh adaptation. *Computers and Structures*, 87:670–679, 2009.
- [57] Phillipe Pebay and Timothy J. Baker. Analysis of triangle quality measures. *Mathematics of Computation*, 72:1817–1839, 2003.
- [58] E. Pelizzari, V. Selmin, and A. Ghidoni. Fully anisotropic unstructured grid generation with applications to aircraft design. In *European Conference on Computational Fluid Dynamics*.
- [59] R. Radespiel. A cell-vertex multigrid method for the Navier-Stokes equations. NASA TM-101557, 1989.
- [60] Shmuel Rippa. Long and thin triangles can be good for linear interpolation. *SIAM Journal on Numerical Analysis*, 29(1):257–270, February 1992.
- [61] R.Lohner. Generation of unstructured grids suitable for RANS calculations. *AIAA Paper*, 99:3251, 1999.
- [62] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.
- [63] Christopher Roy. Strategies for driving mesh adaptation in CFD. In *Proceedings of the Forty-Seventh AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, 2009.

- [64] J. Ruppert. A new and simple algorithm for quality 2-dimensional mesh generation. In *Proceedings of the Fourth ACM-SIAM Symposium on Discrete Algorithms*, pages 83–92, 1993.
- [65] Jessica Schoen. Robust, guaranteed-quality anisotropic mesh generation. Master’s thesis, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley.
- [66] M. Shashkov, R. Garimella, and P. Knupp. Optimization of surface mesh quality using local parameterization. *Proceedings of the 11th International Meshing Roundtable, Ithaca NY*, pages 41–52, 2002.
- [67] Jonathan Shewchuk. What is a good linear element? Interpolation, conditioning, and quality measures. In *Proceedings of the Eleventh International Meshing Roundtable*, 2002.
- [68] Jonathan R. Shewchuk. *Delaunay Refinement Mesh Generation*. PhD thesis, School of Computer Science, Carnegie Mellon University, May 1997.
- [69] J. Thompson, Z. Warsi, and C. Mastin. Automatic numerical generation of bodyfitted curvilinear coordinate systems. *Journal of Computational Physics*, 24:274–302, 1977.
- [70] D. A. Venditti and D. L. Darmofal. Grid adaptation for functional outputs: Application to two-dimensional inviscid flows. *Journal of Computational Physics*, 175(1):40–69, February 2002.
- [71] Nigel P. Weatherill, Joe F. Thompson, and Bharat K. Soni, editors. *Handbook of Grid Generation*. CRC Press, 1999.