National Technical University of Athens
School of Mechanical Engineering
Fluids Section
Parallel CFD & Optimization Unit

Curvature Computation on Surface Meshes for Adjoint-Based Constrained Optimization

Diploma Thesis

Anna Chotzalli

Advisor: Kyriakos C. Giannakoglou, Professor NTUA

Athens, 2025

Acknowledgments

This work would not have become a reality without the invaluable support of those around me. First and foremost, I would like to express my deepest gratitude to my supervisor and mentor, Professor Kyriakos C. Giannakoglou, for his constant guidance and attention throughout the entire duration of my diploma thesis. His academic and moral support during my undergraduate studies, as well as his scientific insight and inspiration, have been instrumental to my development. His extensive knowledge and experience greatly facilitated my research process and provided the ideal foundation for my journey into the world of research.

Special thanks go to all the members of the Parallel CFD & Optimization Unit (PCOpt/NTUA) for their willingness to answer my questions and support me through various stages of my research. I am especially grateful to Dr. Nikolaos Galanos for generously dedicating his time and sharing his valuable expertise. My sincere thanks also extend to Dr. Evangelos Papoutsis–Kiachagias and Dr. Varvara Asouti for their helpful advice and continuous support.

I would also like to acknowledge the support I received from my closed personal circles. My unwavering gratitude and love go to my parents and sister, who have always been my greatest supporters at every step of my life. Finally, heartfelt thanks to my dear friends Dimitris, Maria, and Alexis; their kindness and motivation were a steady source of energy and inspiration every step of the way.

National Technical University of Athens
School of Mechanical Engineering
Fluids Section
Parallel CFD & Optimization Unit

Curvature Computation on Surface Meshes for Adjoint-Based Constrained Optimization

Diploma Thesis

Anna Chotzalli

Advisor: Kyriakos C. Giannakoglou, Professor NTUA

Athens, 2025

Abstract

This diploma thesis focuses on mathematical formulating, programming and implementing high-accuracy methods to compute curvature over unstructured surface meshes, with the goal of incorporating curvature as a constraint in shape or topology adjoint-based optimization. Curvature-constrained optimization has attracted significant attention in recent years due to its critical role in ensuring durability, manufacturability, and other curvature-related properties of optimized shapes across various engineering applications.

According to the literature, efforts to incorporate curvature revealed significant accuracy issues in existing computational methods. Many state-of-the-art approaches fail to capture local point-wise curvature accurately, leading to misleading estimates of total surface curvature. The performed research revealed a critical sensitivity of curvature estimates to the geometry of mesh elements. This sensitivity is particularly pronounced in unstructured meshes, where local irregularities are often unavoidable. The problem intensifies during an optimization loop, since the mesh might become distorted. This can lead to divergence, as constraints based on inaccurate curvature values may no longer reflect the true surface geometry.

To address these challenges, this diploma thesis presents an in-depth analysis of the fundamental shortcomings present in conventional curvature estimation techniques. The findings identify ill-posed vertex-centered finite volume formulations as a primary source of inaccuracy in the widely used methods. In response, a novel and improved approach is introduced here: the Smoothed Geometry-Adaptive Corrected

(SGAC) Voronoi method, applicable to both mean and Gauss curvature estimation.

The SGAC method builds upon the widely adopted Voronoi-based framework, addressing two key limitations observed in its application to nodal finite volume construction: (i) the assumption that obtuse triangles can be treated as right-angled, and (ii) the uniform area distribution to the vertices in each obtuse triangle, irrespective of its specific geometric configuration.

The SGAC method overcomes these limitations by incorporating geometry-aware corrections for obtuse triangles and ensuring a smooth transition from the original method (used for acute and right triangles) to the corrected one for obtuse triangles. As a result, the proposed method offers enhanced precision in curvature computation, making it well-suited for applications in shape and topology optimization.

Custom algorithms were developed in C++ to compute two fundamental curvature metrics, mean and Gauss curvature, which were subsequently used to define the total surface curvature. These algorithms were designed to handle meshes with triangular and quadrilateral elements, ensuring compatibility with various computational frameworks. To evaluate the accuracy of the proposed method, this was compared with analytical results on surfaces described by closed form expressions and benchmarked against state-of-the-art software for CFD simulation or visualization and post-processing of computed flow fields. In all examined cases, the method demonstrated superior accuracy. Finally, the resulting total curvature function was formulated for use as constraint function in optimization problems and validated for its effectiveness in an example used as a surface roughness metric.

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Μηχανολόγων Μηχανικών
Τομέας Ρευστών
Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής & Βελτιστοποίησης

Υπολογισμός Καμπυλότητας σε Επιφανειακά Πλέγματα για τη Βελτιστοποίηση μέσω Συζυγούς Μεθόδου με Περιορισμούς

Διπλωματική Εργασία

Άννα Χοτζάλλη

Επιβλέπων: Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

Αθήνα, 2025

Περίληψη

Αντικείμενο της παρούσας διπλωματικής εργασίας αποτελεί η μαθηματική διατύπωση, ο προγραμματισμός και η υλοποίηση μεθόδων υψηλής ακρίβειας για τον υπολογισμό της καμπυλότητας σε μη-δομημένα πλέγματα επιφανειών, με τελικό στόχο την ενσωμάτωση της καμπυλότητας ως περιορισμό σε προβλήματα βελτιστοποίησης μορφής ή τοπολογίας με χρήση της συζυγούς μεθόδου. Η βελτιστοποίηση με περιορισμό στην καμπυλότητα έχει προσελκύσει σημαντικό ερευνητικό ενδιαφέρον τα τελευταία χρόνια, καθώς διαδραματίζει κρίσιμο ρόλο στη διασφάλιση της αντοχής, της κατασκευασιμότητας και άλλων ιδιοτήτων που σχετίζονται με την καμπυλότητα των βελτιωμένων σχημάτων σε πληθώρα εφαρμογών μηχανικής.

Σύμφωνα με τη σχετική βιβλιογραφία, οι προσπάθειες ενσωμάτωσης της καμπυλότητας έχουν αποκαλύψει σημαντικά προβλήματα ακρίβειας στις υπάρχουσες αριθμητικές μεθόδους. Πολλές από τις πλέον διαδεδομένες προσεγγίσεις αποτυγχάνουν να αποδώσουν με ακρίβεια την τοπική σημειακή καμπυλότητα, οδηγώντας έτσι σε παραπλανητικές εκτιμήσεις της ολικής καμπυλότητας επιφάνειας. Η έρευνα που πραγματοποιήθηκε ανέδειξε μία κρίσιμη ευαισθησία των εκτιμήσεων καμπυλότητας στη γεωμετρία των στοιχείων του πλέγματος. Το φαινόμενο αυτό είναι ιδιαίτερα έντονο σε μη-δομημένα πλέγματα, όπου οι τοπικές ανωμαλίες είναι συνήθως αναπόφευκτες. Το πρόβλημα εντείνεται κατά τη διάρκεια της βελτιστοποίησης, όπου το πλέγμα ενδέχεται να παραμορφωθεί, οδηγώντας εν δυνάμει σε αποκλίσεις λόγω εσφαλμένης εκτίμησης της καμπυλότητας, που δεν αντιπροσωπεύει πλέον την πραγματική γεωμετρία της επιφάνειας.

Για την αντιμετώπιση αυτών των προκλήσεων, η διπλωματική αυτή εργασία προβαίνει σε εις βάθος ανάλυση των βασικών ελλείψεων των συμβατικών τεχνικών προσδιορισμού καμπυλότητας. Η ανάλυση κατέδειξε ότι η κύρια πηγή σφαλμάτων στις ευρέως χρησιμοποιούμενες μεθόδους είναι η προβληματική διατύπωση των κεντροκομβικών όγκων ελέγχου. Για την αντιμετώπιση αυτών των προβλημάτων, προτείνεται μία νέα και βελτιωμένη προσέγγιση, η οποία ονομάζεται Εξομαλυμένη Γεωμετρικά-Προσαρμοσμένη (SGAC) Voronoi Μέθοδος, κατάλληλη για τον υπολογισμό τόσο της μέσης όσο και της Gauss καμπυλότητας.

Η μέθοδος SGAC βασίζεται στην ευρέως χρησιμοποιούμενη μέθοδο Voronoi, αλλά έρχεται να διορθώσει δύο σημαντικές αδυναμίες της που παρατηρούνται κατά τον ορισμό των κόμβο-κεντραρισμένων όγκων ελέγχου: (i) την υπόθεση ότι τα αμβλυγώνια τρίγωνα μπορούν να αντιμετωπιστούν ως ορθογώνια, και (ii) τον ομοιόμορφο τρόπο κατανομής του εμβαδού στις κορυφές των αμβλυγώνιων τριγώνων, ανεξάρτητα από την ακριβή γεωμετρία τους.

Η μέθοδος SGAC ξεπερνά αυτές τις αδυναμίες εισάγοντας διορθώσεις που λαμβάνουν υπόψη τη γεωμετρία για τα αμβλυγώνια τρίγωνα και εξασφαλίζοντας μία ομαλή μετάβαση από τη συμβατική μέθοδο (που χρησιμοποιείται για οξυγώνια και ορθογώνια τρίγωνα) στη διορθωμένη μέθοδο για τα αμβλυγώνια. Ως εκ τούτου, η προτεινόμενη μέθοδος προσφέρει βελτιωμένη ακρίβεια στον υπολογισμό της καμπυλότητας, καθιστώντας την κατάλληλη για χρήση σε εφαρμογές βελτιστοποίησης μορφής ή τοπολογίας.

Αναπτύχθηκαν ειδικοί αλγόριθμοι σε C++ για τον υπολογισμό των δύο βασικών μετρικών καμπυλότητας, της μέσης και της Gauss καμπυλότητας, οι οποίες στη συνέχεια χρησιμοποιούνται για τον υπολογισμό της ολικής καμπυλότητας επιφανείας. Οι αλγόριθμοι αυτοί σχεδιάστηκαν ώστε να υποστηρίζουν πλέγματα με τριγωνικά και τετραγωνικά στοιχεία, εξασφαλίζοντας συμβατότητα με ένα ευρύ φάσμα υπολογιστικών πλαισίων. Για την αξιολόγηση της ακρίβειας της προτεινόμενης μεθόδου, αυτή επαληθεύτηκε έναντι αναλυτικών αποτελεσμάτων σε επιφάνειες που περιγράφονται από κλειστού τύπου εκφράσεις και συγκρίθηκε με ευρέως γνωστά λογισμικά προσομοίωσης CFD, καθώς και με λογισμικά για την απεικόνιση και τη μετεπεξεργασία των υπολογισθέντων πεδίων ροής. Σε όλες τις περιπτώσεις, η μέθοδος παρουσίασε σημαντικά βελτιωμένη ακρίβεια. Τέλος, διαμορφώθηκε η συνάρτηση ολικής καμπυλότητας ώστε να μπορεί να χρησιμοποιηθεί ως συνάρτηση περιορισμού σε προβλήματα βελτιστοποίησης, και επαληθεύτηκε ως προς την αποτελεσματικότητά της σε ένα παράδειγμα χρήσης της ως μέτρου της τραχύτητας επιφανειών.

Contents

Chapter 1

Introduction

Design optimization is widely used in mechanical engineering fields such as aeronautics, automotive, and turbomachinery, where it involves using advanced computational techniques to solve complex Computational Fluid Dynamics (CFD) equations, such as the Navier-Stokes equations. The primary goal is to create the best design while adhering to the governing field equations. This involves controlling the design variable space under specific constraints to improve the predefined characteristics like performance or weight. In recent years, structural optimization has evolved into shape and topology optimization, which reflects the growing recognition of the importance of modifying a structure's shape and topology to meet design requirements, capabilities that are inherently difficult to achieve merely through size optimization [6, 8]. Some characteristic examples could be the aerodynamic shape optimization of airfoils for drag minimization [11] or topology optimization of two fluid heat exchangers [10]. Producing the desired 3D geometries in shape optimization involves appropriately adjusting the points that define the initial shape, whereas topology optimization focuses on identifying which regions of the design domain should be filled with solid material.

Among various constraints used in design optimization, curvature constraints are of great importance across a wide range of applications. More specifically, they are essential when designing mechanical parts to ensure structural durability by avoiding complex shapes that cause stress concentrations [16]. At the same time, they have a crucial role in maintaining the manufacturability of structures, given that optimized shapes or topologies are of limited value if their production is not feasible. For instance, curvature limitations in milling and cutting processes help avoid sharp corners, which are difficult and consequently costly to process using milling machine cutters [25]. In additive manufacturing processes, curvature restrictions prevent geometric singularities that pose challenges related to resolution and support structures during fabrication [23]. In many applications, curvature constraints

can also help satisfy functional and aesthetic preferences of the improved geometry. Given the importance of applying curvature limitations in CFD-based shape or topology optimization frameworks, it is important to develop reliable methods for accurately determining curvature measures on the boundary surfaces of the structures under consideration and incorporating them as inequality constraints. This objective forms the central focus of the research presented in this diploma thesis.

1.1 Curvature - Mathematical Background

In this section, an introduction of fundamental concepts related to the measurement of curvature is provided. Curvature is one of the important geometric quantities and helps on determining how a surface bends or deviates from being flat in case of surfaces or straight in case of curves, providing a precise measure of its local geometric shape. For a plane curve, curvature quantifies the rate of change of the tangent vector's direction along the curve, whereas for a surface, curvature becomes more complex, indicating the surface's bending in various directions at each point.

A brief definition of the curvature of a plane curve and a surface is given as described in Chapters 4 and 9 of [13]. Firstly, a regular plane curve C represented using the arc - length parameter s by $\vec{x} = \vec{x}(s)$ is assumed. The tangent vector to C at the point $\vec{x}(s)$ is defined by the derivative $\dot{\vec{x}}(s) = \frac{d\vec{x}}{ds}$, which is given by:

$$\dot{\vec{x}}(s) = \lim_{\Delta s \to 0} \frac{\vec{x}(s + \Delta s) - \vec{x}(s)}{\Delta s} \tag{1.1}$$

where $\frac{\vec{x}(s+\Delta s)-\vec{x}(s)}{\Delta s}$ is a secant to C, Fig. 1.1. $\dot{\vec{x}}(s)$ has unit length due to arc-length parameterization and this is referred to as unit tangent vector to the curve C at $x(s)$ and is usually denoted as $\vec{t} = \vec{t}(s) = \dot{\vec{x}}(s) = \frac{d\vec{x}/dt}{ds/dt}$, where t is the parameterization of the curve.

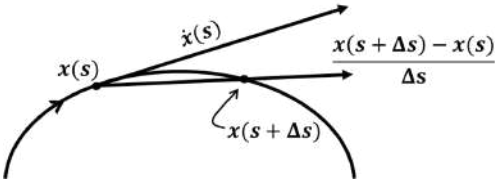


Figure 1.1: *Tangent direction of a plane curve.[13]*

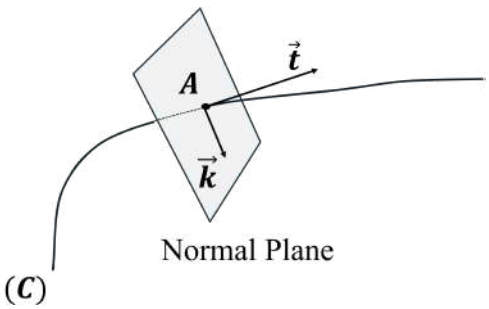


Figure 1.2: *Normal plane, tangent vector $\vec{t}$ and curvature vector $\vec{k}$ at point A of curve C [13].*

Secondly, the normal plane to curve C at A is the plane through A, which is orthogonal to the tangent vector $\vec{t}$ at point A, Fig. 1.2.

The curvature vector $\vec{k}$ of a regular C^2 curve at point A on $\vec{x}(s)$ is the derivative of the tangent vector: $\vec{k}(s) = \dot{\vec{t}}(s) = \frac{d^2\vec{x}}{ds^2}$. Unlike the unit tangent vector, $\vec{k}$ is generally not unit length. Its direction is always orthogonal to the tangent vector $\vec{t}$ and thus lies in the normal plane of the curve at point A. Its magnitude is called curvature of a curve at point A: $\kappa = \left|\vec{k}\right|$. This scalar quantity describes how sharply the curve bends at a given point by measuring the rate at which the tangent vector changes direction and is one of the two essential local invariants that define a plane curve, which means that it doesn't change when subjected to rigid transformations.

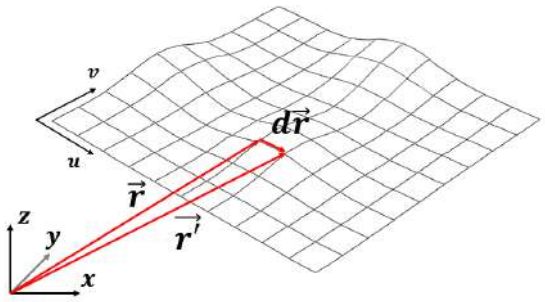


Figure 1.3: *Differential position vector $d\vec{r}$ between nodes of a surface mesh.*

Similarly to the plane curve, there are two local invariant quantities called first and second fundamental forms that characterize a surface mesh, which means that they remain unchanged under parameter transformation. Consider a surface which is parameterized by coordinates u and v, Fig. 1.3. The first fundamental form describes the squared length of any displacement vector on the surface and is given by:

$$I = d\vec{r} \cdot d\vec{r} = (\frac{\partial \vec{r}}{\partial u}du + \frac{\partial \vec{r}}{\partial v}dv) \cdot (\frac{\partial \vec{r}}{\partial u}du + \frac{\partial \vec{r}}{\partial v}dv) = Edu^2 + 2Fdudv + Gdv^2 \quad (1.2)$$

where $E = \vec{r}_u \cdot \vec{r}_u$, $F = \vec{r}_u \cdot \vec{r}_v$ and $G = \vec{r}_v \cdot \vec{r}_v$ are the first fundamental coefficients that characterize each point.

Each point of the surface mesh has a unit normal vector $\underline{\vec{N}}$, Fig. 1.4, which is given by the following expression:

$$\underline{\vec{N}} = \frac{\vec{r}_u \times \vec{r}_v}{|\vec{r}_u \times \vec{r}_v|} \quad (1.3)$$

and is dependent only on the surface variables u and v. The variation of this vector

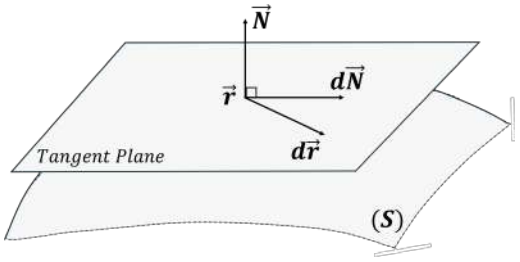


Figure 1.4: *Unit normal vector to the surface S at point $\vec{r}$ [13].*

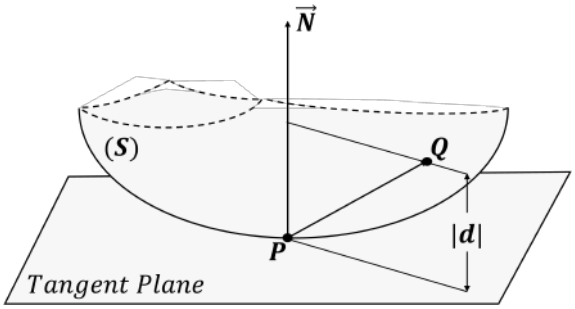


Figure 1.5: *Local shape of the surface S with respect to the distance |d| [13].*

can be expressed as $d\underline{\vec{N}} = \underline{\vec{N}}_u du + \underline{\vec{N}}_v dv$, which lies in the tangent plane at point A. Since $d(\underline{\vec{N}} \cdot \underline{\vec{N}}) = 2\underline{\vec{N}} \cdot d\underline{\vec{N}} = d(1) = 0$, $d\underline{\vec{N}}$ and $\underline{\vec{N}}$ are orthogonal vectors.

The second invariant is the second fundamental form, which describes how the normal vector changes as one moves along the surface. In other words, it characterizes how the surface bends within the $\mathbb{R}^3$ space and is given by:

$$II = -d\vec{r} \cdot d\underline{\vec{N}} = -(\vec{r}_u du + \vec{r}_v dv) \cdot (\underline{\vec{N}}_u du + \underline{\vec{N}}_v dv) = Ldu^2 + 2Mdudv + Ndv^2 \tag{1.4}$$

where $L = \Omega_{11} = -\vec{r}_u \cdot \underline{\vec{N}}_u$, $M = \Omega_{12} = \Omega_{21} = -\frac{1}{2}(\vec{r}_u \cdot \underline{\vec{N}}_v + \vec{r}_v \cdot \underline{\vec{N}}_u)$ and $N = \Omega_{22} = -\vec{r}_v \cdot \underline{\vec{N}}_v$ are the second fundamental coefficients. However, in order for its sign to remain unchanged, the orientation of $\underline{\vec{N}}$ must also be preserved.

The physical meaning of the second fundamental form is closely bonded with the nature of a surface around a node. More specifically, assuming $P(u, v)$ to be a point on the surface S and point $Q(u + du, v + dv)$ to be a near neighbor point on the surface then $|d| = \vec{PQ} \cdot \underline{\vec{N}}$ is the projection of $\vec{PQ}$ onto $\underline{\vec{N}}$ at point P, Fig. 1.5. The sign of $\vec{d} = |d|\underline{\vec{N}}$, Eq. (1.5), depends on which side of the tangent plane the point Q lies. It is considered positive when Q lies on the same side with normal vector $\underline{\vec{N}}$, and negative otherwise,

$$\vec{d} = \frac{1}{2} d^2\vec{r} \cdot \underline{\vec{N}} + O(du^2 + dv^2) = \frac{1}{2} II + O(du^2 + dv^2) \tag{1.5}$$

which means that II is approximately twice the projection of $\vec{PQ}$ to $\underline{\vec{N}}$.

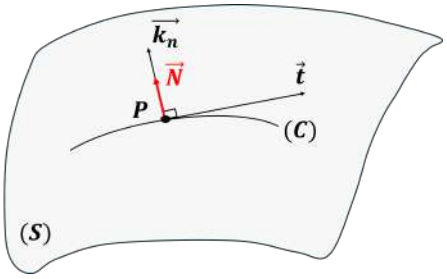


Figure 1.6: *Normal curvature vector $\vec{k}_n$ of curve C at point P, parallel to the surface normal vector $\underline{\vec{N}}$ [13].*

Now that the necessary background has been provided, the curvature of a surface can be defined. The normal curvature of a curve C lying on a surface S at a point P is defined as the projection of the curvature vector of C onto $\underline{\vec{N}}$ at P, Fig. 1.6, and is given by:

$$\vec{k}_n = (\vec{k} \cdot \underline{\vec{N}}) \cdot \underline{\vec{N}} \tag{1.6}$$

$\vec{k}_n$ is independent of the sense of curve C. Its magnitude is called normal curvature of C at P and is given as $\kappa_n = \vec{k} \cdot \underline{\vec{N}}$. The normal curvature depends on the direction of vector $\underline{\vec{N}}$ but not on the direction of C and can be expressed as:

$$\kappa_n = \frac{L du^2 + 2M dudv + N dv^2}{E du^2 + 2F dudv + G dv^2} = \frac{II}{I} \tag{1.7}$$

where du refers to du/dt and dv refers to dv/dt. It is important to note that κ_n, as a function of du/dt and dv/dt, depends only on the ratio of $(du/dt)/(dv/dt)$, in other words on the direction of the tangent line to the curve C at point P. This implies that all curves lying on the surface S and passing through P which are tangent to the same line through P, have the same normal curvature κ_n. Furthermore, given that fundamental forms I and II are invariant under parameter transformation (i.e. by changing the surface mesh), the normal curvature at point P on curve C is also invariant.

There is a convenient local coordinate system in which the normal curvature at P, Eq. (1.7), can be expressed as:

$$\kappa_n = \frac{Ldu^2 + 2Mdudv + Ndv^2}{du^2 + dv^2} \tag{1.8}$$

Assuming that $du^2 + dv^2 = 1$, the parameter derivatives can be set as $du = \cos\theta$ and $dv = \sin\theta$, where θ is the angle representing the direction of motion in the local coordinate system. By changing the angle, different tangent lines corresponding to normal sections are obtained, producing various curves C passing through P on surface S. It can be shown that two specific angles, and thus two curves, exist where the normal curvature at P reaches its maximum and minimum values, Fig. 1.7). These mutually perpendicular directions are called principal directions, and their normal curvatures are the principal curvatures, denoted κ_1 and κ_2, respectively. Given the invariance of κ_n at point P on curve C, κ_1 and κ_2, are invariant too. Furthermore, any change in the orientation of the surface will reverse the sign of κ_n, meaning that the magnitudes of the principal curvatures remain the same, but their signs are reversed.

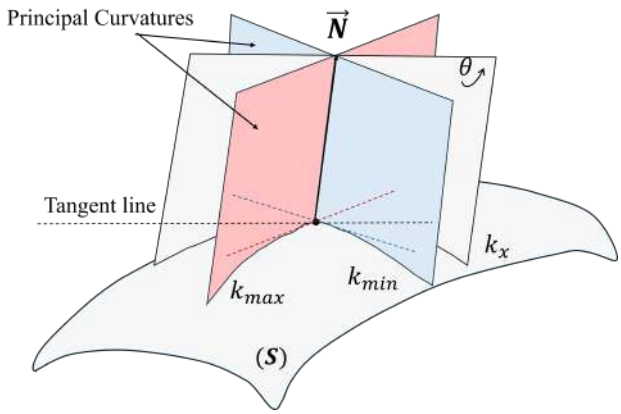


Figure 1.7: *Principal curvature normal sections among various normal sections with varying angle θ at point P on surface S.*

A number κ is principal curvature if and only if κ is a solution of the following equation:

$$(EG-F^2)\kappa^2-(EN+GL-2FM)\kappa+(LN-M^2)=0 \quad \Rightarrow \quad \kappa^2-2H\kappa+K=0, \tag{1.9}$$

where

$$H = \frac{EN + GL - 2FM}{2(EG - F^2)} \quad \text{and} \quad K = \frac{LN - M^2}{EG - F^2}.$$

In Eq. (1.9), $2H$ equals the sum of the two roots, and K equals their product. That being said, the average of the two roots κ_1 and κ_2 is defined as the mean curvature H at point P, and is given by:

$$H = \frac{1}{2}(\kappa_1 + \kappa_2) = \frac{EN + GL - 2FM}{2(EG - F^2)} \tag{1.10}$$

In addition, the product of the two roots κ_1 and κ_2 is defined as the Gauss curvature at P, and is given by:

$$K = \kappa_1 \kappa_2 = \frac{LN - M^2}{EG - F^2} \tag{1.11}$$

The physical meaning and methods for computing the mean and the Gauss curvatures of surfaces will be examined in more detail in the following chapters of this work.

1.2 The Curvature as a CFD-based Optimization Constraint

Curvature constraints have previously been incorporated into shape and topology optimization in structural mechanics across various applications, such as composite laminates [9], infinite plates with circular holes [21], and compliant mechanisms [22]. In these contexts, Finite Element Analysis (FEA) is typically used, where different methods are employed for computing curvature measures. Among these, the most widely used approaches are the Smooth Surface Fit (SSF) using a triple-node knot [21], and curvature based on the Discrete Local Laplace–Beltrami (DLLB) operator [3]. As analyzed in [3], the most accurate method for computing the mean curvature is the DLLB approximation.

In contrast, CFD problems, such as those examined in this work, commonly use the Finite Volume Method (FVM) for discretization. FVM is widely adopted due to its strictly conservative nature, which is important for satisfying the conservation equations that govern fluid fields, and its ease in handling boundary conditions for primary variables. However, incorporating curvature constraints within the FVM framework presents three major challenges: (i) the need to demonstrate that the DLLB method for mean curvature is applicable and equally effective for FVM meshes, (ii) the requirement to transform the widely used cell-centered FVM structure to a vertex-centered representation for curvature computation, as curvature measures are originally defined at surface mesh nodes, and (iii) the need to define, within the FVM, a single metric that effectively characterizes the total curvature of the structure's surface geometry.

All these challenges have been addressed in this diploma thesis.

1.3 Thesis Outline

This diploma thesis focuses on the mathematical formulation, programming, and implementation of high-accuracy methods for curvature approximation over unstructured surface meshes, aiming to incorporate curvature as a constraint in shape or topology optimization. The mathematical framework improves upon existing literature methods and introduces a novel approach for defining vertex-centered finite volumes, significantly enhancing curvature estimation accuracy.

The existing and newly proposed methods were implemented in a C++ framework, supporting structured and unstructured meshes with triangular and quadrilateral elements. Their accuracy was validated and compared with state-of-the-art software for CFD simulation or visualization and post-processing of computed flow fields, using test cases with known analytical curvature values.

The structure of the diploma thesis is as follows:

Chapter 2: Presents a detailed analysis of mean curvature estimation over triangulated surfaces. Existing methods are reviewed and their limitations identified. A novel method for vertex-centered finite volume definition, named the Smoothed Geometry-Adaptive Corrected (SGAC) Voronoi method, is introduced. Its accuracy is evaluated in test cases with known analytical mean curvature, such as the sphere.

Chapter 3: Focuses on Gaussian curvature estimation over triangulated surfaces. The existing methods are reviewed. The SGAC method is extended to this metric and validated in benchmark cases with analytical Gauss curvature.

Chapter 4: Extends the proposed curvature estimation framework to quadrilateral meshes. Different triangulation techniques are examined, and a dual-triangulation averaging method is introduced. The approach is tested in additional cases, such as a surface adjacent to a sphere.

Chapter 5: Evaluates the accuracy of the implemented methods on structured and unstructured meshes over geometries like the torus and saddle surface. The proposed method is benchmarked against literature methods and state-of-art software for CFD simulation or visualization and post-processing of computed flow fields. The final expression of the total surface curvature function is formulated for use as a constraint in optimization and tested for its effectiveness in an arbitrary surface roughness example.

Chapter 6: Provides a summary of the diploma thesis, along with conclusions drawn from this research and recommendations for future work.

Chapter 2

Computation of Mean Curvature on Surface Meshes

2.1 Introduction

In this section, a quick introduction to the meaning of the mean curvature measure at points over a surface is provided.

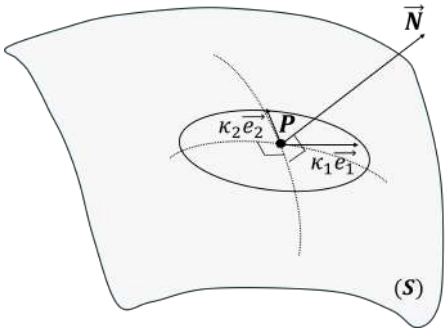
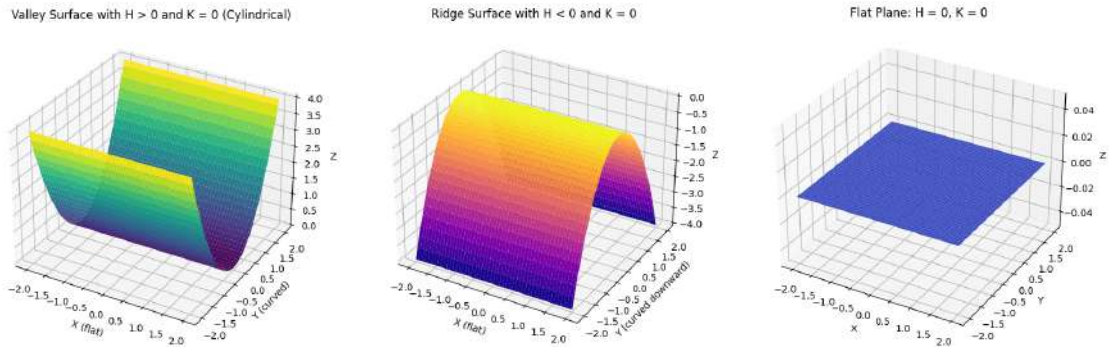


Figure 2.1: *Principal directions and curvatures at point P on surface S.*

As described in Chapter 1, mean curvature H at point P, Fig. 2.1, is the average value of the maximum and minimum normal curvature of a surface point. Its sign depends on the orientation of $\underline{\vec{N}}$. In this work, the sign of H, which indicates the direction of $H\underline{\vec{N}}$, will be taken opposite to the one of normal vector $\underline{\vec{N}}$, such that for convex surfaces (e.g., a sphere), the mean curvature vector at each point points inward. This convention allows the sign of H to convey information about the local shape of the surface: positive H indicates a locally convex region, negative

H indicates a locally concave region, and $H = 0$ corresponds to a locally minimal surface, where the principal curvatures are equal in magnitude but opposite in sign.

As regards its physical meaning, the mean curvature of a smooth surface measures how much the surface area changes compared to neighbor surfaces. In simpler terms, it measures how the surface is evolving when it moves in the direction of its normal. More specifically, the surface is minimal if $H = 0$, which indicates that it does not expand or contract during this motion. The surface tends to expand or contract in the case of negative and positive curvature, respectively. This can be illustrated by observing that the same local surface geometry can appear as a peak or pit depending on the sign of H. This is clearly demonstrated in the simple example of surfaces with Gauss curvature $K = 0$, but differing signs of H, Fig. 2.2.



(a) $H > 0$, $K = 0$ *(Valley)* **(b)** $H < 0$, $K = 0$ *(Ridge)* **(c)** $H = 0$, $K = 0$ *(Flat)*

Figure 2.2: *Surface shapes based on the sign of H.*

2.2 Mean Curvature Approximation on a Surface

This section explores methods for computing H at the nodes of an unstructured surface mesh. In structured meshes, the parameterization is correlated with the mesh itself. H is a local invariant quantity on any surface mesh, derived from the two principal curvatures κ_1 and κ_2 at a point on the surface S. This implies that the local curvature properties belong to the surface and are not affected by any change in parameterization. In contrast, for unstructured meshes, where global parameterization of the entire mesh is not feasible, alternative methods must be employed. The Laplace - Beltrami surface operator applied to the position vector $\vec{r}$ is equal to (Appendix A):

$$\Delta_s \vec{r} = \nabla_s^2 \vec{r} = 2H\underline{\vec{N}} \tag{2.1}$$

The *Laplace - Beltrami* operator Δ_s is the generalization of the *Laplacian* operator

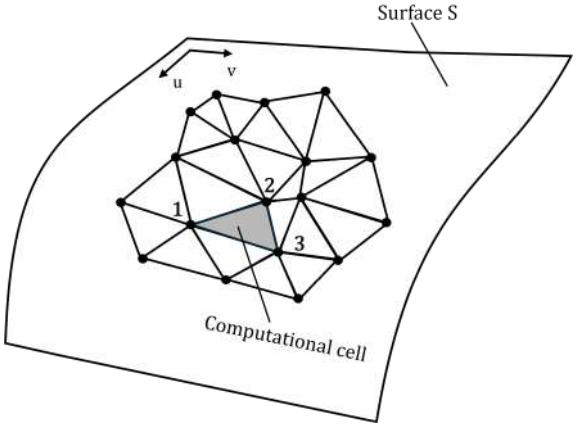


Figure 2.3: *Discretized surface with triangular elements.*

to functions defined on surfaces.

The task of determining H is now framed as computing the *Laplace - Beltrami* operator of the position vector and $\vec{N}$ on a smooth surface.

Instead of computing the continuous Laplacian operator in the global coordinates of $\mathbb{R}^3$, an approach is employed that projects it into a finite-dimensional space. The surface, parameterized by u and v, is discretized using triangular finite volume elements, Fig. 2.3. Consequently, the continuous differential operator is transformed into a discrete form by expressing Eq. (2.1) in its weak formulation, utilizing test basis functions. This transformation is derived from the Galerkin method, commonly used in the FEM.

Eq. 2.1 is written as:

$$\nabla_S^2 \vec{r} = \vec{f} \quad \text{in } S \tag{2.2}$$

where $S \subset \mathbb{R}^3$ is a bounded, simply connected domain and $\vec{f} = 2H\vec{N}$. Given that $\vec{r}$ is known at the nodes of the elements, Dirichlet boundary conditions can be imposed:

$$\vec{r} = \vec{g}^{(D)} \quad \text{in } \vartheta S \tag{2.3}$$

So, since the surface S is approximated by a collection of triangles S_h, Eq. (2.2) can be discretized by integrating over the triangulated surface and, thus, be written as:

$$\iint_{S_h} \nabla_{S_h}^2 \vec{r}_{S_h} \, dS_h = \iint_{S_h} \vec{f} \, dS_h \tag{2.4}$$

where $\vec{r}_{S_h}$ is linear for every element. If X_h denotes the finite-dimensional approxi-

mation space over S_h, and if $\vec{r}_{S_h}$ interpolates the Dirichlet boundary condition $\vec{g}^{(D)}$ along the boundary ϑS_h, then the solution $\vec{r}$ can be approximated by $\vec{r}_{S_h} \in X_h$.

The next stage in the Galerkin method, [12], is to multiply each term by an appropriate test shape function ϕ so that $\phi \subset X_h$ and ϕ satisfies the Dirichlet condition of Eq. (2.3). Since $\vec{r}_{S_h}$ is prescribed on the boundary by the given condition, the trial shape function ϕ is zero on the boundary ϑS ($\phi = 0$ on ϑS), ensuring that the boundary value of $\vec{r}_{S_h}$ remains unchanged,

$$\iint_{S_h} \phi \nabla_{S_h} \cdot \nabla_{S_h} \vec{r}_{S_h} \, dS_h = \iint_{S_h} \phi \vec{f} \, dS_h$$
$$\iint_{S_h} \nabla_{S_h} \cdot (\phi \nabla_{S_h} \vec{r}_{S_h}) \, dS_h - \iint_{S_h} \nabla_{S_h} \vec{r}_{S_h} \cdot \nabla_{S_h} \phi \, dS_h = \iint_{S_h} \phi \vec{f} \, dS_h \tag{2.5}$$

The Green - Gauss theorem is then applied in Eq. (2.5) to express the first term as an integral on the boundary of the region:

$$\oint_{\vartheta S_h} \phi \nabla_{S_h} \cdot \vec{n} \, dS_h - \iint_{S_h} \nabla_{S_h} \vec{r}_{S_h} \cdot \nabla_{S_h} \phi \, dS_h = \iint_{S_h} \phi \vec{f} \, dS_h \tag{2.6}$$

Given that $\phi = 0$ on ϑS, the boundary integral term vanishes and the final Galerkin formulation gives:

$$\iint_{S_h} \nabla_{S_h} \vec{r}_{S_h} \cdot \nabla_{S_h} \phi \, dS_h = \iint_{S_h} \phi \vec{f} \, dS_h \tag{2.7}$$

where the surface gradient $\nabla_{S_h} \vec{r}_{S_h} = \nabla \vec{r}_{S_h} - (\nabla \vec{r}_{S_h} \cdot n_h) n_h$ is the tangential gradient on S_h, ∇ is the 3D gradient and n_h is the normal vector to S_h.

The surface gradient $\nabla_{S_h} \vec{r}_{S_h}$ is also constant since the operation is done in a linear triangular finite element where n_h and $\nabla \vec{r}_{S_h}$ are constant. Therefore, ϕ_{h_1},...,ϕ_{h_N}, where $N = 3$, are considered to be piecewise linear functions on S_h that are globally continuous and $\phi_{hj}(x_k) = \delta_{jk}$, so that $\vec{r}_{S_h} = \sum_{j=1}^{N} \vec{r_j} \phi_{h_j}$. Eq. (2.7) is now rewritten as a linear system of N equations:

$$\sum_{j=1}^{N} \vec{r_j} \iint_{S_h} \nabla_{S_h} \phi_{h_j} \cdot \nabla_{S_h} \phi_{h_k} \, dS_h = \iint_{S_h} \phi_{h_k} \vec{f} \, dS_h \tag{2.8}$$

where k = 1,...,N.

The formulation of Eq. (2.8) is comparable to the analogous one when using the

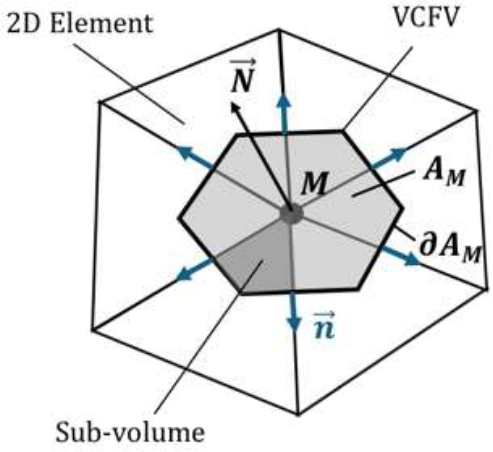


Figure 2.4: *Definition of Node-Centered Finite Volume.*

Galerkin method to a pure 2D (planar) problem, as stated in [5]. The sole difference when the surface is embedded in $\mathbb{R}^3$ is that each vertex has three coordinates rather than two.

Thus, it has been shown that the *Laplace–Beltrami* operator on a surface can be effectively approximated using the *Laplacian operator* within triangular elements, which are by nature 2D. In the proof, certain techniques from FEM were employed. However, this approximation of the *Laplace–Beltrami* operator can also be applied in the context of finite triangular volumes. As a result, this method can be used to compute the gradient of the position vector at each vertex, with the modification that the finite volumes must be transformed to a vertex-centered formulation. The details of these elements will be defined in the following section.

2.3 Definition of the Finite Volume in Triangular Elements

Given that the mean curvature is a node-based quantity, the finite volumes of the unstructured mesh, need to be formed into a vertex-centered representation. To achieve this, various methods for defining the Vertex-Centered Finite Volume (VCFV), Fig. 2.4, are presented in the following subsections. Three different methods for defining the sub-volume within each triangular element are presented. These approaches provide alternative ways to construct vertex-centered finite volumes and determine their area A_M and normal vector $\vec{N}_M$, which are essential for computing H.

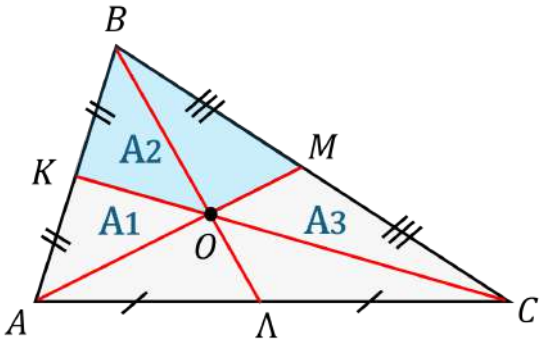


Figure 2.5: *Definition of sub-volume based on the barycenter of* $\triangle ABC$*, where* $A = 1$*,* $B = 2$ *and* $C = 3$*.*

2.3.1 Barycentric Definition of the VCFV

In this method, the sub-volume associated with each node is determined using the barycenter of the triangle, which is the intersection of its medians and the mid-points of all triangle edges. The barycentric approach ensures that the triangle's area is equally divided, assigning a fair share to each node. In Fig. 2.5, the blue area A_2 represents the sub-volume assigned to node B, with corresponding sub-volumes allocated to the other nodes in a similar manner.

It is known that the barycenter of a planar triangle with vertices at coordinates $A(u_1, v_1)$, $B(u_2, v_2)$ and $C(u_3, v_3)$, is given by:

$$(u_c, v_c) = \left(\frac{u_1 + u_2 + u_3}{3}, \frac{v_1 + v_2 + v_3}{3} \right) \tag{2.9}$$

where u_i and v_i are the local parametric coordinates of the surface.

The areas associated with each vertex, written as A_1, A_2 and A_3 in Fig. 2.5 are equal:

$$A_1 = A_2 = A_3 = \frac{A}{3} \tag{2.10}$$

where A is the area of $\triangle ABC$.

Consequently, the corresponding A_M area of each node is determined by:

$$A_M = \sum_{m=1}^{C_{m_i}} A_{m_i} \tag{2.11}$$

where $m = 1, ...C_{m_i}$ are the neighbor triangles of node i and A_{m_i} is the portion of the area of the neighbor triangle m that corresponds to node i of the triangle. The contribution of the triangle dimensional normal vector $\vec{N}$ to each vertex can be easily deduced as follows:

$$\vec{N_1} = \vec{N_2} = \vec{N_3} = \frac{\vec{N}}{3} \tag{2.12}$$

$\vec{N}$ is dimensional, with its magnitude equal to the surface area, thus possessing the same units as an area

Similarly with the area, the normal vector $\vec{N}_M$ that corresponds to each node is given by:

$$\vec{N}_M = \sum_{m=1}^{C_m} \vec{N}_{m_i} \tag{2.13}$$

where $m = 1, ...N_{m_i}$ and $\vec{N}_{m_i}$ is the portion of the normal vector of the neighbor triangle m that corresponds to node i of the triangle.

2.3.2 Voronoi Definition of the VCFV

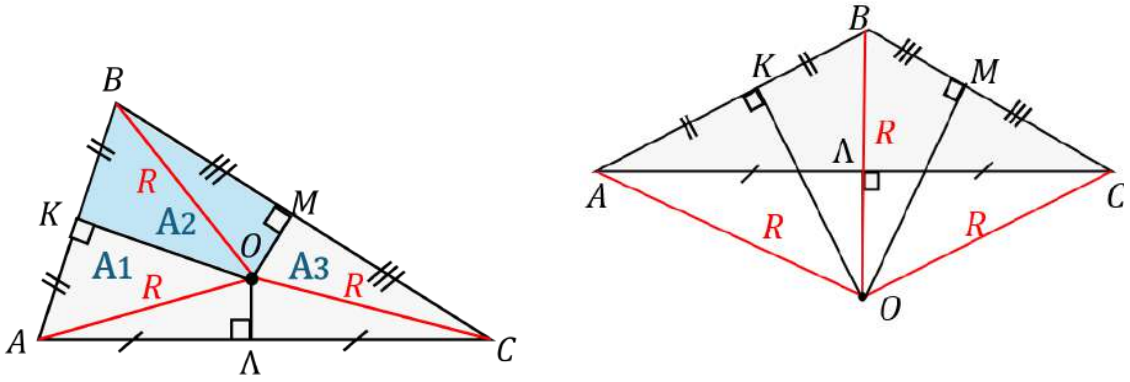
In this method, the portion of the triangle's area attributed to each node is determined using the circumcenter, which is the point of intersection of the three perpendicular bisectors of the triangle's sides. The circumcenter approach ensures that the division is made in such a way that the center (denoted as O) maintains the minimum distance from the triangle's vertices.

The circumradius R of the triangle, Fig. 2.6, is given by (its proof is provided in Appendix B):

$$R = \frac{abc}{\sqrt{(a+b+c)(b+c-a)(c+a-b)(a+b-c)}} \tag{2.14}$$

where $a = (BC)$, $b = (AC)$ and $c = (AB)$ the lengths of the edges of the triangle.

The area A_i sections of the triangle, where i=1,2,3, are given by:



(a) **(b)**

Figure 2.6: *Definition of sub-volume based on the circumcenter of (a) an acute and (b) an obtuse triangle $\triangle ABC$, where $A = 1$, $B = 2$ and $C = 3$.*

$$
\begin{aligned}
A_i &= A_{\triangle BKO} + A_{\triangle BOM} \\
A_i &= \frac{1}{2}\left|x_B(y_K - y_O) + x_K(y_O - y_B) + x_O(y_B - y_K)\right| \\
&+ \frac{1}{2}\left|x_B(y_O - y_M) + x_O(y_M - y_B) + x_M(y_B - y_O)\right|
\end{aligned} \tag{2.15}
$$

The dimensioned normal vector $\vec{N}_i$ of the sections, where i=1,2,3, are given by the formula:

$$
\vec{N}_i = \frac{A_i}{A}\vec{N} \tag{2.16}
$$

where A and $\vec{N}$ represent the area and the dimensional normal of the triangle $\triangle ABC$, respectively. The node's A_M and $\vec{N}_M$ are similar to those before given by Eq. (2.11) and Eq. (2.13).

It is important to note that this method is applicable only to acute triangles, as the circumcenter lies outside the triangle for obtuse triangles, Fig. 2.6, making it questionable to assign a valid area section to each vertex.

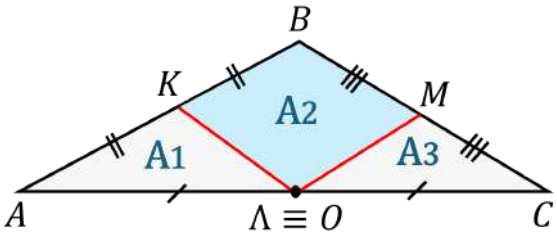


Figure 2.7: *Definition of sub-volume based on the corrected circumcenter of an obtuse triangle $\triangle ABC$, where $A = 1$, $B = 2$ and $C = 3$.*

2.3.3 Corrected Voronoi Definition of VCFV

In this section, the portion of the triangle's area assigned to each node is determined using the circumcenter, following the same approach as in the previous method. The key distinction lies in how Eq. (2.14) is modified for obtuse triangles, where the circumcenter lies outside the triangle. To ensure that Eqs. 2.11 and 2.13 yield valid area portions in such cases, a correction to the circumcenter placement is required. A practical solution [15] is to treat the obtuse triangle as a right triangle by positioning the circumcenter at the midpoint of the edge opposite the obtuse angle, as illustrated in Fig. 2.7. Based on this correction, the sub-area distribution within triangle $\triangle ABC$, with $\angle B$ to be obtuse, is given as follows:

$$A_A = \frac{1}{4}A_{\triangle ABC}, \quad A_B = \frac{1}{2}A_{\triangle ABC}, \quad A_C = \frac{1}{4}A_{\triangle ABC} \tag{2.17}$$

The comparison of H results obtained using different definitions of VCFV is presented in a subsequent section.

2.4 Computation of the Position Vector Gradient at Each Node

Now that the three VCFV definition methods have been prescribed in the previous sections, the mean curvature normal vector $H\vec{N}$ at a point M on the surface is derived through Eq. (2.1) by integrating it in the VCFV area A_M, Fig. 2.4, as follows:

$$\iint_{A_M} \nabla_s^2 \vec{r}\, dS = 2H\underline{\vec{N}} \iint_{A_M} dS \tag{2.18}$$

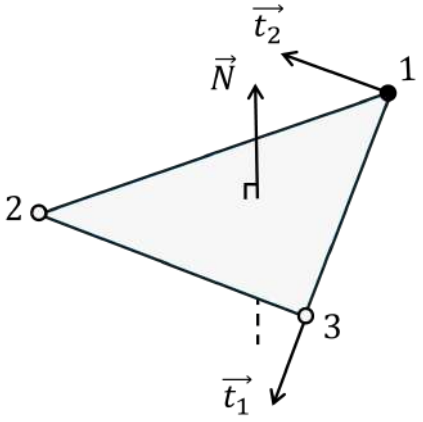


Figure 2.8: *Triangular element of the surface mesh.*

Using the established approximation of the *Laplace–Beltrami* operator on a triangulated surface, $\nabla_s^2 \approx \nabla_{u,v}^2$, where u and v denote the parameters of triangular elements, Eq. (2.19) can be expressed as follows:

$$\iint_{A_M} \nabla_{u,v}^2 \vec{r}\, dS = 2H\underline{\vec{N}} \iint_{A_M} dS \tag{2.19}$$

By applying the Green-Gauss theorem in Eq. (2.19), the final expression for the $H\underline{\vec{N}}$ of each node is obtained:

$$\oint_{\vartheta A_M} \nabla_{u,v}\vec{r} \cdot \vec{n}\, dS = 2H\underline{\vec{N}} \iint_{A_M} dS \tag{2.20}$$

where A_M the surface area corresponding to node M, ϑA_M the boundary of the area A_M and $\vec{n}$ the dimensionless normal vector of the boundary ϑA_M, Fig. 2.4.

In the subsequent sections, two approaches for computing H based on Eq. (2.1) are presented.

2.4.1 1st Approach: Computation of the Position Vector Gradient Using Edge Normals

Computing H at each node of an unstructured surface mesh involves the following steps:

1. *Compute the Local Reference Parameters u and v of the Triangular Element*

   In order to compute H at node 1, the triangulated surface mesh is considered, Fig. 2.8. The dimensional normal vector $\vec{N}$ of the element is calculated as:

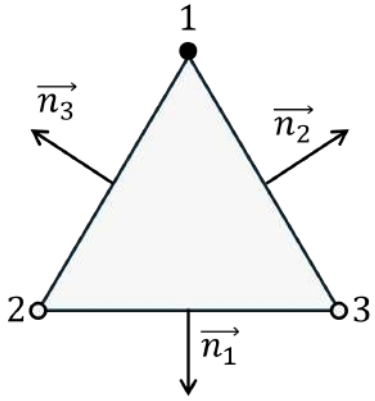


Figure 2.9: *Normal vectors to the edges of the triangular element.*

$$\vec{N} = \frac{1}{2}(\vec{r}_{12} \times \vec{r}_{13}) \tag{2.21}$$

where $\vec{r}_{12} = \vec{r}_2 - \vec{r}_1$ and $\vec{r}_{13} = \vec{r}_3 - \vec{r}_1$, with $\vec{r}_i$ representing the position vector of node i.

The magnitude of the normal vector, is equal to twice the area of the triangle. The dimensionless normal vector of the element is defined as:

$$\underline{\vec{N}} = \frac{\vec{N}}{\|\vec{N}\|} \tag{2.22}$$

The two dimensionless tangent vectors along the edges that intersect at node 1 of the triangle are then computed as follows:

$$\underline{\vec{t_1}} = \frac{\vec{r}_{13}}{\|\vec{r}_{13}\|}, \quad \underline{\vec{t_2}} = \frac{\vec{r}_{12}}{\|\vec{r}_{12}\|} = \frac{\vec{t_1} \times \vec{N}}{\|\vec{t_1} \times \vec{N}\|} \tag{2.23}$$

Thus, the local reference parameters for each node of the triangle are:

$$\begin{aligned} (u_1, v_1) &= (0, 0) \\ (u_2, v_2) &= (\vec{r}_{12} \cdot \underline{\vec{t_1}}, \vec{r}_{12} \cdot \underline{\vec{t_2}}) \\ (u_3, v_3) &= (\vec{r}_{13} \cdot \underline{\vec{t_1}}, \vec{r}_{13} \cdot \underline{\vec{t_2}}) \end{aligned} \tag{2.24}$$

2. *Compute the Normal Vectors to the Edges of the Triangular Element*

The normal vectors to the edges of the triangular element, Fig. 2.9, are computed with respect to the local reference coordinates as follows:

$$\begin{aligned}\vec{n}_1 &= (v_3 - v_2, u_2 - u_3)\\ \vec{n}_2 &= (v_1 - v_3, u_3 - u_1)\\ \vec{n}_3 &= (v_2 - v_1, u_1 - u_2)\end{aligned} \tag{2.25}$$

The corresponding dimensionless normal vectors to the edges are given by $\underline{\vec{n_j}} = \frac{\vec{n_j}}{\|\vec{n_j}\|}$, where $j = 1, \ldots, 3$.

3. *Compute the Position Vector Gradient of the Triangular Element*

   The gradient of a vector field $\vec{\Phi}$ is constant within a triangular element and is represented by a gradient tensor. If $\vec{\Phi} = \vec{r}$, this tensor is expressed as:

$$\nabla_{u,v}\vec{r} = \begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \\ \frac{\partial z}{\partial u} & \frac{\partial z}{\partial v} \end{bmatrix} \tag{2.26}$$

   If r_{i_k} denotes the coordinates (k = 1,2,3) of each triangular vertex (i = 1,2,3) of the position vector and (n_{i_u}, n_{i_v}) represents the coordinates of the normal vector along the edges, then the components of the gradient tensor are given by:

$$\begin{aligned}\frac{\partial r_k}{\partial u} &= -\frac{1}{2A_T}(r_{1_k}n_{1_u} + r_{2_k}n_{2_u} + r_{3_k}n_{3_u})\\ \frac{\partial r_k}{\partial v} &= -\frac{1}{2A_T}(r_{1_k}n_{1_v} + r_{2_k}n_{2_v} + r_{3_k}n_{3_v})\end{aligned} \tag{2.27}$$

   for each k coordinate of $\vec{r} = (x, y, z)$. A_T is the area of the triangular element and is given by $A_T = \frac{1}{2}(u_1v_2 + u_2v_3 + u_3v_1 - u_1v_3 - u_2v_1 - u_3v_2)$.

4. *Compute the Dimensioned Normal Vector $\vec{n}$ of ϑA_M*

   The normal vector of each VCFV at node M is determined by contributions from the neighboring triangles, Fig. 2.4. Within each triangle, the normal vector $\vec{n}$ at the corresponding edges of border ϑA_M is equal to the normal vector of edge 45, which connects the midpoints of edges 12 and 13 of triangle $\triangle 123$, Fig. 2.10.

   Given that $\triangle 123$ and $\triangle 145$ are similar triangles, it follows that $Area[\triangle 145] =$

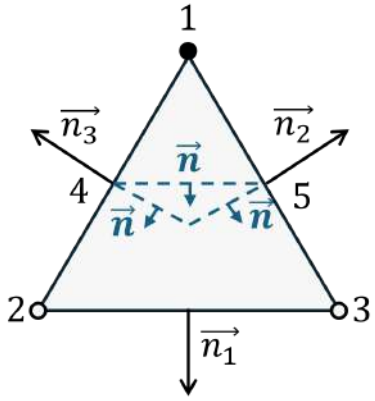


Figure 2.10: *Normal Vector $\vec{n}$ of ϑA_M corresponding to each triangle.*

$\frac{1}{2}Area[\triangle 123]$. Additionally, since the position vector gradient remains constant within each triangle, the normal vector relation holds as $\vec{n}_{45} = \frac{\vec{n}_{23}}{2}$. Generalizing this equation, let i denote a vertex of the triangle and j denote the edge opposite to vertex i. Then, the following relation holds:

$$\vec{n}_i = \frac{\vec{n}_j}{2} \tag{2.28}$$

where i,j =1,2,3.

Substituting Eqs. (2.26), (2.27) and (2.28) into Eq. (2.20), $H\vec{\underline{N}}_M$ at each node is obtained as follows:

$$\sum_{m=1}^{C_m} \frac{1}{2} \begin{bmatrix} \frac{\partial x}{\partial u} v_{\vec{n_1}} + \frac{\partial x}{\partial u} u_{\vec{n}_1} \\ \frac{\partial y}{\partial u} v_{\vec{n_2}} + \frac{\partial y}{\partial v} u_{\vec{n_1}} \\ \frac{\partial z}{\partial u} v_{\vec{n_3}} + \frac{\partial z}{\partial v} u_{\vec{n_3}} \end{bmatrix} = 2H\vec{\underline{N}}_M A_M \tag{2.29}$$

where $m = 1, .., C_m$ represents the neighboring triangular elements of node i, A_M is the area corresponding to node, and $\vec{\underline{N}}_M$ is the dimensionless mean curvature normal vector associated with node i.

5. *Compute the Nodal Mean Curvature*

Finally, H value of the i-th node of the mesh is computed by rewriting Eq. 2.29 as:

$$H_i = \frac{1}{2A_M} \sum_{m=1}^{C_m} \frac{1}{2} \begin{bmatrix} \frac{\partial x}{\partial u} v_{\vec{n_1}} + \frac{\partial x}{\partial u} u_{\vec{n}_1} \\ \frac{\partial y}{\partial u} v_{\vec{n_2}} + \frac{\partial y}{\partial v} u_{\vec{n_1}} \\ \frac{\partial z}{\partial u} v_{\vec{n_3}} + \frac{\partial z}{\partial v} u_{\vec{n_3}} \end{bmatrix} \cdot \vec{\underline{N}}_M \tag{2.30}$$

2.4.2 2nd Approach: Computation of the Position Vector Gradient Using Cotangent Formula

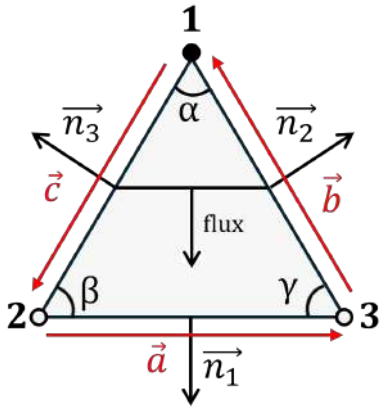


Figure 2.11: *Flux of a node in a triangular element.*

Using Eqs. (2.27) and (2.16) for the triangle shown in Fig. 2.11, the flux for node 1 through the $c'b'$ segment is defined as follows:

$$flux_1 = \nabla_s \vec{r} \cdot \frac{\vec{n}_1}{2} = -\frac{1}{4A_T}(r_1\vec{n}_1 \cdot \vec{n}_1 + r_2\vec{n}_2 \cdot \vec{n}_1 + r_3\vec{n}_3 \cdot \vec{n}_1)$$
$$flux_1 = -\frac{1}{4A_T}(r_1\vec{a} \cdot \vec{a} + r_2\vec{a} \cdot \vec{b} + r_3\vec{a} \cdot \vec{c}) \tag{2.31}$$

It should be noted that the edge normal vectors $\vec{n}_1$, $\vec{n}_2$, and $\vec{n}_3$, as well as the tangent vectors $\vec{a}$, $\vec{b}$, and $\vec{c}$, are defined within the planar triangle. However, in Eq. (2.31), their definition in either 2D or 3D space is irrelevant, as only their inner product is of interest.

As shown in Appendix B:

$$\vec{a} \cdot \vec{b} = \vec{23} \cdot \vec{31} = -\vec{32} \cdot \vec{31} = -2A_T cot\gamma$$
$$\vec{a} \cdot \vec{c} = \vec{23} \cdot \vec{12} = -\vec{32} \cdot \vec{21} = -2A_T cot\beta \tag{2.32}$$

Substituting Eq. (2.32) into Eq. (2.31), one gets the following:

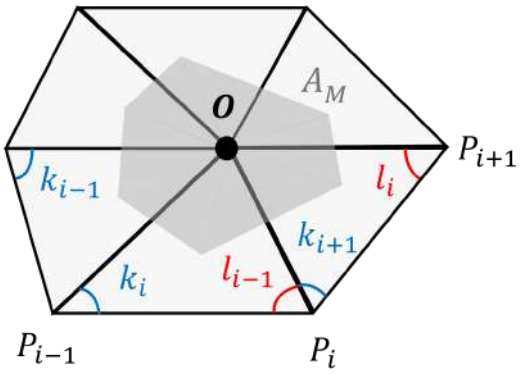


Figure 2.12: *Voronoi area for a surface mesh node.*

$$flux_1 = -\frac{1}{4A_T}\vec{r}_1(\vec{a}\cdot\vec{a}) + \frac{1}{2}(\vec{r}_2 - \vec{r}_1)cot\gamma + \frac{1}{2}(\vec{r}_3 - \vec{r}_1)cot\beta + \frac{1}{2}\vec{r}_1 cot\gamma + \frac{1}{2}\vec{r}_1 cot\beta \tag{2.33}$$

Then, in Eq. 2.33, some of the terms are further expanded as follows:

$$\begin{aligned} &-\frac{1}{4A_T}\vec{r}_1(\vec{a}\cdot\vec{a}) + \frac{1}{2}\vec{r}_1 cot\gamma + \frac{1}{2}\vec{r}_1 cot\beta \\ &= \frac{1}{2}\vec{r}_1[-\frac{1}{2A_T}(\vec{23}\cdot\vec{23}) + \frac{\vec{32}\cdot\vec{31}}{2A^T} + \frac{\vec{21}\cdot\vec{23}}{2A^T}] \\ &= \frac{1}{4A^T}\vec{r}_1(-\vec{23}\cdot\vec{23} + \vec{32}\cdot\vec{31} + \vec{21}\cdot\vec{23}) \end{aligned} \tag{2.34}$$

Given that $\vec{23} = \vec{21} + \vec{13}$, the last term in Eq. 2.34 is expanded as $-\vec{23}\cdot\vec{21} - \vec{23}\cdot\vec{13} + \vec{32}\cdot\vec{31} + \vec{21}\cdot\vec{23} = 0$. Therefore, the contribution of $\nabla_s^2\vec{r}$ to node 1 is given by the remaining terms of 2.35 and the final expression of flux at node 1 is given by:

$$flux_1 = \frac{1}{2}(\vec{r}_2 - \vec{r}_1)cot\gamma + \frac{1}{2}(\vec{r}_3 - \vec{r}_1)cot\beta \tag{2.35}$$

Node O of the surface mesh shown in Fig. 2.12 is examined. The *Laplacian* of the position vector of node O gets a contribution from all neighbor triangles:

$$\nabla_s^2\vec{r}\Big|_o = \sum_{i=1}^{C_m} \frac{1}{2}\left(\vec{r}_{p_i}\big|_m - \vec{r}_{p_o}\right)\cot l_i + \frac{1}{2}\left(\vec{r}_{p_{i+1}}\big|_m - \vec{r}_{p_o}\right)\cot k_{i+1} \tag{2.36}$$

where C_m the number of neighbor triangles of node O.

Eq. 2.36 can otherwise be expressed as the sum of the contributions of all edges emanating from node O:

$$\nabla_s^2 \vec{r}\Big|_o = \frac{1}{2} \sum_{i=1}^{C_g} (\vec{r}_{p_i} - \vec{r}_{p_o}) \left[\cot k_i + \cot l_i\right] \tag{2.37}$$

where C_g the number of the aforementioned edges and $\cot k_i$ and $\cot l_i$, the cotangents of the angles opposite to edge i of the two triangles that share this edge.

Overall, this method concludes with the following formula for H for each node of the surface mesh:

$$H_i = \frac{1}{2A_M} \nabla^2 \vec{r}\Big|_o \underline{\vec{N}}_M \tag{2.38}$$

which is similar to the formula given in [15].

In this method, H is computed using the geometric properties of the triangular elements, specifically the edge lengths and angles. Therefore, $\underline{\vec{N}}_M$ and A_M corresponding to each node must be expressed in terms of the same geometric quantities. In the VCFV methods defined in the previous subsection, the Barycentric Definition assumes that the portion is constant and equals 1/3. However, the Voronoi definition can be expressed differently from Eq. (2.41), as shown by the following formula from [15]:

$$A_M = \frac{1}{8} \sum_{m=1}^{C_m} \left(\vec{r}_{p_i}\big|_m - \vec{r}_{p_o}\right)^2 \cot l_i + \left(\vec{r}_{p_{i+1}}\big|_m - \vec{r}_{p_o}\right)^2 \cot k_{i+1} \tag{2.39}$$

where nodes p_i and p_{i+1}, and angles k_{i+1} and l_i are shown in Fig. 2.12.

The proof of Eq. (2.39) is given in Appendix B. It can also be expressed as the sum of the contribution of edges emanating from node O:

$$A_M = \frac{1}{8} \sum_{i=1}^{C_g} (\vec{r}_{p_i} - \vec{r}_{p_o})^2 \left[\cot k_i + \cot l_i\right] \tag{2.40}$$

From Eq. (2.39), the contribution of each triangle to the node area can be derived as:

$$A_{i_m} = \frac{1}{8} \left(\vec{r}_{p_i}\big|_m - \vec{r}_{p_o}\right)^2 \cot l_i + \left(\vec{r}_{p_{i+1}}\big|_m - \vec{r}_{p_o}\right)^2 \cot k_{i+1} \tag{2.41}$$

Similarly to Eq. (2.15) in the first approach, the Voronoi area formula in 2.41 is valid only for acute triangles. For obtuse triangles, a correction must be applied, as described in Section 2.3.3.

The normal vector of each node is given from Eq. (2.16) and Eq. (2.13) that were given in the previous approach.

2.4.3 Comparison of 1st and 2nd Approach

In this section, the accuracy of the two approaches for computing H at each node of the triangulated surface is compared. In addition, the three different definitions of VCFV are evaluated in terms of their effectiveness in determining A_M and $\vec{N}_M$ at each node.

To perform the comparisons, the different formulas are applied to two cases in which H value is analytically known.

A) 1^{st} *Case: Computation of H at a single nodal surface mesh*

In this application, an example surface is used, which is defined by:

$$\vec{x} = (u+v)\vec{e}_1 + (u-v)\vec{e}_2 + (uv)\vec{e}_3 \tag{2.42}$$

where u and v are the local coordinates of the surface, e_1, e_2, e_3 the basis vectors of the 3D coordinate system and $\vec{x}$ the position vector of the node. According to [13], H of this surface at the point located at $u = 1$ and $v = 1$ in local coordinates is equal to $H = \frac{1}{8\sqrt{2}}$.

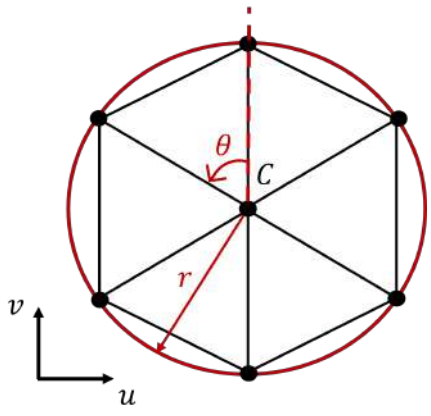


Figure 2.13: *Triangular nodal mesh consisting of one central node and six neighboring nodes on a circle.*

The surface is approximated using an unstructured mesh with triangular elements, where u and v represent the local coordinates of the triangle to which

the node belongs. Specifically, a simple mesh is generated around the node by subdividing a circle of radius r and center c, within the parametric space, into N_t triangular elements, Fig. 2.13. The coordinates of the neighbor nodes of the center node $(1, 1)$ are computed as follows:

$$\theta = \frac{2\pi i}{N_t}, \quad u = u_c + r\cos\theta, \quad v = v_c + r\sin\theta \tag{2.43}$$

where i is the index of the neighboring node, θ is the angle corresponding to the edge that the i-th node belongs to, measured counterclockwise, and (u_c, v_c) are the local coordinates of the center node.

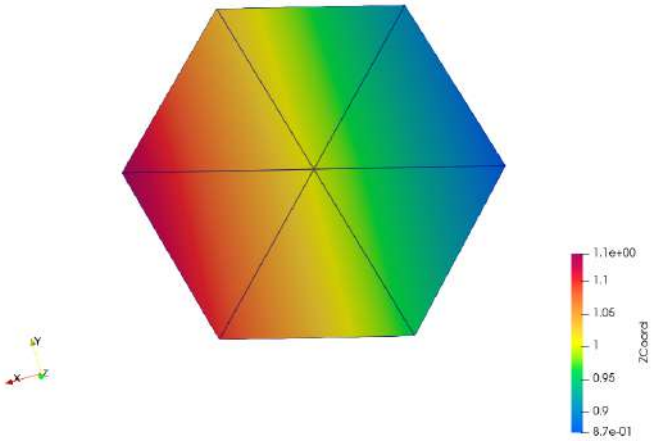


Figure 2.14: *Nodal Surface in 3D space, colored by the z-coordinate.*

Applying Eq. (2.42) to Eq. (2.43), the coordinates of the nodes are transformed from the local coordinate system to the global one, and the nodal surface for $(u_c, v_c) = (1, 1)$, $r = 0.1$ and $N_t = 6$ is generated, Fig. 2.14.

Firstly, the two approaches for computing the position vector gradient and, consequently, H at each node, are examined. The three different methods of defining the VCFV are used and compared for their accuracy against the analytical result. To evaluate their accuracy, an error measure is used as follows:

$$\text{Error} = \frac{|H_{\text{computed}} - H_{\text{true}}|}{|H_{\text{true}}|} \tag{2.44}$$

Obviously, only H at the central node, which is shared by all six neighboring

triangles, is considered.

VCFV Definition	**Mean curvature H_e**	**Error**
Barycentric	0.088277175363	0.131%
Voronoi	0.088235571286	0.173%
Corrected Voronoi	0.088235571286	0.173%

Table 2.1: *Comparison of VCFV definitions for the* 1^{st} *case using the* 1^{st} *approach, with the expected value to be* $H_e = \frac{1}{8\sqrt{2}}$.

VCFV Definition	**Mean curvature H_e**	**Error**
Barycentric	0.088272728665	0.131%
Voronoi	0.088235588150	0.173%
Corrected Voronoi	0.088235588150	0.173%

Table 2.2: *Comparison of VCFV definitions for the* 1^{st} *case using the* 2^{nd} *approach, with the expected value to be* $H_e = \frac{1}{8\sqrt{2}}$.

In this case, the Barycentric definition demonstrates higher accuracy, while the Voronoi and Corrected Voronoi definitions produce identical results due to the absence of obtuse triangles. Additionally, the two approaches used for computing the position vector gradient yield nearly identical results. To draw a more comprehensive conclusion about the accuracy of the different VCFV definitions, a larger surface case is examined.

B) 2^{nd} *Case: Computation of* H *on a spherical surface mesh*

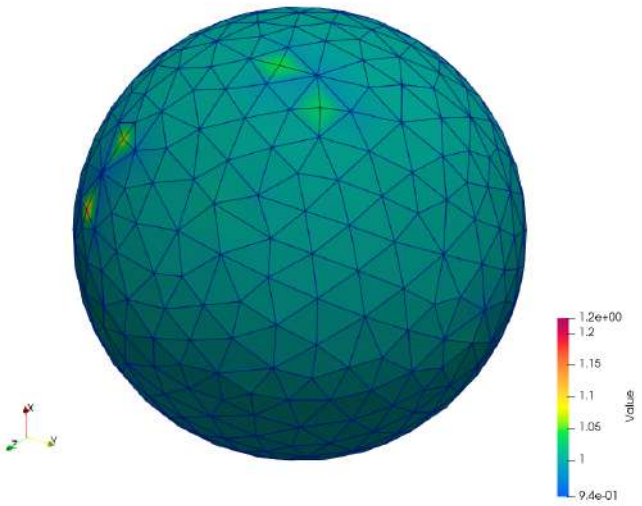


Figure 2.15: *H distribution on a spherical surface mesh using* 1^{st} *approach.*

In this case, a spherical surface mesh composed of 522 nodes and 1040 triangular elements is generated, Fig. 2.15. The sphere is assumed to have a unit radius ($R = 1$), which means that the expected H at each node is equal to one. To compare the different definitions, the accuracy of the computed maximum and minimum H values across the mesh nodes is examined.

VCFV Definition	**Min Value of Mean Curvature**	**Error**	**Max Value of Mean Curvature**	**Error**
Barycentric	0.758948018446	24.105%	1.69839015932	68.398%
Voronoi	0.765953911577	24.305%	1.000518315891	0.052%
Corrected Voronoi	0.941681791673	5.832%	1.222835972957	22.283%

Table 2.3: *Comparison of VCFV definitions for the* 2^{nd} *case using the* 1^{st} *approach, with the expected value to be* $H_e = 1$.

VCFV Definition	**Min Value of Mean Curvature**	**Error**	**Max Value of Mean Curvature**	**Error**
Barycentric	0.758949773150	24.105%	1.689380160545	68.938%
Voronoi	0.756967168630	24.303%	1.000725975853	0.073%
Corrected Voronoi	0.941683572258	5.832%	1.222835972957	22.286%

Table 2.4: *Comparison of VCFV definitions for the* 2^{nd} *case using the* 2^{nd} *approach, with the expected value to be* $H_e = 1$.

This case, along with the previous observations, demonstrates that both approaches for computing the position vector gradient yield similar results. This observation supports the statement that the second approach is essentially an extension of the first, differing only in the form of the final expression, where cotangent formula is used for area and gradient of position vector computation. Consequently, the only difference in the performance between the two approaches lies in the computational time. The second approach avoids computing the normal vectors $\vec{n}_j$ at the edges of each node for H estimation, resulting in significantly lower computational time, Fig. 2.16, with respect to mesh size. Therefore, the second approach, based on the cotangent formula for mean curvature computation, is selected for its effectiveness and superior efficiency.

In addition, it is shown that the Voronoi definitions yield better accuracy than the Barycentric one. However, it is not evident whether the Corrected Voronoi definition provides a significant improvement over the standard Voronoi approach. To better evaluate the effectiveness of the correction, a larger test case is examined: a spherical surface mesh consisting of 65404 nodes and 130804 triangular elements, Fig. 2.17. Table 2.5 demonstrates that the Corrected Voronoi definition reduces the maximum error, but it simultaneously increases

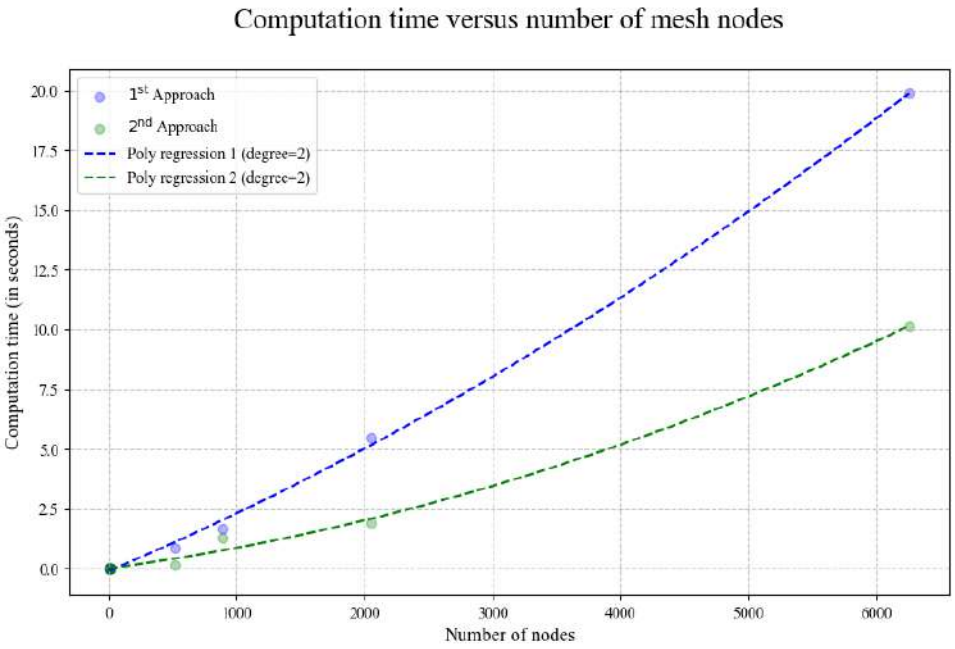


Figure 2.16: *Diagram of computation time in respect of the number of mesh nodes.*

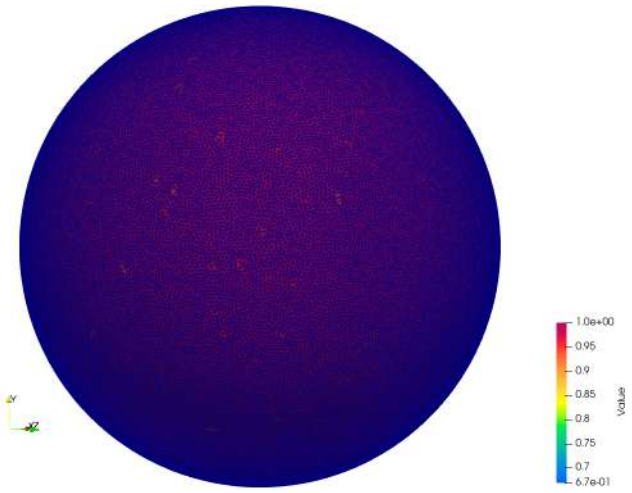


Figure 2.17: *H distribution on a spherical surface with a denser mesh using* 1^{st} *approach.*

VCFV Definition	Min Value of Mean Curvature	Error	Max Value of Mean Curvature	Error
Voronoi	0.667805595388	33.219%	1.000006092205	0.001%
Corrected Voronoi	0.855245023023	14.475%	1.172605969246	17.261%

Table 2.5: *Comparison of VCFV definitions for the 2^{nd} case with a denser mesh and using the 1^{st} approach, with the expected value to be $H_e = 1$.*

the error in the maximum value of H. Consequently, it can be inferred that the Corrected Voronoi approach gives a lower average error compared to the original Voronoi definition. However, the accuracy of the Corrected Voronoi method remains insufficient and requires further improvement, which will be investigated in a subsequent section.

2.5 Improving the Computation Method of Mean Curvature

In Subsection 2.4.3, it was demonstrated that, among the methods suggested in the literature, the most effective and efficient approach to compute H at each node is to use the Voronoi area for the VCFV definition and the Cotangent Formula for the gradient of the position vector. In this section, the weaknesses of this method are demonstrated and further improvements are suggested.

Firstly, the two cases mentioned in subsection 2.4.3 are used to investigate the factors that reduce the accuracy of the method.

A) 1^{st} *Case: Computation of H at a node surrounded by triangular elements*

In this case, various combinations of r and N_t are tested, and their results are compared in terms of accuracy in capturing the expected value at the central node (1,1).

In Table 2.6, it is shown that increasing the number of triangles in the nodal mesh requires a larger radius of the local coordinate system's circle to compute H with higher accuracy. This behavior can be attributed to the fact that, as more triangles surround a node, the internal angles at the central node become more acute which means that their cotangent are bigger. According to Eq. (2.38), H depends on the gradient of the position vector and the nodal area, both of which are expressed in terms of the cotangent of the triangle angles and the lengths of their edges, as shown in Eqs. (2.36) and (2.41). Consequently, to maintain a balanced curvature measure as the number of triangles and the cotangent of their angles increase, the edge lengths must also increase. To achieve this, the radius of the surrounding cycle needs to be appropriately

Number of triangles N_t	Radius r	Mean curvature H	Error
4	0.01	0.000000000001	100%
4	0.1	0.000000000000	100%
4	1	0.000000000000	100%
5	0.01	0.072800299907	17.627%
5	0.1	0.067947125457	23.127%
5	1	0.042823296080	51.551%
6	0.01	0.048223929680	11.511%
6	0.1	0.088235588150	0.173%
6	1	0.071299164710	19.334%
8	0.01	0.109777805277	24.199%
8	0.1	0.109319938405	23.681%
8	1	0.087570226432	0.926%
10	0.01	0.125554136686	45.023%
10	0.1	0.119588786695	35.299%
10	1	0.094460160490	6.869%
12	0.01	0.121375158747	37.320%
12	0.1	0.124723147891	41.108%
12	1	0.097167953030	9.933%

Table 2.6: *H error of 1^{st} case for various values of r and N_t.*

enlarged.

Furthermore, Table 2.6 shows that for each triangle count, there exists a specific radius that yields the lowest error. This relationship is further explored by running simulations with varying numbers of triangles and various radii to identify the combinations that produce the desired H at the central node. Each triangle count corresponds to a specific radius at which the computed curvature matches the target value, Fig. 2.18. It is important to note that for fewer than six triangles, the curvature vs. radius curves do not intersect the target value. This indicates that with fewer than six peripheral nodes, it becomes more challenging to accurately capture H at the center. One way to address this limitation is to rotate the nodal mesh about its center so that the peripheral nodes align more closely with the surface geometry, thereby improving the accuracy of the curvature estimation at the central node. This approach is examined for small numbers of triangles and a relatively small cycle radius of $r = 0.005$, Fig. 2.19.

These observations show that in a mesh, having fewer than six triangles around a node can reduce the accuracy of H for two reasons: first, it's physically hard

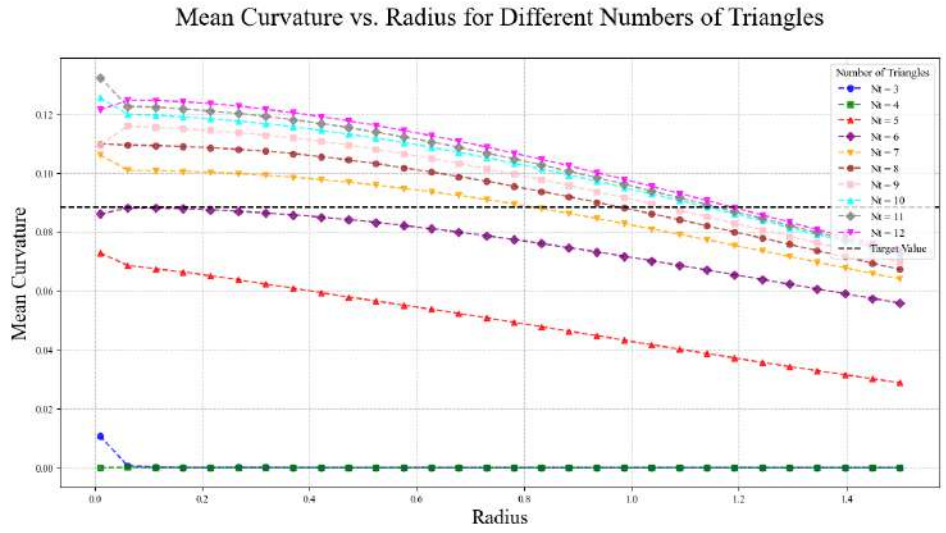


Figure 2.18: *H values at the central point of a nodal mesh for different numbers of triangles and radii.*

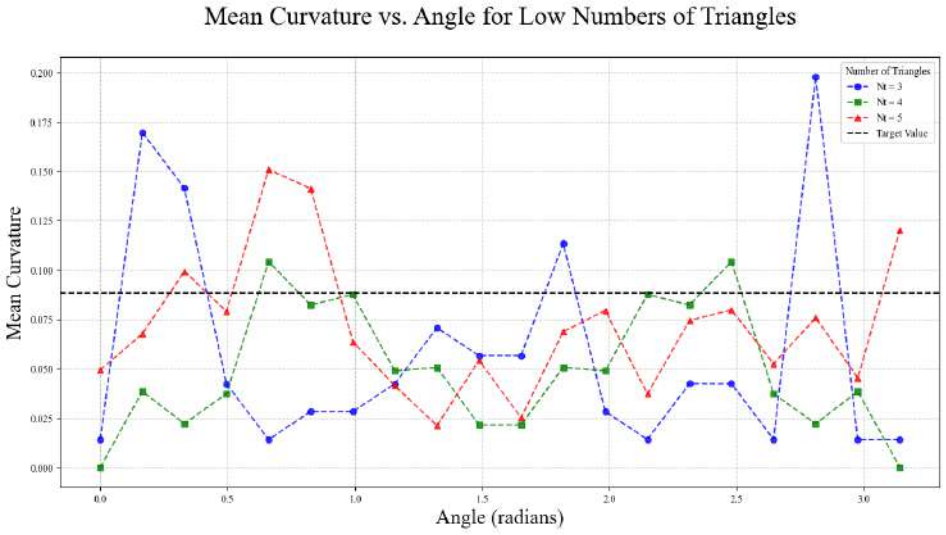
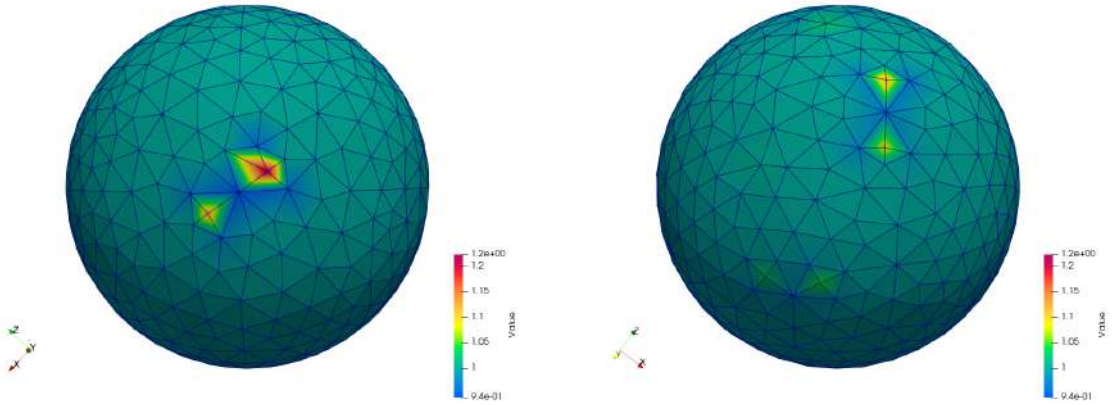


Figure 2.19: *H values at the central point of a nodal mesh for low numbers of triangles and radius* $r = 0.005$.

to capture the shape of a surface with so few triangles; second, the method used to calculate the nodal area can give wrong results in the presence of obtuse triangles. These problems will be explored further in the next case.

B) 2^{nd} *Case: Computation of* H *on a spherical surface mesh*

This section examines the spherical surface mesh with radius $R = 1$, Fig. 2.15, to investigate the causes of the deviation of H from its expected value of 1. The nodes with the largest errors are identified and displayed in Fig. 2.20. These



(a) 1^{st} *perspective view of nodes with deviation in* H*.*

(b) 2^{nd} *perspective view of nodes with deviation in* H*.*

Figure 2.20: *Different perspectives of view of nodes with deviation in* H *from the expected value.*

nodes are shown to belong to at least one obtuse triangle and are surrounded by fewer than six neighboring triangles, supporting the observations made in the previous case.

Of special interest is the example shown in the upper right of Fig. 2.20b, where two nodes with four neighboring triangles appear identical in terms of geometry and structure, yet their computed H differ. Upon closer analysis, it becomes evident that these two cases are not entirely identical, as they exhibit small but significant geometric differences, specifically in the angles of the obtuse neighboring triangles. More precisely, the node with the higher error has an obtuse angle of 104°, while the node with the lower error has an obtuse angle of 100°, which indicates that the larger the obtuse angle, the greater the error in computing H. These observations suggest that the mean curvature computation model is highly sensitive to small changes in the angles of obtuse triangles, a point that will be explored further in the following subsections. Since the challenge of accurately capturing surface curvature with too few surrounding nodes is not easily addressed, this section explores potential improvements to the computational model. Specifically, it separately examines the behavior of the position vector gradient and the Voronoi area formulations in the presence of obtuse triangles.

2.5.1 Behavior of the Position Vector Gradient Model in Obtuse Triangles

For the computation of the position vector gradient, Eq. (2.39) is employed. This formula evaluates the flux passing through the boundaries of the VCFV, as discussed in Section 2.4.2. The method used to compute the flux associated with each triangular element is applicable to all triangle types, including obtuse triangles. This indicates that Eq. (2.36), along with the integral form of the *Laplace–Beltrami* operator, remains valid even in the presence of obtuse angles. Therefore, no modifications are strictly required for the computation of the gradient of the position vector. However, in triangles with obtuse angles, the cotangent of those angles becomes negative. In Eq. (2.36), one might interpret the cotangent terms as weights assigned to the edges of the triangle. This becomes questionable for obtuse triangles, as negative weights lack a clear physical interpretation. Therefore, changes to Eq. (2.36) are considered to better handle cases with obtuse angles.

1. *Evenly distributing the weight of an obtuse angle among the other angles*

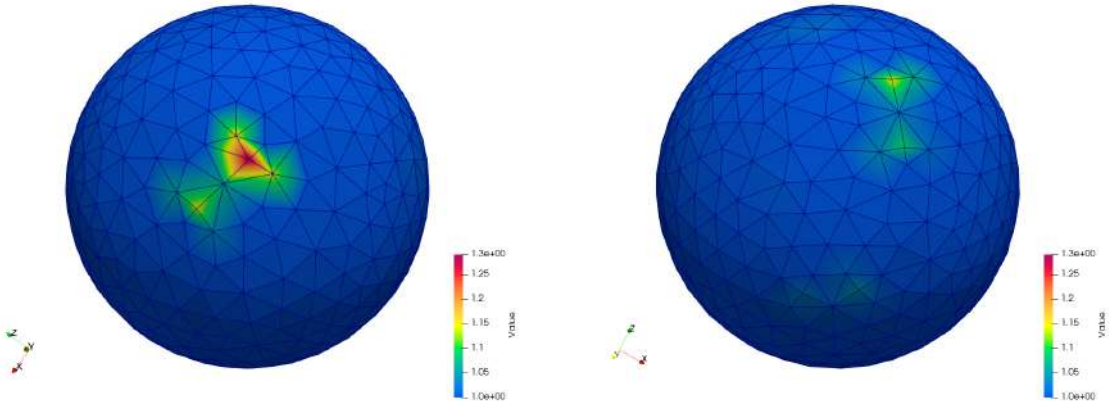
   The first approach handles an obtuse angle in a triangle by setting its cotangent to zero and redistributing its value equally to the cotangents of the other two angles. This ensures that all weights remain non-negative while preserving their total sum. For example, consider a triangle $\triangle ABC$, where α is the obtuse angle at vertex A, and β and γ are the acute angles at vertices B and C, respectively. In this case, the adjusted cotangent values are given by:

$$\begin{cases} \cot\beta = \cot\beta + \frac{\cot\alpha}{2} \\ \cot\gamma = \cot\gamma + \frac{\cot\alpha}{2} \\ \cot\alpha = 0 \end{cases} \tag{2.45}$$

   Eq. (2.45) is appropriately adapted when either β or γ is the obtuse angle. This method is applied to the second case of the spherical surface mesh described in Section 2.4.3, and the results are presented in Table 2.7.

Cotangent Weights	**Min Value of Mean Curvature**	**Error**	**Max Value of Mean Curvature**	**Error**
Original Method	0.941683572258	5.832%	1.222835972957	22.286%
Equal distribution to acute angles	1.000000983861	0.0001%	1.290301563957	29.030%

Table 2.7: *Comparison of the 1^{st} modification of Cotangent Weights with the Original Method in a spherical surface mesh, with the expected value to be $H = 1$.*



(a) 1^{st} *perspective view of nodes with deviation in* H.

(b) 2^{nd} *perspective view of nodes with deviation in* H.

Figure 2.21: *Different perspectives of view of nodes with deviation in* H *from the expected value after the* 1^{st} *cotangent weights modification.*

Table 2.7 shows that the average error in the computed H did not improve with this modification. Furthermore, the problematic nodes identified in Fig. 2.20 continue to produce the highest errors. The key difference is that both the minimum and maximum values of H have increased, Fig. 2.21.

2. *Distributing the weight of an obtuse angle proportionally to the cotangent values of the other angles*

   In the previous approach, the cotangent of the obtuse angle was evenly distributed to the two acute angles, regardless of the triangle's shape. In this approach, the distribution is weighted: each acute angle receives a portion of the obtuse angle's cotangent proportional to its own cotangent relative to the sum of the two acute cotangents. This way, the sharper the angle, the larger the share it receives.

   Similarly to Eq. (2.45), here the adjusted cotangent values are given by:

$$\begin{cases} S = \cot\beta + \cot\gamma \\ \cot\beta = \cot\beta + \frac{\cot\beta}{S}\cot\alpha \\ \cot\gamma = \cot\gamma + \frac{\cot\gamma}{S}\cot\alpha \\ \cot\alpha = 0 \end{cases} \tag{2.46}$$

   By applying Eq. (2.46) to the spherical surface mesh, the results presented in Table 2.8 are obtained. This modification yields slightly improved accuracy. To further validate this improvement, the denser spherical surface mesh, Fig. 2.17, is also examined, and the corresponding results are reported in Table 2.9.

Cotangent Weights	Min Value of Mean Curvature	Error	Max Value of Mean Curvature	Error
Original Method	0.941683572258	5.832%	1.222835972957	22.286%
Cot-based distribution to acute angles	1.000000983861	0.0001%	1.221313445349	22.131%

Table 2.8: *Comparison of the 2^{nd} modification of Cotangent Weights with the Original Method in a spherical surface mesh, with the expected value to be $H = 1$.*

Cotangent Weights	Min Value of Mean Curvature	Error	Max Value of Mean Curvature	Error
Original Method	0.855245023012	14.475%	1.1726059692463	17.261%
Cot-based distribution to acute angles	1.000000000012	10^{-9}%	10.085489146948	908.548%

Table 2.9: *Comparison of the 2^{nd} modification of Cotangent Weights with the Original Method in a denser spherical surface mesh, with the expected value to be $H = 1$.*

In this case, the error in the maximum H, Table 2.9, is significantly high. The nodes with the largest errors are mostly shared by four triangles, , Fig. 2.22, with the obtuse ones causing the biggest deviations. In addition, these obtuse triangles strongly influence nearby nodes, which exhibit large errors even when connected to six triangles. Consequently, this modification of the weights fails to improve the accuracy of the mean curvature computation.

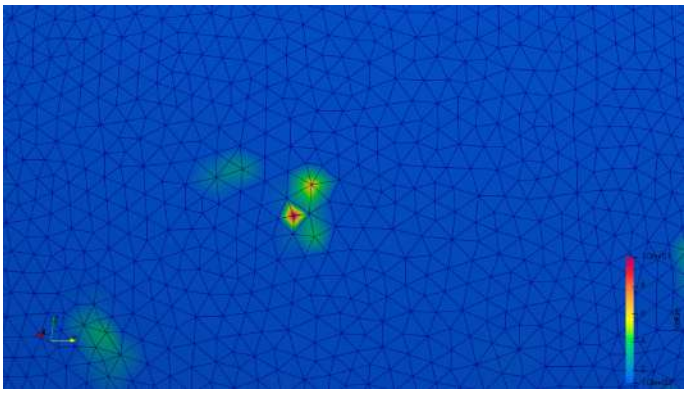
3. *Distributing the weight of an obtuse angle proportionally to the edges opposite to the other angles*

   In this approach, the distribution is weighted differently: each acute angle receives a portion of the obtuse angle's cotangent proportional to the length of the edge opposite it, relative to the sum of the lengths of the two opposing edges.

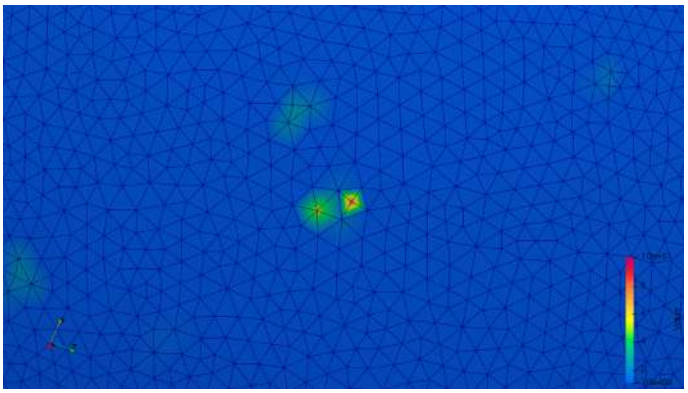
   In a similar way to Eq. (2.46), the cotangent values are modified as follows:

$$\begin{cases} S = L_{AB} + L_{AC} \\ \cot\beta = \cot\beta + \frac{L_{AC}}{S}\cot\alpha \\ \cot\gamma = \cot\gamma + \frac{L_{AB}}{S}\cot\alpha \\ \cot\alpha = 0 \end{cases} \tag{2.47}$$

   By applying Eq. (2.47) to the spherical surface mesh, the results presented in Table 2.10 are obtained. As shown, this modification does not improve the accuracy of the mean curvature computation. Similar to the first modification,



(a) 1^{st} *perspective view of nodes with deviation in* H*.*



(b) 2^{nd} *perspective view of nodes with deviation in* H*.*

Figure 2.22: *Different perspectives of view of nodes with deviation in H from expected value after* 2^{nd} *the cotangent weights modification.*

it increases both the maximum and minimum curvature values, while the error remains high for nodes that are shared by four triangles.

Cotangent Weights	**Min Value of Mean Curvature**	**Error**	**Max Value of Mean Curvature**	**Error**
Original Method	0.941683572258	5.832%	1.222835972957	22.286%
Edge-based distribution to acute angles	1.000000983861	0.0001%	1.400338639440	40.034%

Table 2.10: *Comparison of the* 3^{rd} *modification of Cotangent Weights with the Original Method in a spherical surface mesh, with the expected value to be* $H = 1$*.*

Consequently, none of the proposed modifications to the cotangent weights improved the accuracy of mean curvature computation. Based on these adjustments, it can be concluded that setting negative cotangent values to zero and redistributing their weight, as frequently proposed in the literature, did not reduce the deviation in computed H for meshes with obtuse triangles. Therefore, further improvement in the curvature computation depends on the behavior of the VCFV definition in the

presence of obtuse triangles, which will be investigated in the following subsection.

2.5.2 Behavior of the Voronoi Area Model in Obtuse Triangles

As shown in Section 2.4.3, the Corrected Voronoi Model is the most accurate VCFV definition identified so far. This model is generally based on treating an obtuse triangle as a right triangle, placing the circumcenter at the midpoint of the edge opposite the obtuse angle. However, this assumption still results in accuracy issues when obtuse triangles are present. This indicates the need to explore alternative methods for element area partitioning and subarea distribution to each node that belongs to obtuse triangle, which will be examined later in this section. Since the Corrected Voronoi definition performs well for acute triangles, the proposed improvements by this diploma thesis will apply only to the obtuse triangles in the mesh, while the others will remain unchanged.

1. *Barycentric area distribution in obtuse triangles*

   In this approach, the Barycentric definition of VCFV, often favored in literature due to its simplicity, described in Section 2.3.1 is applied only in the case of the obtuse triangles of the mesh. This definition is often favored in the literature and widely used in software implementations due to its simplicity. This modification is applied in the spherical surface mesh of Fig. 2.15.

VCFV Definition for obtuse triangles	**Min Value of Mean Curvature**	**Error**	**Max Value of Mean Curvature**	**Error**
Corrected Voronoi	0.941683572258	5.832%	1.222835972957	22.286%
Barycentric	0.854380453757	14.562%	1.577471699278	57.747%

Table 2.11: *Comparison of the* 1^{st} *modification of VCFV definition for obtuse triangles with the original one in a spherical surface mesh, with the expected value to be* $H = 1$.

   Table 2.11 shows that this modification worsens the mean curvature computation at nodes associated with obtuse triangles. This suggests that assigning a smaller area contribution of $\frac{1}{3}$ to the vertex with the obtuse angle, instead of $\frac{1}{2}$, increases the error, indicating that equal or larger contributions should also be investigated.

2. *Fixed area portioning for Corrected Voronoi in obtuse triangles*

   This section examines different methods of distributing a triangle's area among its vertices. Specifically, it aims to determine the optimal portion assigned to the vertex with the obtuse angle, assuming the remaining area is equally

divided between the two acute-angle vertices. To achieve this, the minimum and maximum values of H on the two spherical surface meshes that were previously used, are examined for their proximity to the expected value of 1.

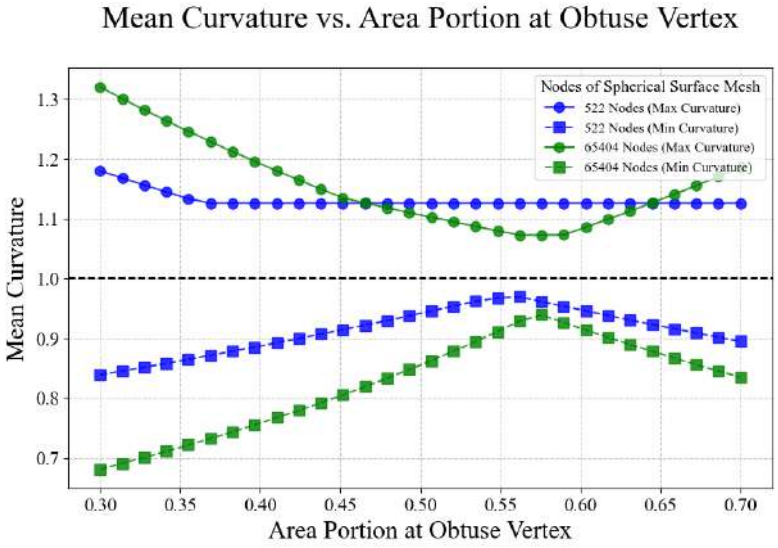


Figure 2.23: *H values on two spherical surface meshes for different area portions at obtuse vertices.*

There is a specific portion in each mesh that is higher than the portion of $\frac{1}{2}$ that was suggested in the Corrected Voronoi model, which minimizes the discrepancy between the minimum and maximum value of H from the target one, Fig. 2.23. However, this approach may not be optimal for each mesh, as it only considers the nodes with the highest errors while neglecting other nodes that belong to obtuse triangles. For a more precise investigation, all nodes belonging to obtuse triangles are examined to determine which area portion associated with the obtuse angle yields the expected H value.

There is a distinct area portion at the obtuse angle of each node that results in the computed H to be equal to the expected value, Fig. 2.24. It is also observed that as the obtuse angle of a node increases, the corresponding area portion of the obtuse triangle required to achieve accurate H also increases. Additionally, it is worth noting that for nodes 21 and 510, which have nearly identical obtuse angles, the required area portions are not exactly the same. This difference arises because the two nodes have different lengths for the edge opposite the obtuse angle, with the node having the longer edge requiring a larger portion of the area. That being said, it can be concluded that the optimal area portions assigned to each vertex are not constant for all obtuse triangles, but are strongly dependent on their specific shape. To validate this observation, a spherical surface mesh is examined by assigning to the obtuse-angle vertex the area portion that minimizes the error across most obtuse triangles. This optimal portion in the case of the spherical surface mesh is

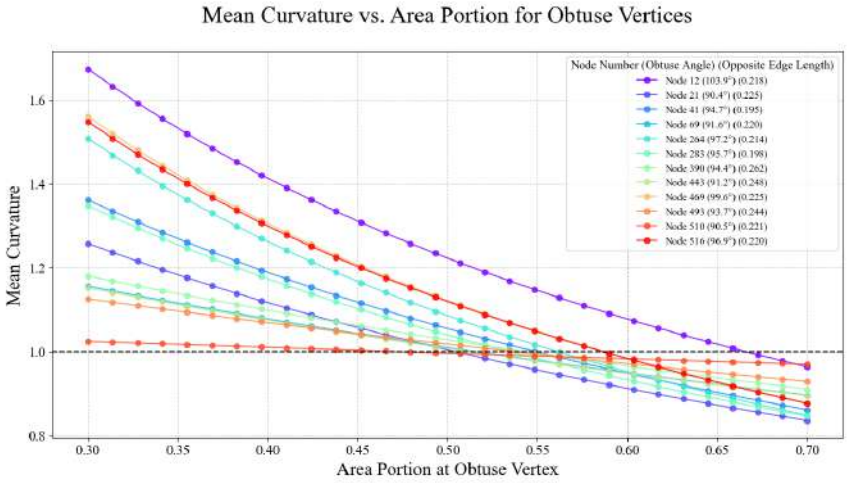


Figure 2.24: *H at nodes belonging to obtuse triangles in a spherical surface mesh of 522 nodes versus the area portion at obtuse vertex.*

approximately 0.57 for the vertex with the obtuse angle, Fig. 2.24, so the area distribution goes as follows:

$$A_A = 0.57 A_{\triangle ABC}, \quad A_B = 0.215 A_{\triangle ABC}, \quad A_C = 0.215 A_{\triangle ABC} \tag{2.48}$$

where A is assumed to be the obtuse angle in the triangle $\triangle ABC$. Using this distribution, the error in the maximum value of H is reduced, while the error in the minimum value increases, Table 2.12.

VCFV Definition for obtuse triangles	**Min Value of Mean Curvature**	**Error**	**Max Value of Mean Curvature**	**Error**
Corrected Voronoi	0.941683572258	5.832%	1.222835972957	22.286%
Fixed Portions for Area Distribution	0.937547225940	6.245%	1.117369313942	11.737%

Table 2.12: *Comparison of the 2^{nd} modification of VCFV definition for obtuse triangles with the original one in spherical surface mesh, with the expected value to be $H = 1$.*

The nodes with the maximum and minimum H are shown in Fig. 2.20(b). As illustrated, the node with the maximum curvature and the highest error (node 12) is a vertex shared by two triangles with obtuse angles. This indicates that the selected area portion for obtuse angles effectively reduces the error. This observation is further supported by Fig. 2.24, where the selected obtuse-angle portion at node 12 more closely matches the region intersecting the target value than before. On the other hand, the error at the node with the minimum curvature increases, suggesting that the selected area portion for acute angles

in obtuse triangles is suboptimal. To enable a more comprehensive comparison with the original Corrected Voronoi Method, the Relative Mean Absolute Error (RMAE) of the H should be introduced. This gives a clearer representation of the error distribution across all nodes, rather than focusing only on the extreme cases.

$$RMAE = \frac{1}{N} \sum_{i=1}^{N} \frac{|H_i^{computed} - H_i^{true}|}{|H_i^{true}|} \tag{2.49}$$

VCFV Definition for Obtuse Triangles	Relative Error
Corrected Voronoi	0.223%
Fixed Portions for Area Distribution	0.148%

Table 2.13: *RMAE of the 2^{nd} modification of the VCFV definition for obtuse triangles, compared to the original method on a spherical surface mesh.*

The relative mean absolute error improves when using the mean area portion for obtuse angles in order to reach target value, Table 2.14. To draw more general conclusions, the case of the denser spherical surface mesh is also examined.

VCFV Definition for Obtuse Triangles	Relative Error
Corrected Voronoi	0.059%
Fixed Portions for Area Distribution	0.047%

Table 2.14: *RMAE of the 2^{nd} modification of the VCFV definition for obtuse triangles, compared to the original method on a denser spherical surface mesh.*

In both cases, the RMAE of H is reduced, but this improvement does not hold for each node individually. Due to the local nature of the H, it is essential to achieve optimal accuracy at every point of the mesh. The results from the two cases suggest that, in obtuse triangles, assigning a greater portion than $\frac{1}{2}$ of the area to the vertex with the obtuse angle leads to more accurate curvature estimation. However, as demonstrated in the first case, distributing the remaining area equally between the acute angles does not yield equally effective results. Additionally, different triangles have different ideal portions for the obtuse vertex, indicating that a fixed portion is insufficient. These findings highlight the need for a new method that adapts the correction of the Voronoi area according to the specific geometry of each obtuse triangle.

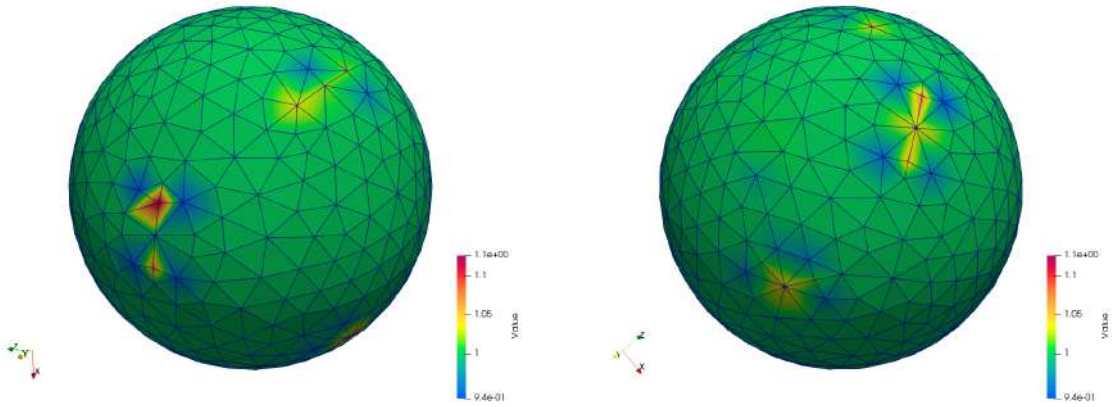
3. *Angle-based modification of Corrected Voronoi area distribution in obtuse triangles*

   In this approach, the area of the triangle is distributed to the obtuse vertex and the two acute-angle vertices based on the relative sizes of their angles,

rather than with a constant portion. Assuming that α is the obtuse angle at vertex A of triangle $\triangle ABC$, and β and γ are the acute angles at vertices B and C, respectively, the area assigned to each vertex is determined as follows:

$$\begin{cases} S = \alpha + \beta + \gamma \\ A_A = \frac{\alpha}{S} A_{\triangle ABC} \\ A_B = \frac{\beta}{S} A_{\triangle ABC} \\ A_C = \frac{\gamma}{S} A_{\triangle ABC} \end{cases} \tag{2.50}$$

Eq. (2.50) is appropriately adapted when either β or γ is the obtuse angle. The method is applied in the spherical surface mesh of Fig. 2.15 and the results of H measures are presented in Table 2.15.



(a) 1^{st} *perspective of view of nodes with deviation in* H.

(b) 2^{nd} *perspective of view of nodes with deviation in* H.

Figure 2.25: *Different perspectives of view of nodes with deviation in H from expected value after the* 3^{rd} *modification of Corrected Voronoi area.*

VCFV Definition for obtuse triangles	Min Value of Mean Curvature	Error	Max Value of Mean Curvature	Error
Corrected Voronoi	0.941683572258	5.832%	1.222835972957	22.286%
Angle-based modification of Corrected Voronoi	0.957099947443	4.290%	1.106193049107	10.619%

Table 2.15: *Comparison of the* 3^{rd} *modification of VCFV definition for obtuse triangles with the original one in a spherical surface mesh, with the expected value to be* $H = 1$.

The error in the maximum value of H is reduced, Table 2.15. However, a detailed analysis of the spherical surface mesh in Fig. 2.25 reveals that although

the maximum value improves, new nodes - previously unaffected under the Corrected Voronoi definition - now show significant errors. This raises concerns that the improvement may be coincidental rather than systematic. To investigate this further, the denser spherical mesh is also examined and its results are given in Table 2.16. As shown, while H error may improve at certain nodes with this modification, there is a risk that it may worsen at others. This indicates that the correction is not universally effective, and further investigation into alternative improvement methods is necessary.

VCFV Definition for obtuse triangles	**Min Value of Mean Curvature**	**Error**	**Max Value of Mean Curvature**	**Error**
Corrected Voronoi	0.8552450230	14.475%	1.172605969246	17.261%
Angle-based modification of Corrected Voronoi	0.7887844490	21.122%	1.089335782697	8.933%

Table 2.16: *Comparison of the 3^{rd} modification of VCFV definition for obtuse triangles with the original one in a denser spherical surface mesh, with the expected value to be $H = 1$.*

4. *Geometry-adaptive modification of Corrected Voronoi area distribution in obtuse triangles*

   In this approach, a new correction to the previously described Corrected Voronoi Method is proposed in this diploma thesis, taking into account the fact that the circumcenter of an obtuse triangle lies outside the triangle. In the existing correction discussed in Section 2.3.3, an obtuse triangle is treated as if it were a right triangle. While this approximation may be effective when the obtuse angle is close to 90^o, the accuracy of the Voronoi area computation decreases as the angle becomes more obtuse. To address this limitation, a new modification of the Corrected Voronoi Method is proposed. This enhanced correction adapts according to the degree of deviation of the obtuse triangle from a right triangle. Assuming an obtuse triangle $\triangle ABC$, with C being the vertex with obtuse node, the area distributed to each vertex is given as:

$$A_A = \frac{1}{4} p_A A_{\triangle ABC}, \quad A_B = \frac{1}{4} p_B A_{\triangle ABC}, \quad A_C = \frac{1}{2} p_C A_{\triangle ABC} \tag{2.51}$$

   where p_A, p_B and p_C are the corrective portions applied to the right triangle distribution in the case of obtuse triangles.

   To determine the appropriate correction, the area portions of the obtuse triangle are compared to those of its corresponding right triangle. Following the same model used for acute triangles, the sub-area associated with each vertex is composed of two triangular regions. Each of these regions is defined by a bisector, a radius R connecting the circumcenter to the vertex, and half of the

adjacent edge passing through that vertex. This geometric correspondence is illustrated analytically in Fig. 2.26.

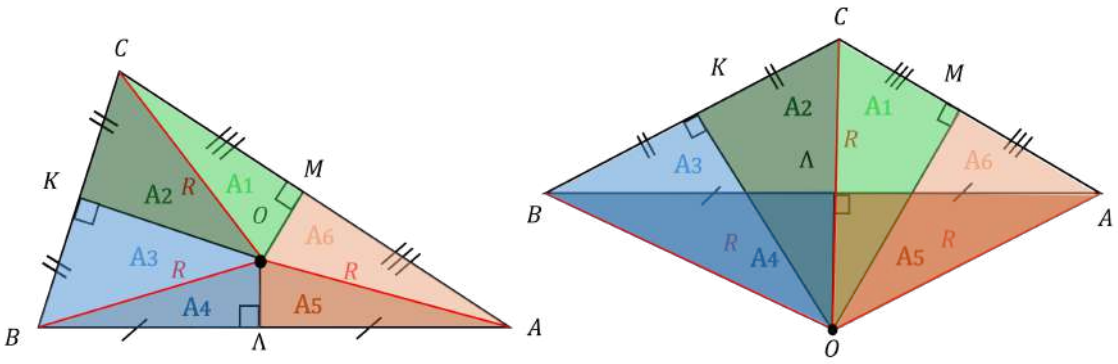


Figure 2.26: *Correspondence of sub-areas from acute to obtuse triangle.*

The correspondence to the right triangle is the same as for the obtuse one, with the key difference that point O lies at the midpoint of the hypotenuse and as a result areas A_4 and A_5 are not visible.

Now that the desired sub-area distribution has been defined, a correction based on a corresponding right triangle can be proposed. To do so, each acute vertex of the obtuse triangle is the approximation of a corresponding right triangle, Fig. 2.27.

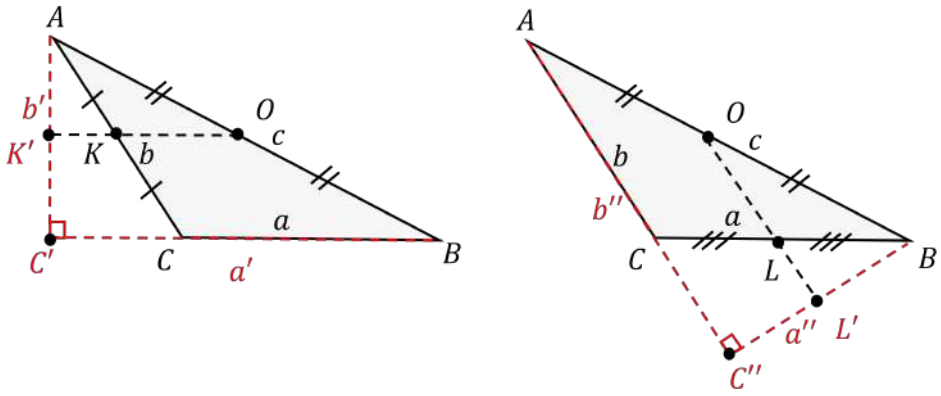


Figure 2.27: *Corresponding right triangles for the acute vertices A and B of the obtuse triangle $\triangle ABC$.*

For the vertex A of the obtuse $\triangle ABC$ and following Section 2.3.3, the area of $\triangle AOK$ is assigned. This can be viewed as the approximation of the sub-area $\triangle AOK'$ in the corresponding right triangle $\triangle ABC'$. For the vertex B of the obtuse $\triangle ABC$ as well, the area of $\triangle BOL$ is assigned. This can be viewed as the approximation of the sub-area $\triangle BOL'$ in the corresponding right triangle $\triangle ABC''$.

In the right triangle $\triangle ABC'$, the area distribution portion is defined as $\frac{Area[\triangle AOK']}{Area[\triangle ABC']}$. In order to find to appropriate correction p_1 for the corresponding vertex of the obtuse one, the correction of the sub-area of the vertex and the area of the triangle from the obtuse to the right triangle needs to be provided. The sub-area of vertex A and the area of the right triangle are given as follows:

$$A_{\triangle AOK'} = \frac{1}{2}\,|K'O| \cdot |AK'|, \quad A_{\triangle ABC'} = \frac{1}{2}\,|C'B| \cdot |AC'| \tag{2.52}$$

where

$$|AK'| = \frac{b}{2}\cos(\angle K'AK), \quad |K'O| = \sqrt{\frac{c^2}{4} - \frac{b^2}{4}\cos^2(\angle K'AK)}.$$

Given that

$$\angle K'AK = \frac{\pi}{2} - (\pi - \angle C) = \angle C - \frac{\pi}{2},$$

and denoting the deviation of the obtuse angle from the right as $\angle\Delta C$, we have:

$$|AK'| = \frac{b}{2}\cos(\angle\Delta C), \quad |K'O| = \frac{1}{2}\sqrt{c^2 - b^2\cos^2(\angle\Delta C)}.$$

Similarly,

$$|AC'| = b\cos(\angle\Delta C), \quad |C'B| = \sqrt{c^2 - b^2\cos^2(\angle\Delta C)}.$$

As a result, Eq. (2.52) becomes:

$$\begin{aligned} A_{\triangle AOK'} &= \frac{1}{8}b\cos(\angle\Delta C)\sqrt{c^2 - b^2\cos^2(\angle\Delta C)} \\ A_{\triangle ABC'} &= \frac{1}{2}b\cos(\angle\Delta C)\sqrt{c^2 - b^2\cos^2(\angle\Delta C)} \end{aligned} \tag{2.53}$$

The corresponding areas in the case of the obtuse triangle $\triangle ABC$ are given as:

$$A_{\triangle AOK} = \frac{1}{8}ab\cos(\angle\Delta C), \quad A_{\triangle ABC} = \frac{1}{2}ab\cos(\angle\Delta C) \tag{2.54}$$

The corrective portion p_A is derived by applying corrections to both the sub-area associated with vertex A sub-area and and the overall area of the triangle.

Therefore, by appropriately dividing the expressions in Eqs. (2.54) and (2.53), the correction for the vertex A is given as:

$$p_A = \frac{A_{\triangle AOK}}{A_{\triangle AOK'}} \cdot \frac{A_{\triangle ABC}}{A_{\triangle ABC'}} = \frac{a^2}{c^2 - b^2 \cos^2(\angle \Delta C)} \tag{2.55}$$

By following the exact same procedure for vertex B, its correction is given by:

$$p_B = \frac{A_{\triangle BOL}}{A_{\triangle AOL'}} \cdot \frac{A_{\triangle ABC}}{A_{\triangle ABC''}} = \frac{b^2}{c^2 - a^2 \cos^2(\angle \Delta C)} \tag{2.56}$$

Following the same approach as in Eqs. (2.55) and (2.56), and considering that the area attributed to vertex C in the corresponding right triangle consists of the sum of the sub-areas assigned to the other two vertices, it follows that the correction portion p_C is given by:

$$p_C = \frac{c^2}{(a^2 + b^2) \cos^2(\angle \Delta C)} \tag{2.57}$$

Portions p_A, p_B, and p_C represent the adjustments made to the sub-area distribution when an obtuse triangle is approximated as a right triangle. However, this approach neglects the fact that, in an obtuse triangle, the circumcenter lies outside the triangle. This causes areas A_4 and A_5 to appear behind the other area regions, Fig. 2.26. As a result, the sub-areas attributed to vertices A and B are underestimated, since they miss the parts distorted by this geometric folding, while the sub-area of vertex C is overestimated. To address this issue, additional corrections must be introduced to properly account for the redistribution of areas A_4 and A_5. Specifically, the overestimated portions of A_1 and A_2, which extend in front of A_4 and A_5, should be subtracted from the area attributed to vertex C. Meanwhile, the portions of A_3 and A_6 that lie in front of A_4 and A_5 should be added to the areas assigned to vertices A and B, respectively.

These additional corrections can be expressed as the ratio of the folded areas to the corresponding areas in the case of a right triangle. However, since these areas are not easily identifiable in the right triangle configuration, the corrections can alternatively be formulated as the product of the area attributed to each vertex in the obtuse triangle with the corresponding relative area when the triangle is right. More specifically, for vertex A, the correction corresponding to the folded region can be expressed as the ratio of area A_6, attributed to vertex A, to the total area of the triangle, multiplied by the corresponding portion in the right triangle case. To formulate this, the area A_6 is defined as follows:

$$A_{\triangle AOM} = \frac{1}{2}|MA| \cdot |OM| = \frac{1}{2}|MA|\sqrt{R^2 - |MA|^2} \tag{2.58}$$

Denoting the edges of the obtuse triangle in Fig. 2.26 as $|BC| = a$, $|AC| = b$, and $|AB| = c$, and using the expression for R given in Eq. (B.3), Eq. (2.58) becomes:

$$\begin{aligned} A_{\triangle AOM} &= \frac{1}{2} \cdot \frac{b}{2} \cdot \sqrt{\left(\frac{b}{2\sin(\angle B)}\right)^2 - \left(\frac{b}{2}\right)^2} \\ &= \frac{1}{8}b\sqrt{\frac{b^2(1-\sin^2(\angle B))}{\sin^2(\angle B)}} = \frac{1}{8}b^2\sqrt{\frac{\cos^2(\angle B)}{\sin^2(\angle B)}} \\ A_{\triangle AOM} &= \frac{1}{8}b^2\cot(\angle B) \end{aligned} \tag{2.59}$$

Using the cotangent formula of $\angle B$, Appendix C, Eq. (2.59) becomes:

$$A_{\triangle AOM} = \frac{1}{32}b^2\frac{a^2 + c^2 - b^2}{A} \tag{2.60}$$

The area of the obtuse triangle is given by the 2^{nd} Eq. of (2.54). Dividing Eq. (2.60) with Eq. (2.54), one gets:

$$\begin{aligned} \frac{A_{\triangle AOM}}{A_{\triangle ABC}} = \frac{1}{32}b^2\frac{a^2 + c^2 - b^2}{A^2} &= \frac{1}{32}b^2\frac{a^2 + c^2 - b^2}{\frac{1}{4}a^2b^2\cos^2(\angle \Delta C)} \\ \frac{A_{\triangle AOM}}{A_{\triangle ABC}} &= \frac{1}{8}\frac{a^2 + c^2 - b^2}{a^2\cos^2(\angle \Delta C)} \end{aligned} \tag{2.61}$$

As previously discussed, for a right triangle, the circumcenter O coincides with the midpoint of the edge opposite to the right angle. In this scenario, the sub-areas A_4 and A_5 vanish. However, this simplification poses a difficulty for defining additional corrections for vertex C, since the corresponding portion in the right triangle would be zero and the number of sub-areas reduce from six to four. To resolve this, the corresponding portion of A_6 in the right triangle is considered to be half of the sub-area attributed to vertex A. This sub-area is equal to $\frac{1}{8}$ of the total area of the right triangle. Taking this into account,

the corrective portion p_A must be adjusted by an additional correction due to the contribution of A_6, which lies outside the obtuse triangle. This correction is given by $\frac{1}{8} \cdot \frac{A_{\triangle AOM}}{A_{\triangle ABC}}$.

Therefore, the final corrected portion for vertex A, using Eqs. (2.55) and (2.61), is expressed as:

$$p_A = \frac{a^2}{c^2 - b^2 \cos^2(\angle \Delta C)} + \frac{1}{64} \frac{a^2 + c^2 - b^2}{a^2 \cos^2(\angle \Delta C)} \tag{2.62}$$

Similarly, for vertex B, the additional correction is expressed as the ratio of area A_3, attributed to vertex A, to the total area of the triangle, multiplied by the corresponding portion in the right triangle case. Following the same procedure as for the additional correction of vertex A, the following correction is obtained:

$$p_B = \frac{b^2}{c^2 - a^2 \cos^2(\angle \Delta C)} + \frac{1}{64} \cdot \frac{b^2 + c^2 - a^2}{b^2 \cos^2(\angle \Delta C)} \tag{2.63}$$

Similarly to vertices A and B, vertex C requires an additional correction due to portions of areas of A_1 and A_2 that lie outside the obtuse triangle and a and are incorrectly attributed to it. Applying the same methodology used for the other two vertices, it can be shown that the term to be subtracted from p_C has the same form as the corrections for p_A and p_B but with the appropriate substitution of geometric quantities corresponding to vertex C. The final corrective portion p_C is thus given as follows:

$$p_C = \frac{c^2}{(a^2 + b^2) \cos^2(\angle \Delta C)} - \frac{1}{64} \cdot \frac{a^2 + b^2 - c^2}{c^2 \cos^2(\angle \Delta C)} \tag{2.64}$$

By applying the proposed geometry-adaptive correction method to the previously discussed first case involving a spherical mesh, the results shown in Table 2.17 were obtained. As illustrated, a significant reduction in error is achieved when obtuse triangles are treated in accordance with their geometric characteristics. To enable a more comprehensive evaluation, the RMAE of the proposed method is compared with that of the original Corrected Voronoi Method for both the initial and the denser spherical surface meshes, as presented in Tables 2.18 and 2.19. In all cases, the RMAE is reduced by up to an order of magnitude. Furthermore, all nodes previously identified as outliers, due to their deviation from expected values, Fig.2.15, demonstrate notable improvement, which leads to a smoother distribution of H across the spherical mesh, Fig.2.28.

In order to achieve a smoother and more uniform distribution of H across the

VCFV Definition for obtuse triangles	Min Value of Mean Curvature	Error	Max Value of Mean Curvature	Error
Corrected Voronoi	0.941683572258	5.832%	1.222835972957	22.286%
GAC Voronoi	0.995123547932	**0.488%**	1.006219135819	**0.622%**

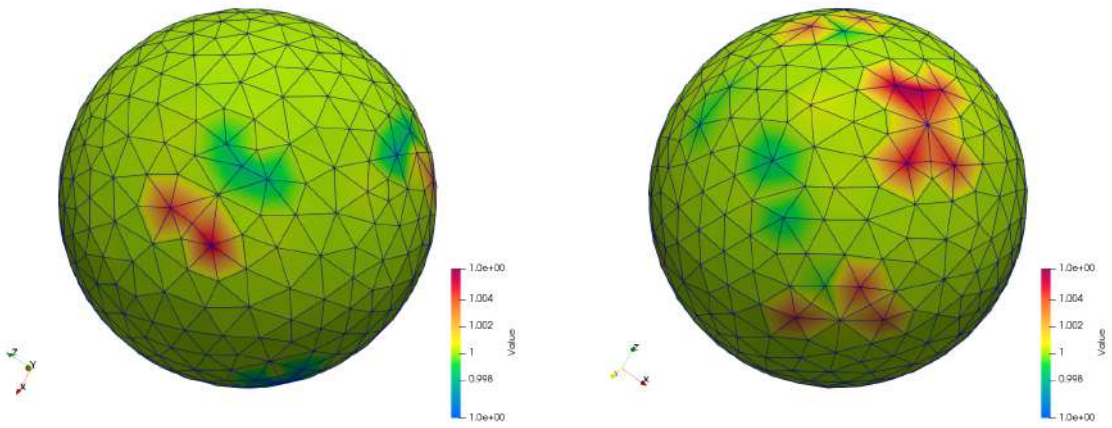
Table 2.17: *Comparison of the 4^{th} modification of VCFV definition for obtuse triangles with the original one in spherical surface mesh, with the expected value to be $H = 1$.*

VCFV Definition for Obtuse Triangles	Relative Error
Corrected Voronoi	0.223%
GAC Voronoi	**0.025%**

Table 2.18: *RMAE of the 4^{th} modification of the VCFV definition for obtuse triangles, compared to the original method on a spherical surface mesh.*

VCFV Definition for Obtuse Triangles	Relative Error
Corrected Voronoi	0.059%
GAC Voronoi	**0.0066%**

Table 2.19: *RMAE of the 4^{th} modification of the VCFV definition for obtuse triangles, compared to the original method on a denser spherical surface mesh.*



(a) 1^{st} *perspective view of nodes with deviation in H.*

(b) 2^{nd} *perspective view of nodes with deviation in H.*

Figure 2.28: *Different perspectives of view of nodes with deviation in H from expected value after 4^{th} modification of Corrected Voronoi.*

mesh, a smooth transition between area attribution formulas is introduced. Specifically, the transition between the Corrected Voronoi formula for acute triangles and the Geometry-Adaptive Corrected (GAC) Voronoi formula for obtuse triangles near the right-angle threshold is carefully treated. A smooth

blending is applied between the standard Corrected Voronoi area attribution and its geometry-adaptive version that accounts for obtuse triangle configurations. The objective is to avoid abrupt changes in area distribution when a triangle's internal angle approaches or slightly exceeds 90°.

To achieve this, a smooth transition function is defined using a sigmoid-type function centered at 90°. The transition is controlled by two parameters: steepness k, which controls the steepness of the transition function, and threshold, which is the central angle value (in degrees) at which the transition is balanced. When the angle at a vertex falls within a predefined interval around 90°, specifically $[88°, 92°]$, the method computes a weighted average of the area values obtained from the two methods. The transition weight t is calculated using a sigmoid-like function, defined as:

$$t = \frac{1}{1 + e^{-k \cdot (\theta - 90)}},$$

where θ is the vertex angle in degrees. The final area attributed to each vertex is given as a convex combination:

$$A_i = (1 - t) \cdot A_i^{\text{Voronoi}} + t \cdot A_i^{\text{Obtuse}},$$

ensuring smooth variation and numerical stability when involving obtuse triangles. For a moderately smooth transition, a value of $k = 2$ was used. By applying this transition, the error in minimum value of H is further improved, Table 2.20.

VCFV Definition for obtuse triangles	**Min Value of Mean Curvature**	**Error**	**Max Value of Mean Curvature**	**Error**
Corrected Voronoi	0.9416835723	5.832%	1.2228359730	22.286%
SGAC Voronoi	0.9980752485	**0.192%**	1.0062191358	**0.622%**

Table 2.20: *Comparison of the 4^{th} modification with smooth transition of VCFV definition for obtuse triangles with the original one in spherical surface mesh, with the expected value to be $H = 1$.*

As demonstrated, the Smoothed Geometry-Adaptive Corrected (SGAC) Voronoi Method achieves the highest level of accuracy. The SGAC Voronoi method is also important for the case of adjoint-based design optimization, as it reassures differentiability, when combining the Corrected Voronoi Method for the acute and right triangles with the GAC Voronoi Method for the obtuse ones. In a subsequent section it will be further illustrated that this method surpasses all others proposed in the literature. Therefore, it has been chosen as the foundation for the methodology employed in this study.

Chapter 3

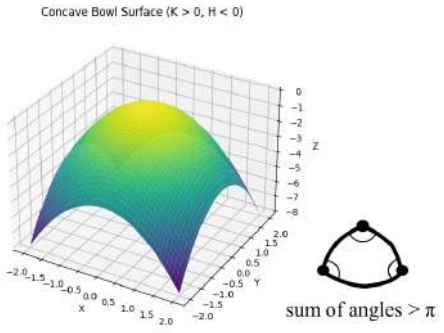
Computation of Gauss Curvature on Surface Meshes

3.1 Introduction

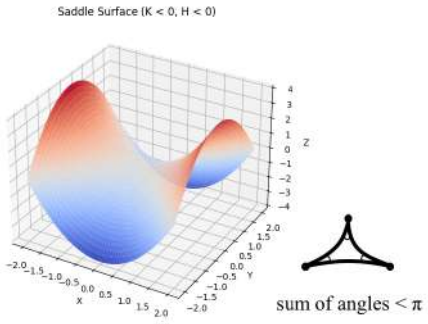
In this section, a quick introduction to the meaning of the Gauss curvature measure os a surface in the $\mathbb{R}^3$ space is provided. As described in Chapter 1, Gauss curvature of a point is the product of the two principal curvatures κ_1 and κ_2. To ultimately compute the principal curvatures κ_1 and κ_2 for determining the total curvature at each node, it is necessary to define both the mean curvature H and the Gaussian curvature K. Therefore, similarly to the mean curvature H, computational methods for estimating the Gauss curvature K must also be thoroughly investigated. In this work, the magnitude of Gauss curvature will be denoted as K and its sign depends on the signs of the principal curvatures, which can be defined from the way the normal vector $\vec{N}$ changes along tangent directions. For the smooth surface it measures the infinitesimal bending of the surface compared to the flat tangent plane. Instead of comparing the surface with the tangent plane, it can equally be considered as the turn of the normal vector along the surface. That being said, the sign of K conveys information about the local shape of a surface at a point. Positive K means that both principal curvatures have the same sign, which means that surface curves in the same direction along all direction. This indicates that the surface is locally convex or elliptic. $K = 0$ means that the surface is flat or cylindrical at that point, according to if one or both principal curvatures are zero. Finally, negative K means that he surface curves upward in one direction and downward in the perpendicular direction and indicates that the surface is saddle-shaped or hyperbolic at that point.

Regarding the physical meaning of the Gauss curvature can be obtained by consid-

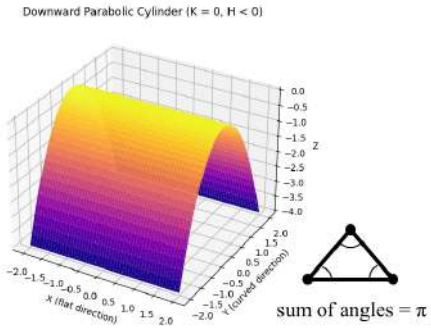
ering how triangles behave on different types of surfaces. More specifically, if $K = 0$ that means that a triangle embedded in the surface will have sum of angles equal to π. The cylindrical surface is behaving similarly to the flat one as regards the sum of triangle's angles. If K is positive, the surface bends outward like a dome, which means that an embedded triangle to it will have sum of angles greater than π. If K is negative, the surface bends like saddle, which means that an embedded triangle to it will have sum of angles less than π. These observations are demonstrated in the simple example of surface with mean curvature $H < 0$, but different signs of Gauss curvature K, Fig. 3.1.



(a) $K > 0$, $H < 0$ *(Bowl surface).*



(b) $K < 0$, $H < 0$ *(Saddle surface).*



(c) $K = 0$, $H < 0$ *(Parabolic Cylinder).*

Figure 3.1: *Surface shapes with different signs of Gauss curvature K.*

3.2 Approximation of Gauss Curvature on a Surface

This section explores methods for calculating K at the nodes of an unstructured mesh. Similar to H, K is invariant quantity for a structured mesh, meaning that it depends solely on surface geometry and remains unaffected to any change in parameterization. However, since an unstructured mesh cannot be parameterized

in a straightforward way, alternative computational methods must be considered. To achieve this, as described in Section 1.4 of [19], the Gauss map function must be introduced. This function $g : S \to \mathbb{S}^2$ maps each point p on the surface patch S to the tip of its normal vector $\mathbf{n}(p)$, translated onto the unit sphere $\mathbb{S}^2 \subset \mathbb{R}^3$, Fig. 3.2. All the image points constitute a region on the unit sphere, whose area is called image area and is equal to the total K of the patch. It is easy to see that the flatter the surface patch, the more its normal vectors at p and q align on the unit sphere, leading to a smaller image area consequently near- zero K. However, the reverse is not necessarily true, as even when the image area on the unit sphere is small, the patch may not be flat. This can happen when the normal vectors vary in direction and change sign across the patch, causing the Gauss map to fold the surface over itself and concentrate the image into a small region, despite the curvature being significant.

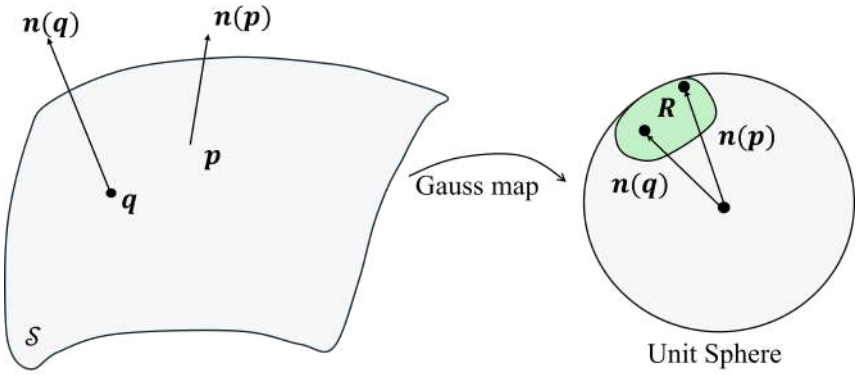


Figure 3.2: *Gauss map of the normal vectors at points on a patch S of a smooth surface onto the unit sphere $\mathbb{S}^2$.*

The above definition is extended to the case of a discretized surface by computing the Gaussian curvature at each mesh node. As described above, K over a domain $\Omega \subset S$ is given by the area of the image to the unit sphere, which is expressed as $k(\Omega) = Area(g(\Omega))$. In the case of the discretized surface, K is evaluated at each point, which means that the domain Ω here corresponds to a small area around a node, denoted as A_M. The normal vectors that are mapped to the unit sphere are those of the node's adjacent faces. As a result, the total K at point p shown in Fig. 3.3 is given by:

$$K(p) = \lim_{\text{diam}(A)\to 0} \frac{A_M^G}{A_M} \tag{3.1}$$

where A^G is the area of the region on the unit sphere formed by the image of the normals of faces around p.

As mentioned, in the discretized surface the curvature measure is defined in its nodes, while its elements, when triangles, are flat and thus contribute zero curvature. Since

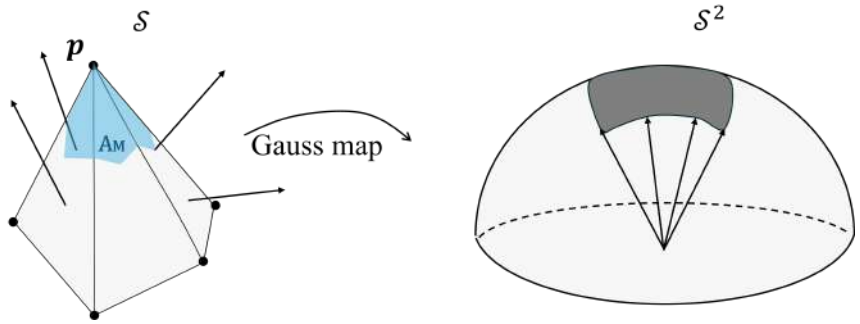


Figure 3.3: *Gauss map function of normal vectors of the faces adjacent to a point p on a discretized surface onto the unit sphere* $\mathbb{S}^2$.

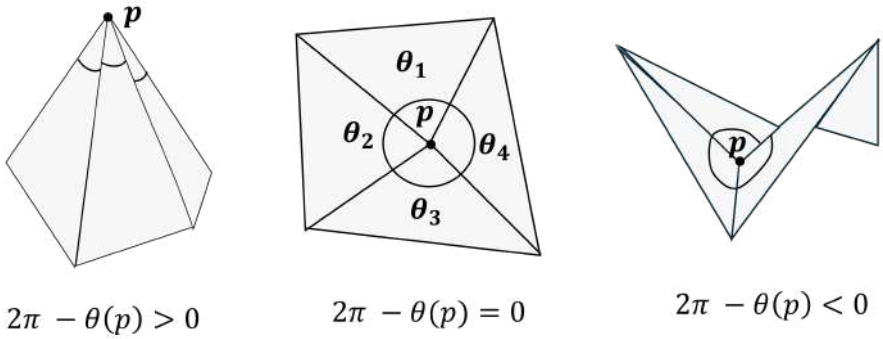


Figure 3.4: *Variation of local curvature around node p with respect to the vertex angle excess* $2\pi - \theta(p)$.

K is an invariant geometric quantity, its total value at each node remains constant even as the associated area A_M shrinks, as the topological structure is preserved. This leads to a fundamental issue in Eq. (3.1) as the denominator tends to zero, while the total curvature remains fixed, which makes the limit to diverge whenever the neighborhood is not locally flat. Therefore, Eq. (3.1) is not suitable for defining Gauss curvature on unstructured meshes and another definition of it is required.

To define discrete Gauss curvature, the total vertex angle definition needs to provided. If p is a point in surface S with N number of adjacent elements, then the vertex angle of the adjacent face f_i at vertex p is denoted as θ_i. Then, the total vertex angle is given as:

$$\theta(p) = \sum_{i=1}^{N} \theta_i(p) \tag{3.2}$$

If the local neighborhood (star) of a vertex is flat, the total vertex angle is equal to 2π. Otherwise, the sign of the vertex angle excess $2\pi - \theta(p)$ provides information about the local curvature at point p, Fig. 3.4.

In order to derive the formula of K at each point, the Gauss-Bonnet Theorem is used. This theorem has multiple expressions, but the one that is more useful in this scope is the Gauss-Bonnet Formula for an embedded triangle. Assuming a surface S in $\mathbb{R}^{\not\Vdash}$ and T to be a a triangle embedded in surface and ∂T to be the boundary of this triangle, then the Gauss-Bonnet Theorem is expressed as follows:

$$\iint_T K\, dA + \int_{\partial T} \kappa_g\, ds + \sum_{i=1}^{3} \alpha_i = 2\pi, \tag{3.3}$$

where κ_g is the geodesic curvature along the boundary ∂T, which is a measure that describes how much a curve on a surface deviates from being perfectly embedded to it, and α_i are the exterior jump angles at the three corners of the triangle.

If Eq. (3.3) is applied to a triangulated discrete surface, then each triangle T is considered flat. This implies that $K = 0$ almost everywhere inside each triangle. Consequently, the total K is concentrated only at the nodes of the mesh. As described in the definition of Eq. (3.1), the discrete Gauss curvature at a node is computed over its corresponding area A_M, which represents the area of its finite volume cell, Fig. 2.4. Therefore, Eq. (3.3) can be expressed for each node p as:

$$\iint_{A_M} K\, dA + \int_{\partial A_M} \kappa_g\, ds + \sum_{i=1}^{N_t} \alpha_i = 2\pi, \tag{3.4}$$

where N_t is the number of triangles adjacent to the node, and α_i denotes the exterior angle at each vertex of the polygonal boundary ∂A, Fig. 3.5.

The boundary ∂A_M is comprised by straight lines in the embedding space and the geodesic curvature along each edge is zero. Therefore, the boundary integral vanishes:

$$\int_{\partial A_M} \kappa_g\, ds = 0,$$

Hence, Eq. (3.4) reduces to:

$$\iint_{A_M} K\, dA + \sum_{i=1}^{N_t} \alpha_i = 2\pi \tag{3.5}$$

In order to express the angle α_i in terms of quantities from the triangular elements, it is necessary to specify which definition of the VCFV area is being used. As discussed in Section 2.4.3, the most accurate formulation is the GAC Voronoi area. This method defines the area as the standard Voronoi method when the triangles adjacent

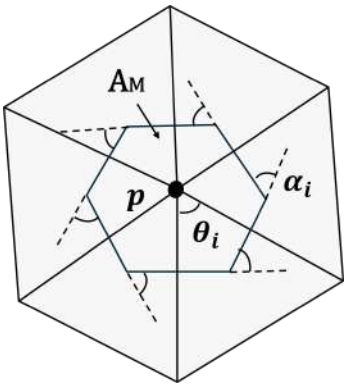


Figure 3.5: *Exterior jump angles α_i and interior angles θ_i of Voronoi area A_M that corresponds to node p.*

to the node are non-obtuse, while for obtuse triangles, it applies corrections to the sub-area distributions assigned to each vertex. In the context of mean curvature computation, these corrections primarily aim to improve the accuracy of the node-associated area, which directly influences the precision of the curvature calculation, Eq. (2.38). This is achieved by adjusting the way in which the area of each triangle is distributed among its vertices. Consequently, the corrective approach focuses on redistributing sub-areas without altering the definition of the circumcenter, even in the case of obtuse triangles.

Therefore, it is acceptable to use the original Voronoi area definition for both obtuse and non-obtuse triangles in order to derive the K formula. In this definition, the polygonal A_M of the node is constructed by connecting the midpoints of edges and the circumcenters of the adjacent triangles, Fig. 3.5. As a result, when this construction is applied to the Voronoi region, the external angles along each edge are zero, since the boundary segments remain perpendicular to the corresponding triangle edges. Furthermore, the external angle at each circumcenter is equal to the interior angle θ_j of the corresponding triangle at vertex p, since it is formed by edges that are perpendicular to the edges forming θ_j.

Using these observations, the formula for K at a node p can be expressed as:

$$\iint_{A_M} K \, dA = 2\pi - \sum_{j=1}^{N_t} \theta_j \tag{3.6}$$

where θ_j are the interior angles at vertex p across all adjacent triangles. Given that, as discussed, K at a node p remains constant over its associated area A_M, Eq. (3.6) can be rewritten as:

$$K \iint_{A_M} dA = 2\pi - \sum_{j=1}^{N_t} \theta_j$$

$$K(p) = \frac{1}{A_M}(2\pi - \sum_{j=1}^{N_t} \theta_j) \tag{3.7}$$

or in a simpler form:

$$K(p) = \frac{1}{A_M}(2\pi - \theta(p)) \tag{3.8}$$

where $\theta(p)$ is the total vertex angle of node p, Eq. (3.2).

Eq. (3.7) can also be written as a sum over the edges incident to node p, where each angle θ_j is assigned to one of the two edges of the element that define it, as follows:

$$K(p) = \frac{1}{A_M}(2\pi - \sum_{j=1}^{N_e} \theta_j) \tag{3.9}$$

where N_e is the number of edges incident to node p.

In Eqs. (3.7) and (3.9) the contribution to total vertex angle from each triangle can be derived as follows:

$$\theta_j = \arccos\left(\frac{(\vec{r}_{p_j} - \vec{r}_{p_o}) \cdot (\vec{r}_{p_{j+1}} - \vec{r}_{p_o})}{\left|\vec{r}_{p_j} - \vec{r}_{p_o}\right| \left|\vec{r}_{p_{j+1}} - \vec{r}_{p_o}\right|}\right) \tag{3.10}$$

3.3 Comparison of the VCFV Definitions

Now that the final formula for Gauss curvature computation has been specified, the computation methods are going to be analyzed. More specifically, in Eq. (3.7), K, similarly to H, is heavily influenced by the nodal area A_M, as in both cases it appears in the denominator, a positioning that significantly affects the accuracy of the result. To further verify the conclusions drawn in Section 2.5 regarding the optimal method for defining the VCFV, the five different approaches of Barycentric, Voronoi, Corrected Voronoi, GAC Voronoi and SGAC Voronoi area will be compared in terms of their accuracy in Gauss curvature computation. The five methods will be compared based on their error in the maximum and minimum K values, as well

as their Relative Mean Absolute Error (RMAE), when applied to the two spherical mesh cases presented in Section 2.4.3. As it is well known, the K of a unit sphere with radius $R = 1$ is expected to be $K = 1$ at every point, Fig. 3.6. The reason for analyzing the errors at the extreme cases separately from the RMAE is that K, just like H, is a local property of a surface. This means it must be computed as accurately as possible at each individual node.

Tables 3.1 and 3.2 show that, among the different methods, the Geometry-Adaptive approaches demonstrate the highest accuracy. There is no significant difference in the error between the Smoothed and the original GAC Voronoi method. This is attributed to the fact that only 1.9% of the triangles in the mesh are nearly right-angled, resulting in a negligible impact on the final result. The case of the denser spherical mesh yields similar results. Indicatively, the comparison of the RMAE for the Geometry-adaptive methods is presented in Table 3.3, demonstrating that the Smoothed variant gives practically the same accuracy in the computation of the Gauss curvature.

VCFV Definition for obtuse triangles	**Min Value of Gauss Curvature**	**Error**	**Max Value of Gauss Curvature**	**Error**
Barycentric	0.764502115632	23.520%	1.696782700131	69.678%
Voronoi	0.762505006729	23.750%	1.008347475541	0.835%
Corrected Voronoi	0.948572762938	5.143%	1.228222737202	22.822%
GAC Voronoi	1.002448622644	0.245%	1.011539116559	1.154%
SGAC Voronoi	1.002448622644	**0.245%**	1.011539116560	**1.154%**

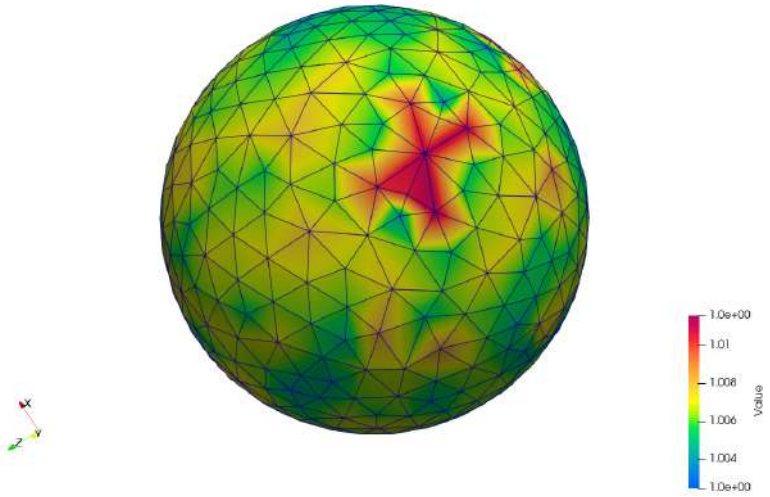
Table 3.1: *Comparison of different VCFV definitions for K on a spherical surface mesh, with the expected value to be $K = 1$.*

VCFV Definition for obtuse triangles	**Relative error**
Barycentric	7.237%
Voronoi	0.833%
Corrected Voronoi	0.774%
GAC Voronoi	**0.613%**
SGAC Voronoi	**0.616%**

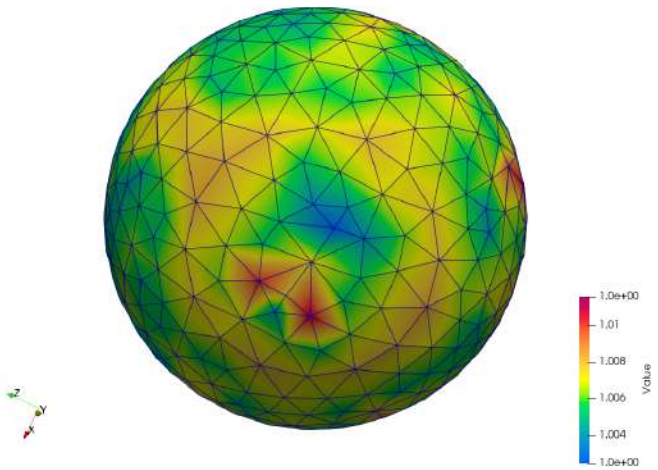
Table 3.2: *RMAE comparison of different VCFV definitions for K on a spherical surface mesh.*

VCFV Definition for obtuse triangles	**Relative error**
GAC Voronoi	0.896%
SGAC Voronoi	**0.870%**

Table 3.3: *RMAE comparison of different VCFV definitions for K on a denser spherical surface mesh.*



(a) 1^{st} *perspective view of nodes with deviation in* K .



(b) 2^{nd} *perspective view of nodes with deviation in* K.

Figure 3.6: *Different perspectives of view of nodes with deviation in* K.

Chapter 4

Computation of Curvature Measures on Quadrilateral Meshes

4.1 Introduction

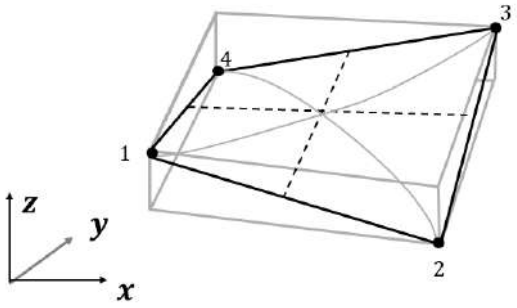


Figure 4.1: *3D quadrilateral element.*

This chapter examines several techniques for computing Gauss and mean curvature on quadrilateral meshes. The main objective is to generalize the curvature estimation methods that were initially created for triangular meshes and introduced in earlier chapters to other kinds of meshes. Since quadrilateral meshes can be both structured and unstructured, they provide a useful test case for this generalization. While many curvature computation approaches for quadrilateral meshes are adapted from those used for triangular elements, a key distinction arises: unlike triangles, which are inherently planar, quadrilateral elements may not lie on a single plane, Fig. 4.1. This property introduces a new challenge in the computation, as

the elements themselves may exhibit non-zero curvature. To address this, a method for properly defining VCFV and computing the required differential operators are investigated, which are based on triangulating the quadrilateral elements effectively.

4.2 Triangulation Methods

An easy and effective way to handle quadrilateral meshes is to subdivide each quadrilateral into two triangles and then apply methodologies developed for triangular meshes. However, a key challenge lies in selecting the triangulation method that yields the highest accuracy for computed geometric quantities. As discussed in [26] and [14], different triangulations of the same quadrilateral mesh can result in varying values for both mean and Gauss curvature. This observation is further investigated here through a simple example where the exact values of these curvatures are known.

Suppose a surface S lying upon a sphere of radius $R = 1$, Fig. 4.2. The mean and Gauss curvature at central point p, which is tangent to the sphere, should be $H = 1$ and $K = 1$. Fig. 4.3 presents the different triangulation strategies used to compute curvature at point p. Both globally consistent and alternating triangulations across the entire mesh that give specific properties to central node are examined. It is important to note that each quadrilateral can be divided in two ways, which means that there exist 2^n possible triangulation combinations for a mesh of n quadrilaterals, as for example different combinations with consistent triangulations over a line or a row in the mesh, and examining all of them is impractical and out of the scope of this part of study.

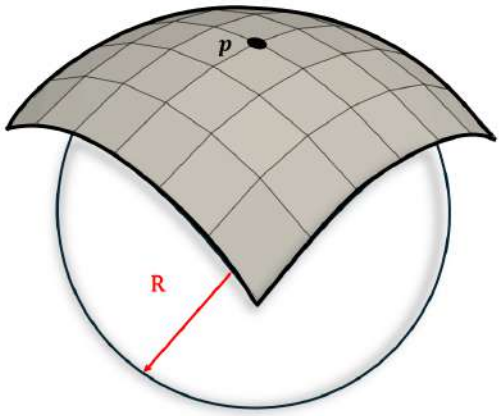
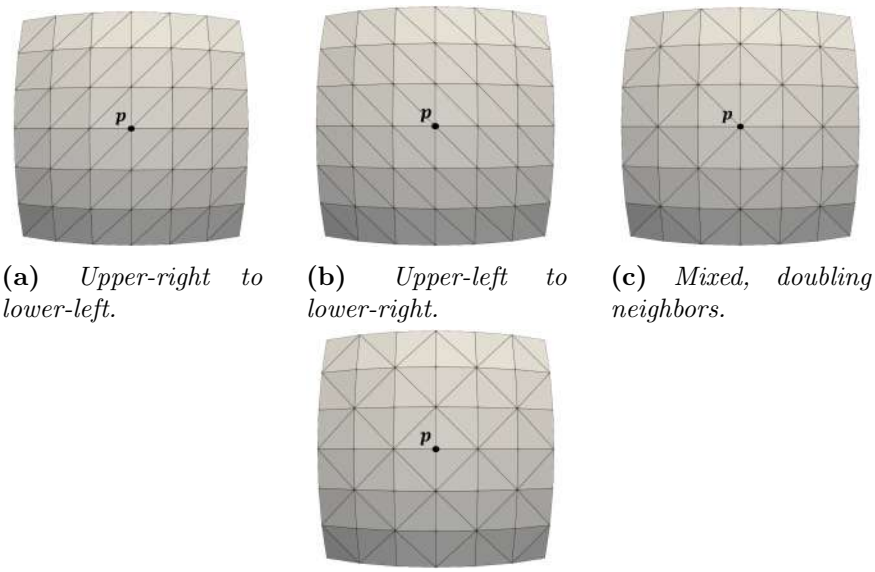


Figure 4.2: *Unit sphere with surface S tangent to it at point p.*

The strategies illustrated in Figs. 4.3a and 4.3b aim to produce a uniform triangulation of the mesh, which in this case results mixed type of triangulation for the central point p. However, the approaches shown in Figs. 4.3c and 4.3d focus on alternating



(a) *Upper-right to lower-left.*

(b) *Upper-left to lower-right.*

(c) *Mixed, doubling neighbors.*

(d) *Mixed, preserving neighbors.*

Figure 4.3: *Different triangulation patterns applied to the same surface mesh for curvature computation at point p.*

triangulations for the entire mesh, targeting a particular number of neighbors for central point p. While it is possible to target specific number of neighbor triangles to every point in the mesh, this is beyond the scope of the current study, as only the point p, tangent to the sphere, has known expected curvature values and is therefore examined to assess the accuracy of the computed results. H and K are computed according the methodology presented in Chapters 2 and 3, using the SGAC Voronoi method and their errors are presented in Table 4.1.

Triangulation methods	**Mean curvature H_p error**	**Gauss curvature K_p error**
Method of 4.3a	0.140%	0.733%
Method of 4.3b	0.140%	0.733%
Method of 4.3c	0.280%	0.760%
Method of 4.3d	0.002%	0.706%

Table 4.1: *Accuracy of H and K at the central point p for different methods of triangulation.*

There are important conclusions derived from Table 4.1. Firstly, as it is shown in the first two triangulations, curvature measure in symmetrical geometries is independent of the directions of the diagonals splitting when uniformly divided. To confirm this conclusion, the same test is done in a more complex geometry, like a saddle. A saddle surface, defined by $z = x^2 - y^2$, exhibits zero mean curvature along the diagonals $y = x$ and $y = -x$, which connect opposite corners of the domain boundary. Fig.

4.4 shows one of the two uniform triangulations applied to the saddle surface and the resulting mean curvature computed across the mesh.

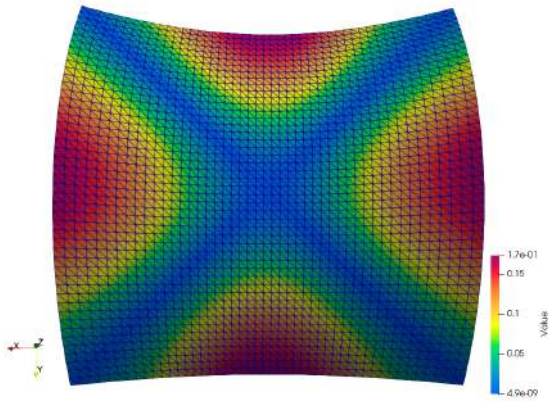


Figure 4.4: *Mean curvature deviation in a triangulated quadrilateral mesh in saddle surface.*

Triangulation methods	Maximum error of mean curvature	Mean error of mean curvature
Upper-left to lower-right	0.003%	0.001%
Upper-right to lower-left	0.003%	0.001%

Table 4.2: *Accuracy of H at the diagonals of saddle for different methods of triangulation.*

The mean and maximum error computed in the diagonals of the saddle are compared for two different directions of uniform triangulation of the saddle in Table 4.2. As shown, the two methods give in this case give same results for the two triangulations, which confirms the previous observation.

Another observation from Table 4.1 is that certain triangulation methods yield better results for a given geometry. Moreover, the triangulation that provides the best accuracy for mean curvature also performs best for Gauss curvature. To ensure that this observation is not coincidental for the specific example, the same test is conducted for the case examined in Section 2.4.3, where the H and K values are evaluated on a nodal surface mesh defined by the surface equation given in Eq. (2.42). The expected values of mean and Gauss curvature at node $(1, 1)$ are $H = \frac{1}{2\sqrt{8}}$ and $K = \frac{1}{16}$, respectively. For two different mixed triangulations of the surface, similar to those of Figs. 4.3c and 4.3d, with the outer and inner radii of the nodal star of quadrilaterals, show in Fig. set to $R = 1.2$ and $r = 1$, respectively, the resulting values are presented in Table 4.3.

Both mean and Gauss curvatures have better performance at the same type of triangulation, Table 4.3. Another important observation from the surface examples presented in Tables 4.1 and 4.3 is that the most accurate triangulation method

Triangulation methods	Mean Curvature	Error	Gauss Curvature	Error
Similar to Fig. 4.3c	0.109720795125	24.135%	0.097661200251	19.334%
Similar to Fig. 4.3d	0.071299164710	56.257%	0.058407593841	6.548%

Table 4.3: *Accuracy of H and K at the nodal star of quadrilaterals for different triangulation methods.*

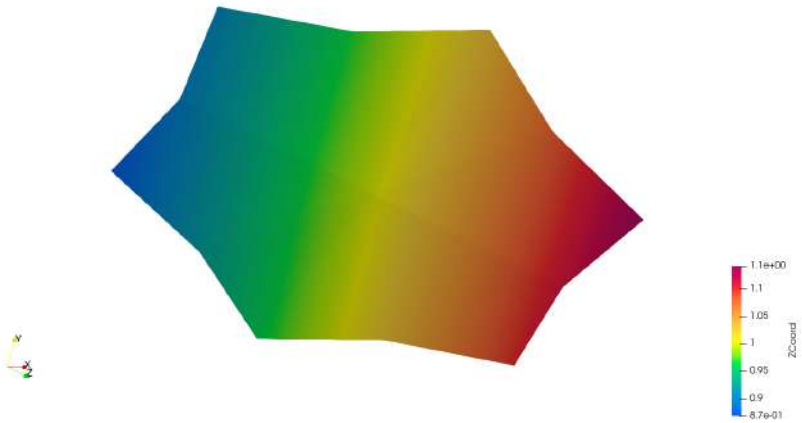


Figure 4.5: *Nodal star of quadrilateral elements.*

at each node appears to be the one in which the number of triangles associated with a node matches the number of surrounding quadrilaterals. To ensure this investigation, another surface equation is also examined for the central node of a nodal star of quadrilaterals of $R = 0.12$ and $r = 0.1$, which is given as follows:

$$\vec{x} = (\cosh u \cos\theta)\vec{e_1} + (\cosh u \sin\theta)\vec{e_2} + u\vec{e_3} \tag{4.1}$$

where u and θ are the surface parameters. For this surface, the expected value of the H is zero at every point. In Table 4.4, the error in the computed H at node $(1, 1)$ is examined. As shown, in this case as well, the triangulation method that preserves the number of elements around the node yields the most accurate results.

Triangulation methods	Mean Curvature	Error
Similar to Fig. 4.3c	0.000386735572	0.039%
Similar to Fig. 4.3d	0.000315361906	0.032%

Table 4.4: *Accuracy of mean curvature at nodal star of quadrilaterals for different triangulation methods, with the expected value to be $H = 0$.*

Based on the examples examined, it can be concluded that the best practice of triangulation of a quadrilateral mesh is node-specific. In the cases analyzed, the best

results were obtained when the quadrilaterals around each node were divided in a way that preserved the number of neighboring elements. However, this observation cannot be generalized to all quadrilateral meshes. Furthermore, as previously noted, both mean and Gauss curvatures are intrinsic properties of a surface and should remain invariant under changes in mesh resolution or triangulation. Therefore, a general method for computing these curvatures should be used, which could account for the different triangulation possibilities of each quadrilateral. To this end, the average of the two possible triangulations is used as the basis for the computational methods, which will be explored in the following sections.

4.3 VCFV Definition on Quadrilateral Meshes

As was shown in Chapters 2 and 3, both mean and Gauss curvature computations are dependent on the way the VCFV is defined. In this section, the way that the average VCFV is derived for the average of the triangulation methods, Fig. 4.6. More specifically, the Voronoi area is defined as the mean value of the sub-areas of each element that occur from two different triangulations. For the central node, the first triangulation attributes both triangles to it, while the second attributes only the one that belongs to. As a result, the mean Voronoi area is given as follows:

$$\overline{A_M} = \sum_{t=1}^{N_t} \left(\frac{A_1 + A_2 + A_3}{2} \right)_t \tag{4.2}$$

where N_t is the number of triangles that the central node belongs to. Eq. 4.2, the resulting mean Voronoi area is the same whether it is calculated by first averaging the sub-areas of each individual element and then summing them, or by computing the Voronoi area for each triangulation and then taking the average.

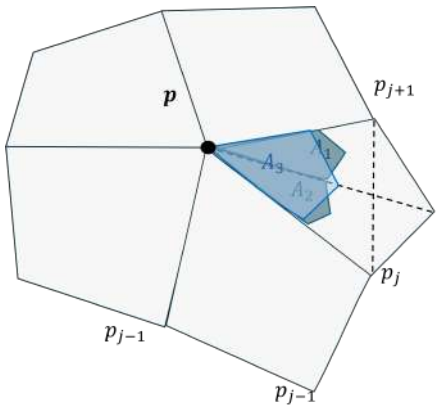


Figure 4.6: *Voronoi sub-areas resulting from two different triangulation methods applied to each element.*

4.4 Mean Curvature on Quadrilateral Meshes

The mean curvature at a triangular mesh node is computed based on Eq. 2.1, which shows that it largely depends on the application of the *Laplace–Beltrami* operator to the position vector, along with the normal vector at that node. To proceed with the computation, the *Laplace - Beltrami* operator in quadrilateral elements need to be defined. While it may seem reasonable to extend the previously defined VCFV framework to this case, such an extension must also be proven.

Firstly, following the foundation that is provided in [26], the mean curvature can be defined as follows:

$$H(p) = -\lim_{diam(A)\to 0} \frac{1}{2}\frac{\nabla A(p)}{A(p)} \tag{4.3}$$

where $A(p)$ is the area attributed to node p on the surface, diam(A(p)) the diameter of the region of area $A(p)$ and $\nabla A(p)$ is the gradient of $A(p)$ with respect to the surface parameters.

Using Eq. 4.2, the gradient of the area around the node p, can be expressed as:

$$\overline{\nabla A_M} = \frac{1}{2}\sum_{t=1}^{N_t}(\nabla A_1 + \nabla A_2 + \nabla A_3) \tag{4.4}$$

Applying Eq. (2.39) in Eq. (4.4) the gradient of each sub-area is expressed as follows:

$$\nabla A_j = \frac{1}{4}\left((\vec{r}_{pj} - \vec{r}_{p_o})\cot l_j + (\vec{r}_{p_{j+1}} - \vec{r}_{p_o})\cot k_{i+j}\right) \tag{4.5}$$

If w_j is denoted as the weight that correspond to the edge $\vec{r}_{po} - \vec{r}_{pj}$ from the first triangulation, while w'_j is the weight that corresponds to the imaginary edge $\vec{r}_{po} - \vec{r}_{pj}$ from the second triangulation of element $p_o p_{j+1} p'_j p_j$, Fig. 4.7, then combining Eq. (4.5) and Eq. (4.4), one gets:

$$\nabla A_M = \frac{1}{4}\left(\sum_{t=1}^{N_t}\left(w_j\left(\vec{r}_{p_j} - \vec{r}_{p_o}\right) + w'_j\left(\vec{r}_{p'_j} - \vec{r}_{p_o}\right)\right)\right) \tag{4.6}$$

where w_j and w'_j are weights derived based on the cotangents of the angles adjacent to the edges of the element $p_o p_{j+1} p'_j p_j$ under consideration. More specifically, following the way a sub-area is expressed in Eq. (2.41), the cotangent of each angle adjacent to an edge of the element is used in the computation. As a result, for

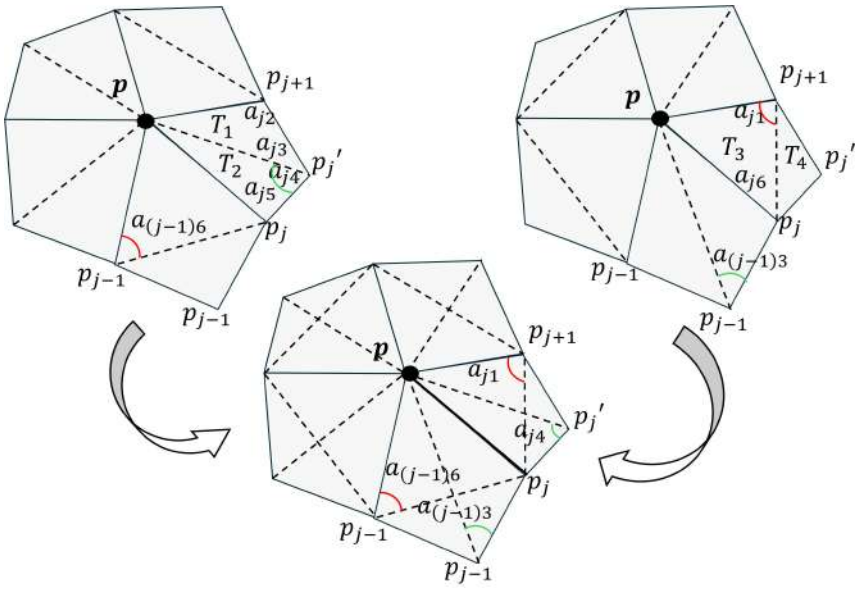


Figure 4.7: *Average triangulation of each element of point p[14].*

each area gradient corresponding to the two possible triangulations, similar to the expressions in [14], the following expressions are obtained:

$$\nabla A_j(T_1) = \frac{1}{2}\left(\cot\alpha_{j2}(\vec{r}_{pj'} - \vec{r}_p) + \cot\alpha_{j3}(\vec{r}_{p_{j+1}} - \vec{r}_p)\right), \tag{4.7}$$

$$\nabla A_j(T_2) = \frac{1}{2}\left(\cot\alpha_{j5}(\vec{r}_{p'_j} - \vec{r}_p) + \cot\alpha_{j4}(\vec{r}_{p_j} - \vec{r}_p)\right), \tag{4.8}$$

$$\nabla A_j(T_3) = \frac{1}{2}\left(\cot\alpha_{j1}(\vec{r}_{p_j} - \vec{r}_p) + \cot\alpha_{j6}(\vec{r}_{p_{j+1}} - \vec{r}_p)\right) \tag{4.9}$$

Substituting in Eqs. (4.10) the index p_{j+1} with p_j and their corresponding weights, then the weights of Eq. (4.6) are expressed as follows:

$$w_j = \cot\alpha_{j1} + \cot\alpha_{j4} + \cot\alpha_{(j-1)3} + \cot\alpha_{(j-1)6}, \quad w'_j = \cot\alpha_{j2} + \cot\alpha_{j5} \tag{4.10}$$

Substituting Eqs. (4.4), (4.6), (4.2) and (2.41) to Eq. (4.3) , the H of point p is given by:

$$||H(p)|| = \frac{1}{8}\left(\sum_{t=1}^{N_t}\left(w_j\left(\vec{r}_{p_j} - \vec{r}_{p_o}\right) + w'_j\left(\vec{r}_{p'_j} - \vec{r}_{p_o}\right)\right)\right) \tag{4.11}$$

Then, using the Eq. (2.1), it can be inferred that the *Laplace - Beltrami* is given as:

$$\nabla_s^2 \vec{r}(p) = \frac{1}{2}\left(\sum_{t=1}^{N_t}\left(\frac{1}{2}w_j\left(\vec{r}_{p_j} - \vec{r}_{p_o}\right) + \frac{1}{2}w_j'\left(\vec{r}_{p_j'} - \vec{r}_{p_o}\right)\right)\right) \tag{4.12}$$

which by using Eq. (2.36) can be expressed as:

$$\overline{\nabla_s^2 \vec{r}(p)} = \frac{1}{2}\left(\sum_{t=1}^{N_t}\left(\nabla_s^2 \vec{r}(p)_1 + \nabla_s^2 \vec{r}(p)_2 + \nabla_s^2 \vec{r}(p)_3\right)\right) \tag{4.13}$$

This means that the gradient of the position vector at the central node p of a quadrilateral star can be expressed as the average of the gradients obtained from the two possible triangulations of the surrounding elements.

From the above, it follows that the H at each node in a quadrilateral mesh can be calculated either by averaging the nodal area and position vector gradient from the two possible triangulations, or by using Eq. (4.13) together with the corresponding nodal area formula, given as:

$$A_M = \frac{1}{16}\left(\sum_{t=1}^{N_t}\left(w_j\left(\vec{r}_{p_j} - \vec{r}_{p_o}\right)^2 + w_j'\left(\vec{r}_{p_j'} - \vec{r}_{p_o}\right)^2\right)\right) \tag{4.14}$$

This computational model is evaluated on the saddle surface discussed in Section 4.2. More specifically, the maximum and mean errors of the H are computed using the two-triangulation averaging method applied to each element and their results are shown in Table 4.5. As it is shown, the two-triangulation averaging method yields the most accurate results. This observation will be further evaluated in the context of Gaussian curvature.

Triangulation methods	**Maximum error of mean curvature**	**Mean error of mean curvature**
Similar to Fig. 4.3c	0.0028%	0.0014%
Similar to Fig. 4.3d	0.0028%	0.0016%
Two-triangulation averaging	0.0028%	0.0014%

Table 4.5: *Accuracy of mean curvature at the diagonals of saddle using two-triangulation averaging in contrast to each individual triangulation.*

4.5 Gauss Curvature on Quadrilateral Meshes

The Gauss curvature at a triangulated mesh node is computed using Eq. (3.9), which shows that it depends on the sum of the interior angles of the nodal area A_M. Using the observations for the case of the mean curvature to the Gauss curvature computation in the case of quadrilateral meshes, the methodology of the two-triangulation averaging is using here as well, as it is mentioned to [14]. For the element $p_o p_{j+1} p'_j pj$ shown in Fig. 4.7, the interior angles averaging from the two different triangulations, is expressed as follows:

$$\overline{\theta}_j = \frac{1}{2}(\theta_j + \alpha_j + \beta_j) \tag{4.15}$$

where θ_j is the interior angle of each element from the first triangulation, while α_j and β_j are the interior angles that occur from the second triangulation of each element.

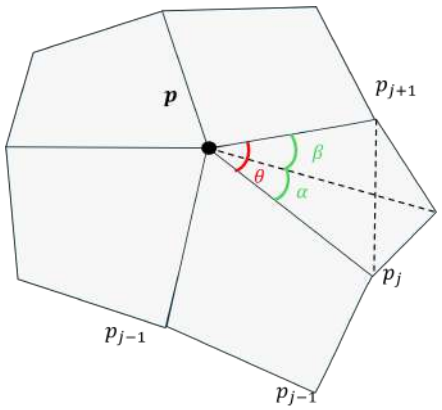


Figure 4.8: *Interior angles resulting from different triangulations of each element.*

That being said, the Gauss curvature at a node belonging to a quadrilateral star is computed using Eqs. (4.15) and (4.2), and is expressed as follows:

$$K(p) = \frac{1}{\overline{A_M}} \left(2\pi - \sum_{t=1}^{Nt} \left(\frac{\theta_j + \alpha_j + \beta_j}{2} \right) \right) \tag{4.16}$$

Eq. (4.16) is applied in the case of the surface lying upon a sphere of radius $R = 1$, which was described in Section 4.2 and its results are shown in Table 4.6. The same is done for the case of the nodal quadrilateral star in the surface given by Eq. (2.42), Table 4.7. In both cases, the error is reduced compared to the triangulation with the highest error. It is important to note that, in the case of Table 4.7, one of the triangulations yields a very high error. However, as discussed in Section 2.4.3, this

result is not fully representative, since the accuracy in this configuration is highly sensitive to the chosen radius and rotation angle. However, this case is presented in its current form here to highlight the effectiveness of the two-triangulation averaging method in absorbing the influence of high-error triangulations and producing results that are closer to those of the more accurate triangulation, as is demonstrated in Table 4.6 as well.

Triangulation methods	**Gauss curvature value**	**Error**
Method of 4.3c	1.007598423372	0.760%
Method of 4.3d	1.007060650620	0.706%
Two-triangulation averaging	1.007009389981	0.701%

Table 4.6: *Accuracy of Gauss curvature at the central point p for different methods of triangulation, with the expected value to be* $K = 1$.

Triangulation methods	**Gauss curvature value**	**Error**
Method of 4.3c	0.097661200252	56.258%
Method of 4.3d	0.058407593840	6.548%
Two-triangulation averaging	0.079626581635	27.403%

Table 4.7: *Accuracy of Gauss curvature at the nodal star of quadrilaterals for different triangulation methods, with the expected value to be* $K = \frac{1}{16}$.

Chapter 5

Proposed Curvature Computation Method: Validation and Benchmarking

5.1 Introduction

In this chapter, the proposed method for computing mean and Gauss curvature is evaluated using various types of meshes. To this end, it is applied to a range of general 3D geometries beyond the simple case of a sphere. The results obtained on both unstructured and structured meshes, where the latter typically provide higher accuracy in curvature estimation, are compared to analytical solutions by evaluating the error at each node. For unstructured meshes, which often exhibit irregular element shapes, varying sizes, and non-uniform node density, an additional analysis is conducted to investigate the causes of higher errors in curvature computation. Furthermore, the accuracy of the proposed curvature computation methods is benchmarked against those employed by widely used software tools, in order to assess their performance and reliability.

5.2 Validation of the Proposed Method on Structured Meshes

In this section, curvature measures in triangulated structured meshes in saddle and torus geometries are evaluated, compared to the analytical expressions. Beginning with a saddle surface given by:

$$z = \alpha x^2 - \beta y^2 \tag{5.1}$$

where α and β are scaling factors for the x and y dimensions, respectively, with respect to the third dimension. These parameters control the degree and direction of curvature of the surface in the $\mathbb{R}^3$ space. In this case, $\alpha = 0.5$ and $\beta = 0.5$ are selected to intensify the curvature of the surface, resulting in a wider variation in curvature values across the domain. Eq. (5.2) can also be rewritten as:

$$z = x'^2 - y'^2 \tag{5.2}$$

where $x' = \sqrt{0.5}x$ and $y' = \sqrt{0.5}y$.

The mean and Gauss curvature expressions can easily be derived as described in Chapter 1. However, as shown in Eqs. (1.10) and (1.11), the two curvature measures are computed using the first and second fundamental form coefficients over a parameterized surface. The saddle surface is explicitly described in Eq. (5.2) in the form of $z = f(x, y)$, so in order to parameterize it, the coordinates x and y can be used as parameters u and v, respectively. The first and second fundamental coefficients can then be expressed as:

$$\begin{aligned} &E = \vec{r}_x \cdot \vec{r}_x = 1 + x^2, \quad F = \vec{r}_x \cdot \vec{r}_y = -xy, \quad G = \vec{r}_y \cdot \vec{r}_y = 1 + y^2, \\ &L = \vec{r}_{xx} \cdot \vec{n} = \frac{-1}{\sqrt{1 + x^2 + y^2}}, \quad M = \vec{r}_{xy} \cdot \vec{n} = 0, \quad N = \vec{r}_{yy} \cdot \vec{n} = \frac{1}{\sqrt{1 + x^2 + y^2}}. \end{aligned} \tag{5.3}$$

Substituting this expressions to Eqs. (1.10) and (1.11), one gets the following expressions for the mean and Gauss curvature:

$$H = \frac{(1+y^2)(-1)+(1+x^2)(1)}{2(1+x^2+y^2)^{3/2}} = \frac{-1-y^2+1+x^2}{2(1+x^2+y^2)^{3/2}} = \frac{x^2-y^2}{2(1+x^2+y^2)^{3/2}}$$
$$K = \frac{(-1)(1)-0^2}{(1+x^2+y'^2)^2} = \frac{-1}{(1+x^2+y^2)^2} \tag{5.4}$$

In order to proceed to the valuation of the computation method of the curvatures, a single measure to express curvature should be defined. To do so, firstly the two principal curvatures κ_1 and κ_2 are expressed using Eqs. (1.10) and (1.11), as follows:

$$\kappa_1 = H + \sqrt{H^2 - K}, \quad \kappa_2 = H - \sqrt{H^2 - K} \tag{5.5}$$

As discussed in Chapter 1, κ_1 and κ_2 represent the maximum and minimum principal curvature at each point on the surface. In order to condense the two principal curvatures into a single scalar quantity, a measure of total curvature κ is defined. This total curvature should reflect the combined contribution of both principal curvatures, while also ensuring that $\kappa = 0$ when the surface is flat. The mean curvature, given by $\frac{\kappa_1+\kappa_2}{2}$, is not suitable in this context, since it can be zero even on non-flat surfaces. For example, when $\kappa_1 = -\kappa_2$, the surface is not flat, but the mean curvature still vanishes. Therefore, to ensure that the total curvature measure only vanishes in the flat case, the Root Mean Square (RMS) curvature is introduced, which is defined as:

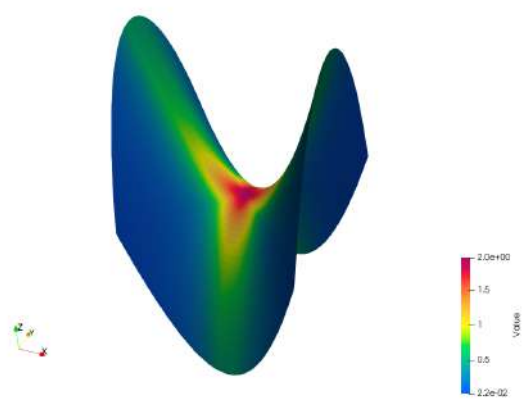
$$RMS = \sqrt{\frac{\kappa_1^2 + \kappa_2^2}{2}} \tag{5.6}$$

For simplicity, and in order to provide a scalar measure that reflects the total curvature while preserving the aforementioned property, the following expression is used:

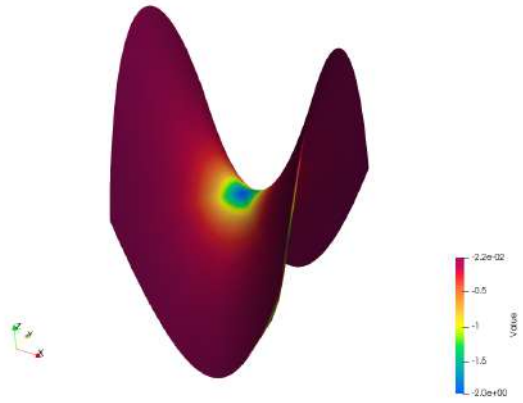
$$\kappa = \kappa_1^2 + \kappa_2^2 \tag{5.7}$$

For the case of saddle surface described in Eq. (5.2), the principal curvatures and total curvature are given in Fig. 5.1.

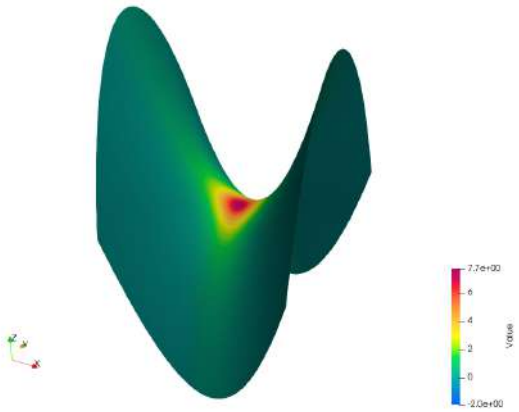
Given that in geometries such as the saddle and the torus there are regions where the expected total curvature is zero, an alternative error metric to the RMAE should be used. In such cases, the Root Mean Square Error (RMSE) provides a more robust measure of error, as it remains well-defined even when the reference curvature is zero. The RMSE is defined as:



(a) *Maximum curvature κ_1 variation over saddle surface.*



(b) *Minimum curvature κ_1 variation over saddle surface.*



(c) *Total curvature κ variation over saddle surface.*

Figure 5.1: *Principal curvatures and total curvature measures over a saddle surface.*

$$RMSE = \sqrt{\sum_{i=1}^{N} \frac{|\kappa_i^{computed} - \kappa_i^{true}|^2}{N}} \tag{5.8}$$

where N is the number of mesh nodes.

Now that a scalar measure for assessing the total curvature accuracy of a surface has been established, the proposed curvature computation method can be evaluated. To this end, the RMSE of the proposed method is compared against that of a well-known visualization software [1]. According to its documentation, this software computes Gauss and mean curvature values using the discrete differential geometry approach described in [15].

The curvature computation method proposed in this diploma thesis achieves higher accuracy than this popular software in determining the total curvature of a struc-

Computation Method	RMSE
Proposed Method	**0.0726**
Well-known visualization software	2.2944

Table 5.1: *Comparison of the total curvature RMSE on the saddle surface between the proposed computation method and a well-known visualization software.*

tured triangular mesh on a saddle surface, Table 5.1. To further validate the robustness of the method, an additional test is performed using a different geometry with a structured triangular mesh. More specifically, the torus example described in [4] is considered, with a major radius $R = 2$, minor radius $r = 1$, and a structured mesh of size 36×36, Fig. 5.2. The total curvature error is then compared with other well-known computation methods.

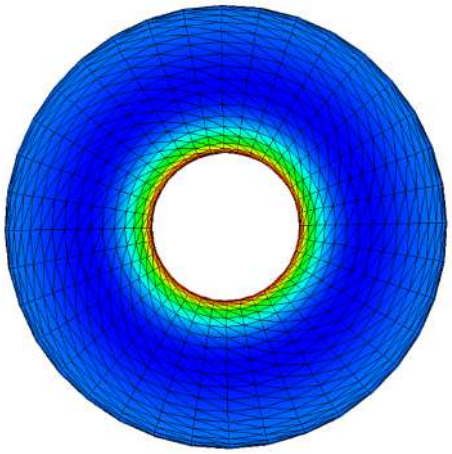


Figure 5.2: *Total curvature κ over a torus surface mesh* 36×36.

Given that the torus surface is described by:

$$\left(\sqrt{x^2 + y^2} - R\right)^2 + z^2 = r^2 \tag{5.9}$$

Then, in [2], which follows the same procedure as in the case of the saddle, the mean and Gauss curvature are derived:

$$H = \frac{-(2r + R\cos\theta)}{2r(R + r\cos\theta)}, \quad K = \frac{\cos\theta}{r(R + r\cos\theta)} \tag{5.10}$$

The proposed curvature computation method achieves the highest accuracy compared to all other known approaches, Table 5.2. It is important to note that, except for the visualization software, these methods estimate curvature directly by analyzing variations in vertex positions or face normals using techniques such as polynomial

Computation method	RMSE
Proposed Computation Method	**0.02977**
Well-known visualization software	2.2202
Libigl [18]	1.28
Meshlab [24]	2.7072
Trimesh2 [20]	1.0621
[Crane He Chen 2023] [4]	0.0372

Table 5.2: *Comparison of the total curvature RMSE on the torus surface between the proposed computation method and state-of-the-art software and established methods.*

fitting, normal variation, or quadratic patch approximation, without first computing the H and K.

5.3 Validation of the Proposed Method on Unstructured Meshes

In this section, the accuracy of the proposed computation method is examined on unstructured triangular meshes for the geometries analyzed thus far. More specifically, the H estimation is compared against that of another state-of-the-art software for CFD simulation, not previously considered. In this software, the H is computed based on the variation of the normal vector across mesh faces. This discrete approach estimates curvature by measuring the angular deviation of face normals within the mesh, which provides a scalar curvature value per face:

$$H_f = \frac{1}{2}\|\nabla \vec{n}\| \tag{5.11}$$

where $\vec{n}$ is the unit normal vector of a face and $\nabla \vec{n}$ represents its discrete gradient computed from neighboring faces.

This software approximates the curvature vector $\vec{k}_N$ on a mesh face f by summing corrected edge vectors around the face:

$$\vec{k}_N = \sum_{e \in f} \vec{L}_e \cdot \text{edgeLengthCorrection}(e) \tag{5.12}$$

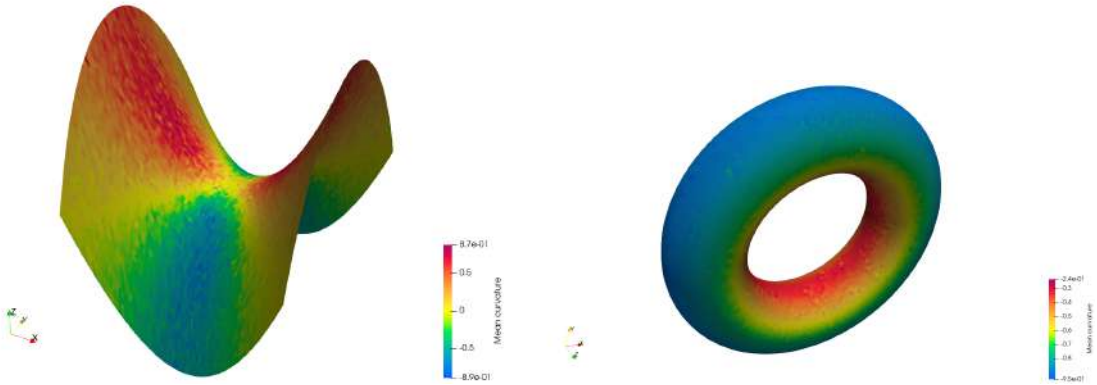
where $\vec{L}_e$ is the edge vector along edge e, edgeLengthCorrection(e) is a correction factor to improve integration accuracy and the sum is over all edges e bounding the face f.

Then, the divergence of the normal vector is approximated as:

$$\nabla_s \cdot \vec{n} \approx -\frac{\vec{k}_N \cdot \vec{S}_f}{|\vec{S}_f|^2} \tag{5.13}$$

where $\vec{S}_f$ is the face area vector, which is the normal vector to the face with magnitude equal to face area.

Since this software does not support Gauss curvature computation by default, only mean curvature will be considered in the following comparison. Nevertheless, the accuracy of both curvature measures in the proposed method strongly depends on the Voronoi area computation. Therefore, analyzing one of them is sufficient for evaluating the method.



(a) *Mean curvature on a saddle.* **(b)** *Mean curvature on a torus.*

Figure 5.3: *Mean curvature on triangular unstructured surface meshes.*

In Fig. 5.5, the H on unstructured meshes of a saddle and a torus is illustrated using the proposed method. It is evident that the contour plots on both unstructured meshes appear less smooth compared to those on structured meshes.

Computation method	**RMSE**
Proposed Computation Method	0.2783
Well-known visualization software	0.2763
Well-known CFD software	0.5564

Table 5.3: *Comparison of the H RMSE on the saddle surface between the proposed computation method and state-of-the-art software and established methods.*

In Table 5.3, it is shown that in the case of the unstructured saddle mesh, the proposed method has similar results with the visualization software [1], while the CFD simulation software [17] has lower accuracy. However, unstructured meshes,

due to their non-uniform cell shapes and sizes, typically exhibit lower accuracy than structured ones and are more prone to extreme error outliers. To examine how each method responds to significant variations in mesh quality, unstructured meshes including problematic element types, such as highly skewed, or stretched triangles are also considered in the analysis. Hence, it is not only important for a method to achieve lower error than existing software, but also for that error to be consistent and reliable for subsequent curvature-based analyses. For this reason, the Maximum Relative Absolute Error (MRAE) is used in the comparison, as it reflects more easily the validity of the curvature error at outlier points. In the case of the saddle surface, where some points have zero analytical curvature and MRE becomes undefined, the Maximum Absolute Error (MAE) is used instead.

Computation method	**MRAE**
Proposed Computation Method	**30.197%**
Well-known visualization software	66.786%
Well-known CFD software	31.670%

Table 5.4: *Comparison of the H MRAE on the torus surface between the proposed computation method and state-of-the-art software and established methods.*

Computation method	**MAE**
Proposed Computation Method	**67.161%**
Well-known visualization software	68.880%
Well-known CFD software	67.284%

Table 5.5: *Comparison of the H MAE on the saddle surface between the proposed computation method and state-of-the-art software and established methods.*

Tables 5.4 and 5.5 demonstrate that the proposed method for mean curvature computation yields to slightly better accuracy than the widely-used CFD software [17]. However, it is important to be noted that the MRAE and MAE errors of the curvature seem to be extremely high in order the results to be used an objective in the context of the design optimization. For this reason, a further investigation needs to be done for the nodes that exhibit the highest errors.

Beginning with the torus surface, the outliers, Fig. 5.4), appear to be surrounded by five or six triangles. This number of elements is generally sufficient to capture the local curvature at a node, as analyzed in Section 2.4.3. Furthermore, these nodes do not exhibit any obtuse angles. Since the interpolation of the analytical surface from the mesh did not introduce any errors, the source of the discrepancy appears to be related to a subtle geometric property of these elements that is not immediately visible. Upon closer inspection, it was found that the elements surrounding the problematic nodes each contained at least one stretched triangle, meaning that

one of the triangle's dimensions was significantly larger or smaller than the other two. This introduced a non-uniformity in edge lengths. To evaluate the effect of this distortion, a corrective method was introduced for these triangles, aimed at balancing the dominant direction in which the triangle is stretched. Because the curvature vector is closely tied to the edge vectors of neighboring faces at a node, this balancing was applied directly to the corresponding component of the curvature vector. A simple correction was chosen for this case: the component of the curvature vector in the stretched direction was scaled by the ratio of the stretched edge component to the mean of the other two edge components. When this modification was applied to the outlier that originally exhibited a 30.197% error in Table 5.4, the error at that node was significantly reduced. Consequently, applying this correction, or a similar modification, to the dominant edges of the triangles surrounding problematic nodes can substantially improve the accuracy of the method.

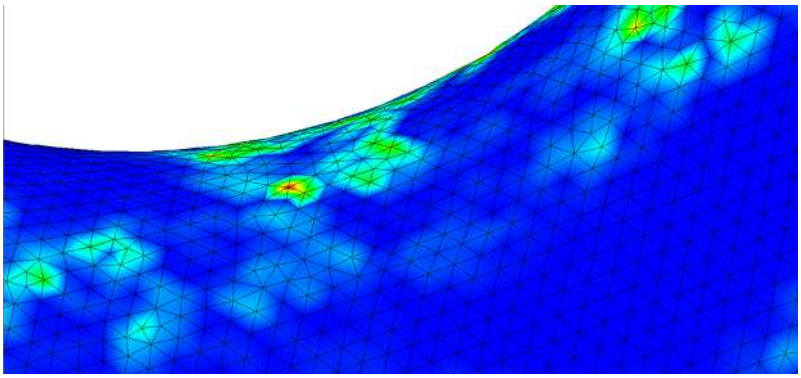


Figure 5.4: *Zoomed-in view of the nodes with the highest curvature error in the unstructured torus mesh.*

Regarding the saddle surface, where the error reported in Table 5.5 is notably high, mesh statistics revealed a more pronounced occurrence of stretched triangles. Additionally, the node with the highest error, along with several others, was surrounded by only four neighboring triangles. As discussed earlier, this limitation is difficult to be addressed to address due to physical constraints inherent in the domain. One approach suggested by [7] is to use quadric surface fitting to estimate the curvature at nodes with a low number of neighbors. However, this method has the risk of falsely estimating the curvature due to the significant deviation of the fitted surface from the actual geometry. Another possible solution is to improve the mesh quality by refining the elements to eliminate nodes with a low number of neighbors. Additionally, it is proposed here as a potential to include contributions from second-order neighbors when the number of first-order neighbors is insufficient. This could help enhance curvature estimation in sparsely connected regions without relying on fitting techniques.

5.4 Validation of the Curvature Aggregated Function

As was mentioned in 1, the final goal of total curvature computation to each node was to extract a final measure of curvature for the boundary surface of the structure's geometry in order this to be imposed as a constraint to shape or topology optimization. The reason why one measure to characterize all the surface is used, is because multi-constrained optimization, with a specific curvature constraint for each node would make the convergence more difficult. For this reason, a measure that gathers all the nodes constraints of total curvature and characterizes the total curvature of the surface should be defined. There are various ways to gather all the point-wise curvature measures into one. Among these, the p-norm aggregation function is widely employed due to its ability to approximate the maximum value of a distributed constraint field in a smooth and differentiable manner. Unlike the true maximum function, the p-norm provides a continuously differentiable approximation that enables efficient use of adjoint solvers. This smoothness is important in curvature-constrained problems to ensure stable convergence. The p-norm aggregation function is defined as:

$$f_p(x_1, x_2, \ldots, x_n) = \left(\sum_{i=1}^{n} x_i^p \right)^{\frac{1}{p}} \tag{5.14}$$

where x_i represents the i-th local constraint value (here, nodal curvature), and $p \in \mathbb{R}^+$ controls the closeness of the approximation to the maximum.

As $p \to \infty$, the function f_p converges to $\max(x_1, \ldots, x_n)$, acting like a maximum function. This property is particularly useful for curvature constraints, where it is important not only to aggregate nodal curvature values but also to emphasize regions of high curvature. In practice, moderate values such as $p = 3$ often often are used as they balance smoothness and conservativeness effectively.

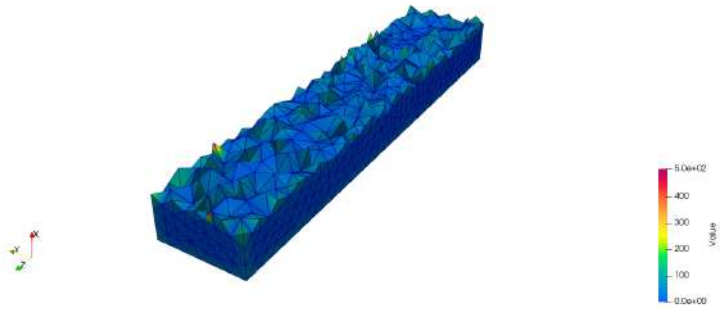
As a result the total curvature measure is given as:

$$f_3(\boldsymbol{\kappa}) = \left(\sum_{i=1}^{n} \kappa_i^3 \right)^{\frac{1}{3}} \tag{5.15}$$

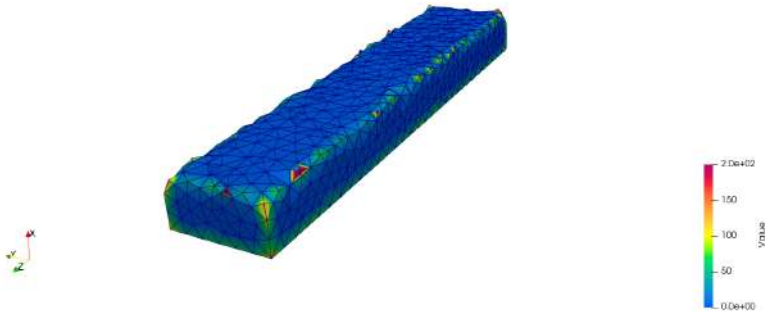
To validate the effectiveness of the proposed curvature-based objective function, it was applied to a surface roughness minimization problem. Specifically, an initially rough surface was considered, Fig. 5.5a), and then smoothed through shape optimization with the objective of reducing roughness, Fig. 5.5b. As expected, the initial

surface exhibits significantly higher total surface curvature, which in this application serves as a quantitative indicator of surface roughness. This correlation is clearly reflected in the total curvature measure.

It is also observed that the difference in total curvature between the rough and the smoothed surfaces is not as pronounced as one might expect based on visual inspection. This is primarily due to a small number of localized regions on the smoothed surface where curvature was not sufficiently minimized. Despite being few, these regions have a significantly large impact on the final aggregated curvature value. This outcome supports the effectiveness of the proposed aggregation function, which successfully emphasizes regions with high curvature, as intended.



(a) *Total curvature measure of rough surface* $\kappa_{tot} = \boldsymbol{862.559}$.



(b) *Total curvature measure of smoothed surface* $\kappa_{tot} = \boldsymbol{763.604}$.

Figure 5.5: *Total curvature measure of an arbitrary surface before and after roughness minimization.*

Chapter 6

Conclusions and Recommendations for Future work

6.1 Conclusions

The aim of this diploma thesis was to investigate the behavior of curvature metrics, specifically, mean curvature and Gauss curvature, on both structured and unstructured surface meshes. The goal was to formulate a unified constraint function to be used in gradient-based shape or topology optimization problems in CFD. However, further analysis of the key components used to compute total curvature, based on methods proposed in the literature, revealed significant accuracy issues, even for simple geometries, such as the surface of a sphere. These limitations raise a critical question: How can curvature be reliably imposed as a constraint in CFD optimization problems when its accurate computation is not guaranteed? In other words, how can curvature be accurately computed on arbitrary surface meshes and incorporated as a constraint function in optimization?

Accurately estimating surface curvature has long been a challenge, as existing methods are often inadequate for the intended applications. This research was driven by the need to ensure that curvature constraints are properly defined and accurately computed, so they do not unnecessarily restrict the optimization process. This is translated into the minimization of the maximum computational error of curvature. It is crucial to determine curvature accurately at each node on the surface mesh; therefore, the maximum error, alongside the mean error, in curvature computation has been considered.

The study investigated computational methods for triangular meshes and then gen-

eralized these approaches to any structured or unstructured mesh, with a demonstration using quadrilateral meshes. The analysis of computation methods for both mean and Gauss curvature showed that the principal cause of poor estimation in these metrics is the inaccurate calculation of the area assigned to the nodes, which corresponds to a poor definition of their VCFV.

The literature offers three different approaches to defining VCFV for triangular meshes:

- Barycentric: This method distributes the subareas to each vertex of the element using the barycenter of the triangle. Despite its simplicity, the accuracy of the barycentric method is often insufficient for curvature computations, exhibiting maximum errors of the order of 50% even for simple geometries.
- Voronoi: This method assigns subareas to each vertex based on the circumcenter of the triangle. It involves dividing each triangle along its perpendicular bisectors to define the Voronoi region associated with each vertex. Since the subdivision depends on the specific geometry of each triangle, the area distribution better adapts to the actual shape of each element. This results in a significant improvement, reducing maximum curvature computation errors to approximately 30% for the same geometries. However, a notable drawback arises for obtuse triangles: since the circumcenter lies outside the triangle in these cases, the assigned subarea does not accurately reflect the true area associated with the vertex.
- Corrected Voronoi: This method modifies the original Voronoi approach to account for obtuse triangles. The circumcenter is replaced by the midpoint of the edge opposite to the obtuse angle, effectively treating the obtuse triangle as if it were right-angled. This correction reduces the maximum curvature estimation error to approximately 20%. Nonetheless, the correction assigns constant portions of subareas to the vertices for all obtuse triangles, regardless of the magnitude of the obtuse angle, which can still lead to inaccuracies depending on the case.

The accuracy of curvature estimation methods proposed in the literature was found to be inadequate, highlighting the need to develop new and improved approaches for defining VCFV. This work builds upon the most accurate existing approach, the Corrected Voronoi Method, by introducing several enhancements aimed at addressing a key shortcoming: the method fails to account for the specific geometry of obtuse triangles when distributing area to nodes. Through detailed analysis, it was shown that the vertex opposite the the dominant angle in an obtuse triangle should receive a significantly larger share of the area than the uniform half typically assigned in right triangles. Modifications based on angle ratios or edge lengths were investigated, resulting in error reductions to below 10%.

Following these observations, a new method was introduced in this diploma thesis, called the Geometry-Adaptive Corrected (GAC) Voronoi Method. This method fo-

cuses on addressing the primary flaw of the original Corrected Voronoi approach, which assumes that obtuse triangles can be treated as right triangles. To do so, two corrective factors were introduced to each subarea attributed to a vertex: one correcting this assumption, and another addressing the fact that parts of the subareas extend outside the actual obtuse triangle. Applying these corrections significantly reduced the maximum curvature error to a satisfactory level under 1%, corresponding to an order of magnitude improvement over the original method.

To smoothly integrate these corrections for obtuse triangles with the original method for acute and right triangles, a sigmoid function was used to create a smooth transition between triangle types, resulting in the Smoothed Geometry-Adaptive Corrected (SGAC) Voronoi Method, which slightly improves accuracy over the non-smoothed version and reassures the differentiability. This proposed method was applied to both mean and Gauss curvature metrics, on different mesh types, structured and unstructured, and consistently outperformed existing methods. It was further validated against widely used methodologies from the literature and state-of-the-art software for CFD simulation or visualization and post-processing of computed flow fields, demonstrating outstanding performance improvements.

The SGAC Voronoi Method was also applied to more complex geometries, with special focus on unstructured meshes, where irregularities are more common than in structured meshes. Even here, the proposed methodology achieved better accuracy, though significant errors, up to approximately 60%, were observed at some nodes. These errors were further investigated to identify their root cause, confirming an important observation when approximating curvature on discretized geometries: although true curvature is an intrinsic geometric property of a smooth surface, its computation on discrete surfaces is highly sensitive to mesh quality and irregularities. This imposes two key considerations for accurate curvature computation:

- The surface mesh must be of high quality, with minimal skewed or stretched triangles, uniform element density, and a sufficiently large number of neighbors per node (ideally more than 5) to ensure accurate curvature estimation.
- The mesh must have sufficiently high node and element density to capture all relevant curvature features at smaller scales, providing good resolution and reliable interpolation of the actual surface geometry.

These properties can be ensured through appropriate processing of the mesh within the mesh generation software. For example, the unstructured meshes used in this diploma thesis were produced using the Gmsh software, which allows control over various statistical indices that determine mesh quality. One such index is the Signed Inverse Condition Number (SICN), which reflects the quality of elements based on the Jacobian determinant. Appropriate mesh refinement can also prevent the presence of nodes with too few neighboring elements.

As a result, curvature computation accuracy is highly influenced by the geometry of the mesh elements. In particular, unstructured meshes require special treatment to

achieve the best possible accuracy. Nonetheless, in both structured and unstructured meshes of any element type, vertices with obtuse angles are likely to be present; these can be effectively handled using the proposed method, which is general and applicable to any mesh type. Following these steps ensures the highest possible point-wise accuracy in curvature computation. Consequently a more representative and reliable final measure of curvature of the structure's mesh is obtained, which can be safely used as a constraint function for shape or topology optimization.

6.2 Recommendations for Future Work

In this diploma thesis, the method for accurately capturing the curvature of a structure's surface was demonstrated, with the ultimate goal of applying it to shape and topology optimization problems. Several important considerations should be taken into account when implementing the curvature objective within the optimization process:

- Curvature is, by definition, a point-wise measure. However, in software for CFD simulation, flow quantities are typically computed and stored at cell centers, and boundary conditions are imposed on the boundary face centers. To maintain consistency in data storage, the curvature at the boundary should first be computed at each node using the proposed methods, and then interpolated to each boundary face.

- When face-centered curvature is used as a constraint in optimization, it is advisable to normalize the face curvature values, due to their high dependence on the geometry of triangular elements, as follows::

$$\kappa_{f_n} = \frac{\kappa_f S_f}{\sum S_f} \tag{6.1}$$

  where S_f is the area of each face and $\sum S_f$ is the total area of all faces. This normalization ensures that larger faces have proportionally greater influence.

- An appropriate aggregation function should be selected to combine all nodal curvature constraints into a single scalar constraint, facilitating better convergence in the optimization.

Especially for the last point, there exists a wide variety of aggregation functions used in shape and topology optimization constraints, among these, the p-norm aggregation function is widely used due to its ability to approximate the maximum value of a distributed constraint field in a smooth and differentiable manner.

Thus, the curvature constraint can be expressed as:

$$f_3(\kappa) = \left(\sum_{i=1}^{n} \kappa_i^3 \right)^{\frac{1}{3}} \tag{6.2}$$

and constrained within the optimization as:

$$f_3(\kappa) \leq \kappa_{\max} \tag{6.3}$$

The aggregated curvature function was used to facilitate smoother convergence in CFD-based optimization. However, if curvature in specific regions is of critical importance for the problem under consideration, it is advisable to impose separate point-wise curvature constraints, or alternatively, apply aggregation functions over smaller subregions rather than over the entire boundary surface.

Finally, as previously mentioned, it is critical to ensure appropriate mesh quality in the beginning in order the curvature constraint to perform well. This can largely be assured by pre-processing the mesh in the mesh generator software. However, even if the initial mesh quality is satisfactory, mesh irregularities may develop during the optimization cycles, where the mesh of the geometry changes. In order the proposed method to work well, a few extra consideration should be taken when incorporated in optimization:

- A small number of neighboring nodes should be avoided. When increasing the number of neighbors is not feasible, a possible solution is to validate the curvature estimation using second-order neighboring nodes.
- In cases where the surface geometry changes abruptly, extremely high or undefined curvature values may arise, potentially causing issues in the computational model. A practical solution is to apply local quadric surface fitting to impose curvature bounds, which helps avoid large, non-physical curvature values. Alternatively, a simple filtering condition can be introduced to automatically exclude outlier points.

Appendix A

Mean Curvature on Structured Surface Meshes

Considering a surface in $\mathbb{R}^3$ with parameters u and v, the basic differential equation that determines the structured mesh generation is given below:

$$\nabla_s^2 \vec{r} = 2H\underline{\vec{N}} \tag{A.1}$$

The right hand side of this equation comprises of two invariant quantities in the right hand side, the mean curvature H and unit normal vector $\vec{N}$. This indicates that it can create the mesh based solely on the geometry of the surface and is not affected by the surface parameters. In order to prove the Eq. (2.1), first some basic concepts of 2D structured meshes need to be demonstrated.

Firstly, assuming a surface mesh, a curvilinear system of coordinates (CCS) is used to determine the coordinates of each point. The 2D structured mesh is parameterized using (ξ, η). Alongside the concept of the curvilinear coordinate system, there is always the concept of a transformation or mapping. These two equivalent notions indicate that there exists a one-to-one correspondence between each mesh node in the physical space x_i (i = 2 for 2D mesh), also referred to as the physical domain, and a much simpler uniform mesh composed of square cells in the case of 2D structured meshes with unit side length in the space ξ^i, also referred to as the computational or transformed domain, Fig. A.1. This transformation allows the governing differential equations to be expressed in a form independent of the surface parameterization, using geometric metrics that fully describe the transformation. This makes it possible to apply numerical schemes like finite differences or finite volumes on a uniform mesh more effectively.

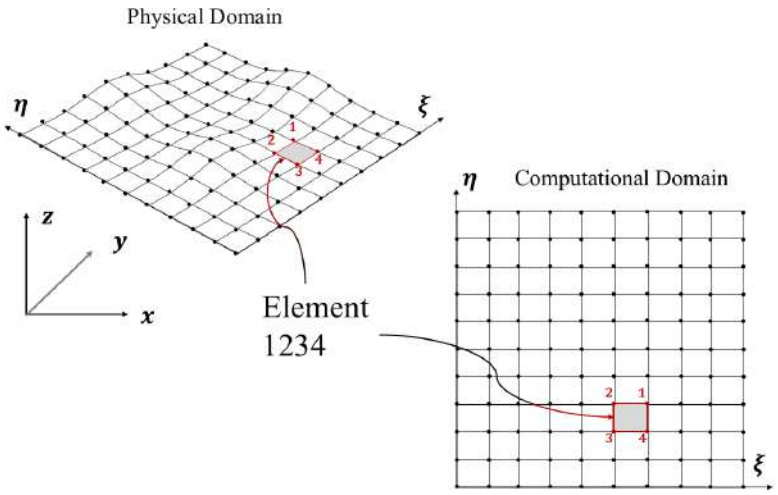


Figure A.1: *Mapping of element 1234 from physical to computational domain.*

Assume that for the computational domain of Fig. A.1 a curvilinear coordinate mesh has been generated. Through the geometrical transformation mentioned above, a node M of the mesh in the physical domain is mapped onto the node M' in the transformed domain. Through the node M two families of mesh curves pass: one curve along which ξ varies and one along which η varies. For these two mesh curves the tangent vectors at node M are defined, which are given respectively by the following:

$$\vec{g}_i = \frac{\partial \vec{r}}{\partial \xi_i}, \quad (i = 1, 2) \tag{A.2}$$

where $\vec{r}$ is the position vector of point M, $\xi_1 = \xi$ and $\xi_2 = \eta$. These are linearly independent vectors (not necessarily of unit length) and form what is called the *covariant* vector basis at M.

From the same point M emanate two curves: the curve corresponding to ξ = constant and η = constant. The normals to these two curves at M (which are not necessarily unit vectors) are given by:

$$\vec{g}^i = \nabla \xi_i, \quad (i = 1, 2) \tag{A.3}$$

where $i = 1, 2$ and (ξ^1, ξ^2) denote the curvilinear coordinates (ξ, η) respectively. These normal vectors form the so-called *contravariant* vector basis at M. Fig. A.2 symbolically shows these two vector bases at point M in the 2D domain.

J and G are the Jacobian determinants of the transformation, given by:

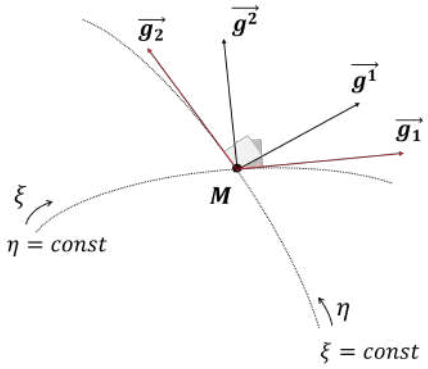


Figure A.2: *The contravariant and covariant vector bases of mesh node M in the 2D field.*

$$J = \det \begin{bmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{bmatrix} \quad \text{and} \quad G = \det \begin{bmatrix} \xi_x & \eta_x & \zeta_x \\ \xi_y & \eta_y & \zeta_y \\ \xi_z & \eta_z & \zeta_z \end{bmatrix} \tag{A.4}$$

The determinant of J has a clear physical meaning since it represents the measure of the area (in 2D) or volume (in 3D) of a cell in the mesh.

Now some useful relations of tensor metric will be provided. The elementary vector of displacement $d\vec{r}$ at any point is defined as

$$d\vec{r} = \vec{g}_i \, d\xi_i \tag{A.5}$$

and the differential of the covariant vector basis is expressed by:

$$d\vec{g}_i = \Gamma^k_{ij} \, d\xi_j \, \vec{g}_k, \tag{A.6}$$

where Γ^k_{ij} denotes the Christoffel symbols defined by:

$$\Gamma^k_{ij} = \frac{1}{2} \, g^{km} \left(\frac{\partial g_{mj}}{\partial \xi^i} + \frac{\partial g_{im}}{\partial \xi^j} - \frac{\partial g_{ij}}{\partial \xi^m} \right). \tag{A.7}$$

For the *Laplace-Beltrami* operator to be expressed over a surface mesh it is necessary to give expressions for the the gradient of a scalar field Φ and the divergence of a vector $\vec{A}$ as follows:

$$\text{grad } \Phi = \nabla \Phi = \mathbf{g}^i \frac{\partial \Phi}{\partial \xi_i}, \quad \text{div } \vec{A} = \nabla \cdot \vec{A} = \frac{1}{J} \frac{\partial (J A^i)}{\partial \xi_i} \tag{A.8}$$

and by using the operators definitions of Eq. (A.8), the Laplacian operator is derived:

$$\Delta\Phi = \mathrm{div}\big(\mathrm{grad}\ \Phi\big) = \frac{1}{J}\frac{\partial}{\partial\xi_j}\Big(J\, g^{ij}\,\frac{\partial\Phi}{\partial\xi_i}\Big). \tag{A.9}$$

Now that key metrics and operators for 2D and 3D structured meshes have been defined, the formulation can be extended to the case of surface meshes in the $\mathbb{R}^3$ space. Firstly, similar to the planar 2D case where each index takes values 1 and 2 and Eq. (A.9) applies, the following expression holds for the *Laplace–Beltrami* operator of the position vector on a surface mesh:

$$\Delta_s\vec{r} = \frac{1}{J}\frac{\partial}{\partial\xi^i}\Big(J\, g^{ij}\frac{\partial\vec{r}}{\partial\xi^j}\Big) \tag{A.10}$$

where J denotes the Jacobian of the transformation, Eq. (A.4).

In order to process the previous expression further, two auxiliary relations concerning the derivatives of the determinant J and the contravariant metric components are stated without proof:

$$\frac{\partial g^{ij}}{\partial\xi^\kappa} = -\,g^{aj}\Gamma^i_{a\kappa} - \,g^{ai}\Gamma^j_{a\kappa} \tag{A.11}$$

and

$$\frac{\partial J}{\partial\xi^i} = J\Gamma^j_{ji} \tag{A.12}$$

As derived from the Gauss–Weingarten framework, the Gauss equations take the following form:

$$\vec{r}_{,ij} = \Gamma^\kappa_{ij}\vec{r}_{,\kappa} + \Omega_{ij}\underline{\vec{N}} \tag{A.13}$$

where Ω_{ij} the second fundamental coefficients, Eq. (1.4), and Γ^κ_{ij} the Christoffel symbols of second kind as they are given by Eq. (A.7). As shown in their expression, they depend only on the first fundamental coefficients E, F and G, as well as their derivatives. In addition, by their definition, for Christoffel symbols it holds that $\Gamma^\kappa_{ij} = \Gamma^\kappa_{ji}$.

Combining the definition of the covariant tensor and the definition of the first fundamental coefficient, Eq. (1.2), it follows that:

$$E = g_{11}, \quad F = g_{12}, \quad G = g_{22} \tag{A.14}$$

Substituting the Eqs. (A.14) to the expression of mean curvature as given by Eq. (1.10), one gets:

$$H = \frac{1}{2} g^{ij} \Omega_{ij} \tag{A.15}$$

Now that the geometric definitions for surface meshes have been established, the proof of Eq. (A.1) can be developed. First, Eq. (A.10) can be expanded to obtain:

$$\Delta_s \vec{r} = \frac{1}{J} \frac{\partial J}{\partial \xi^i} g^{ij} \frac{\partial \vec{r}}{\partial \xi^j} + \frac{\partial g^{ij}}{\partial \xi^i} \frac{\partial \vec{r}}{\partial \xi^j} + g^{ij} \frac{\partial^2 \vec{r}}{\partial \xi^i \xi^j} \tag{A.16}$$

Then, substituting Eqs. (A.11), (A.12) and (A.13) to Eq. (A.16), it follows:

$$\Delta_s \vec{r} = \frac{1}{J} J \Gamma^{\lambda}_{\lambda i} g^{ij} \frac{\partial \vec{r}}{\partial \xi^j} - g^{\alpha j} \Gamma^{i}_{\alpha i} \frac{\partial \vec{r}}{\partial \xi^j} - g^{\alpha i} \Gamma^{j}_{\alpha i} \frac{\partial \vec{r}}{\partial \xi^j} + g^{ij} \Gamma^{\kappa}_{ij} \frac{\partial \vec{r}}{\partial \xi^\kappa} + g^{ij} \Omega_{ij} \underline{\vec{N}} \tag{A.17}$$

Since all indices λ, i, j, α and κ represent coordinate directions in 3D space, they take values in $1, 2, 3$. As a result, index substitution is acceptable in cases where it results in more convenient expressions. Taking advantage of this and by substituting Eq. (A.15) to Eq. (A.17), the final form of Eq. (A.1) is derived:

$$\begin{aligned}
\Delta_s \vec{r} &= [\Gamma^{\lambda}_{\lambda i} g^{ij} - g^{\lambda j} \Gamma^{i}_{\lambda i} - g^{\lambda i} \Gamma^{j}_{\lambda i} + g^{i\lambda} \Gamma^{j}_{i\lambda}] \frac{\partial \vec{r}}{\partial \xi^j} + 2\mu \underline{\vec{N}} \\
\Delta_s \vec{r} &= [\Gamma^{\lambda}_{\lambda i} g^{ij} - g^{ij} \Gamma^{\lambda}_{i\lambda} - g^{\lambda i} \Gamma^{j}_{\lambda i} + g^{i\lambda} \Gamma^{j}_{i\lambda}] \frac{\partial \vec{r}}{\partial \xi^j} + 2\mu \underline{\vec{N}} \\
\Delta_s \vec{r} &= 2\mu \underline{\vec{N}}
\end{aligned} \tag{A.18}$$

The above equation plays a key role in structured mesh theory, as it provides an elliptic formulation for generating structured surface meshes. In particular, the invariance of both the mean curvature and the unit normal vector under reparameterization gives the equations that together with others are governing the structured surface mesh generation process.

Appendix B

Cotangent Formula for Voronoi Area

In this section the formula of Eq. (2.14) used for calculating the corresponding area to each node will be proven. Firstly, in a triangle $\triangle ABC$ the circumradius is given by:

$$R = \frac{abc}{4rs} \quad \Rightarrow \quad 4Rrs = abc \tag{B.1}$$

where R the radius of the circumscribed circle, r is the radius of the inscribed circle, Fig. B.1, a, b and c the length of the triangle edges and s the semi-perimeter of the triangle which is given by: $s = \frac{a+b+c}{2}$.

To prove Eq. (B.1) the circumscribed circle of the triangle is assumed, Fig. B.2 . It is true that $\widehat{\text{ADB}} = \widehat{\text{ACB}}$ as inscribed angles in the same chord. Also, the triangles $\triangle BAD$ and $\triangle BEC$ are similar and subsequently for their edges the relation $\frac{BD}{BC} = \frac{BA}{BE}$ is valid. Using this relation, the equations below are obtained:

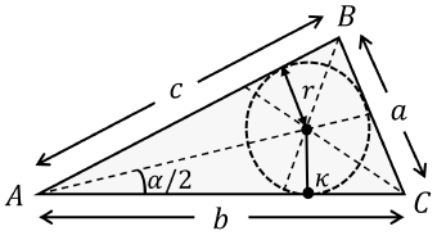


Figure B.1: *Inscribed circle of a triangle.*

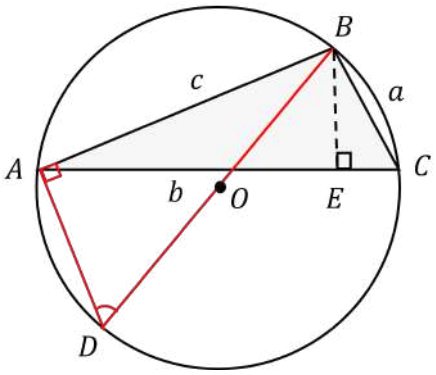


Figure B.2: *Circumscribed circle of a triangle.*

$$\left.\begin{array}{l} \dfrac{2R}{a} = \dfrac{c}{h} \\ h = \dfrac{2\text{Area}(\triangle ABC)}{b} = \dfrac{2A^T}{b} \end{array}\right\} \quad \Rightarrow \quad 2Rh = ac$$
$$\Rightarrow 2R \cdot 2A^T = abc \Rightarrow 4A^T R = abc \tag{B.2}$$

where $A^T = rs$.

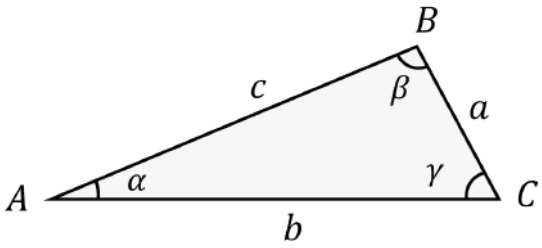


Figure B.3: *Triangle $\triangle ABC$.*

Law of cotangents in the triangle shown in Fig. C.1 is then proved. According to the law of sines it is true that:

$$\frac{a}{sin\alpha} = \frac{b}{sin\beta} = \frac{c}{sin\gamma} = 2R \tag{B.3}$$

The law is then expanded as:

$$\frac{cot\frac{\alpha}{2}}{s-a} = \frac{cot\frac{\beta}{2}}{s-b} = \frac{cot\frac{\gamma}{2}}{s-c} = \frac{1}{r} \tag{B.4}$$

where $r = \sqrt{\frac{1}{s}(s-a)(s-b)(s-c)}$ and $cot\frac{\alpha}{2} = \frac{(AK)}{2} = \frac{s-a}{r}$.

In a triangle, it is true that $\frac{\alpha+\beta+\gamma}{2} = \frac{\pi}{2}$ and since $cot(\frac{\pi}{2}) = 0$, this gives the cotangent rule:

$$cot(\frac{\alpha+\beta+\gamma}{2}) = 0 \tag{B.5}$$

which is then expanded:

$$\begin{gathered}
\frac{cot\frac{\alpha}{2} + cot\frac{\beta}{2} + cot\frac{\gamma}{2} - cot\frac{\alpha}{2}cot\frac{\beta}{2}cot\frac{\gamma}{2}}{1 - cot\frac{\alpha}{2}cot\frac{\beta}{2} - cot\frac{\beta}{2}cot\frac{\gamma}{2} - cot\frac{\gamma}{2}cot\frac{\alpha}{2}} = 0 \\
cot\frac{\alpha}{2} + cot\frac{\beta}{2} + cot\frac{\gamma}{2} = cot\frac{\alpha}{2}cot\frac{\beta}{2}cot\frac{\gamma}{2} \\
\frac{s-a}{r} + \frac{s-b}{r} + \frac{s-c}{r} = \frac{(s-a)(s-b)(s-c)}{r^3} = \frac{3s-2s}{r} = \frac{s}{r}
\end{gathered} \tag{B.6}$$

Additionally, the cotangent of any angle ϕ which is made by two vectors $\vec{a}$ and $\vec{b}$ can be expressed by dividing $\vec{a}\cdot\vec{b} = abcos\phi$ with the law of sines in Eq. (B.3) as follows:

$$\frac{\vec{a}\cdot\vec{b}}{c} = \frac{ab}{2R}\cot\phi \;\Rightarrow\; \cot\phi = \frac{2R\vec{a}\cdot\vec{b}}{abc} \tag{B.7}$$

which by using the Eq. (B.2) gives:

$$\cot\phi = \frac{2R\vec{a}\cdot\vec{b}}{4A^T R} \;\Rightarrow\; \cot\phi = \frac{\vec{a}\cdot\vec{b}}{2A^T} \tag{B.8}$$

Finally, the cotangent formula of Voronoi area of [BKOM] in $\triangle ABC$ will be proved. As it is shown in Fig. B.4, $BO = AO = CO = R$. Also, it is true that:

$$KO = \sqrt{R^2 - (KB)^2} = \sqrt{R^2 - \frac{c^2}{4}} = \sqrt{R^2(1 - sin^2\gamma)} = \sqrt{R^2cos^2\gamma} = Rcos\gamma \tag{B.9}$$

$$MO = \sqrt{R^2 - (MB)^2} = \sqrt{R^2 - \frac{a^2}{4}} = \sqrt{R^2(1 - sin^2\alpha)} = \sqrt{R^2cos^2\alpha} = Rcos\alpha \tag{B.10}$$

Also, law of sine is expressed as:

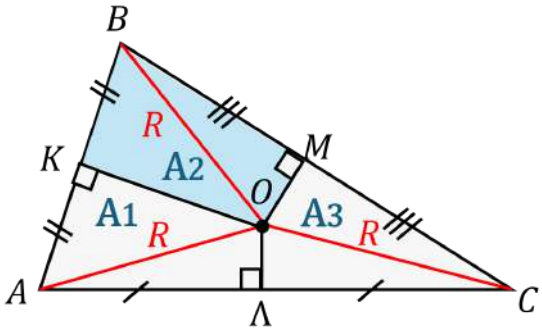


Figure B.4: *Definition of sub-volume based on the circumcenter of an acute and an obtuse triangle $\triangle ABC$, where $A = 1$, $B = 2$ and $C = 3$.*

$$2R = \frac{c}{\sin\gamma} \Rightarrow c = 2R\sin\gamma \Rightarrow c^2 = 4R^2\sin^2\gamma \tag{B.11}$$

So, the area $[\triangle BKO]$ is computed using Eq. (B.9) and Eq. (B.11):

$$[\triangle BKO] = \frac{1}{2}(KB)(KO) = \frac{c}{4}Rcos\gamma = \frac{c}{4}\frac{c}{2sin\gamma}cos\gamma = \frac{c^2}{8}cot\gamma \tag{B.12}$$

Respectively, for $\triangle BOM$:

$$[\triangle BOM] = \frac{\alpha^2}{8}cot\alpha \tag{B.13}$$

Consequently, the area [BKOM] is given:

$$[BKOM] = \frac{1}{8}(c^2cot\gamma + a^2cot\alpha) \tag{B.14}$$

which is the Voronoi area section from $\triangle ABC$ that corresponds to the node B.

Appendix C

Cotangent Formula for Triangle Angles

This section presents the formulae for the cotangent of each triangle angle in relation to its geometric quantities.

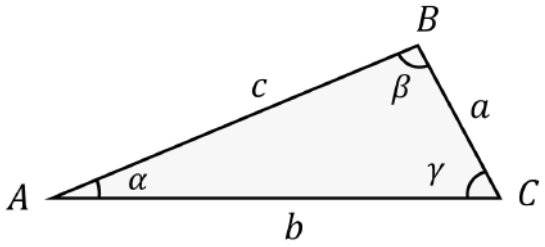


Figure C.1: *Triangle $\triangle ABC$.*

More specifically, consider $\triangle ABC$ shown in Fig. C.1. For this triangle, the Law of Cosines holds, and is given below:

$$c^2 = a^2 + b^2 - 2ab\cos\gamma, \quad a^2 = b^2 + c^2 - 2bc\cos\alpha, \quad b^2 = a^2 + c^2 - 2ac\cos\beta \tag{C.1}$$

Then, the cosine of the triangle's angles are expressed as follows:

$$\cos\gamma = \frac{c^2 - a^2 - b^2}{2ab}, \quad \cos\alpha = \frac{c^2 + b^2 - a^2}{2bc}, \quad \cos\beta = \frac{c^2 + a^2 - b^2}{2ac} \tag{C.2}$$

The area of the triangle is given by the following expressions:

$$A = \frac{1}{2}ab\sin\gamma = \frac{1}{2}cb\sin\alpha = \frac{1}{2}ac\sin\beta \tag{C.3}$$

Then, the sine of the triangle's angles are expressed as:

$$\sin\gamma = \frac{2A}{ab}, \quad \sin\alpha = \frac{2A}{cb}, \quad \sin\beta = \frac{2A}{ac} \tag{C.4}$$

Dividing cosine of each angle with its sine as given by Eqs. (C.2) and Eq. (C.4), the following expressions for the cotangent of each angle are derived:

$$\cot\gamma = \frac{a^2+b^2-c^2}{4A}, \quad \cot\alpha = \frac{c^2+b^2-a^2}{4A}, \quad \cot\beta = \frac{c^2+a^2-b^2}{4A} \tag{C.5}$$

Bibliography

[1] Ahrens, J., Geveci, B., Law, C.: Visualization Handbook, chap. ParaView: An End-User Tool for Large Data Visualization, pp. 717–731. Elsevier Inc., Burlington, MA, USA (2005), https://www.sciencedirect.com/book/9780123875822/visualization-handbook

[2] Brakke, K.: The curvature and geodesics of the torus. http://www.rdrop.com/~half/math/torus/index.xhtml (2004), accessed: 2025-06-28

[3] Cenanovic, M., Hansbo, P., Larson, M.G.: Finite element procedures for computing normals and mean curvature on triangulated surfaces and their use for mesh refinement. Computer Methods in Applied Mechanics and Engineering **370**, 113247 (2020). https://doi.org/10.1016/j.cma.2020.113247

[4] Crane, K., He, Q., Chen, J.: Estimating discrete total curvature with per triangle normal variation. Symposium on Geometry Processing (2022). https://doi.org/10.2312/sgp.20221132

[5] Dziuk, G.: Finite elements for the beltrami operator on arbitrary surfaces. S. Hildebrandt, R. Leis (ed.) Partial differential equations and calculus of variations. Lect. Notes Math **1357**, 142–155 (1998)

[6] Eschenauer, H.A., Olhoff, N.: Topology optimization of continuum structures: A review. Applied Mechanics Reviews **54**(4), 331–390 (2001)

[7] Garimella, R.V., Swartz, B.K.: Curvature estimation for unstructured triangulations of surfaces. Tech. Rep. LA-UR-03-8240, Los Alamos National Laboratory, Los Alamos, NM (October 2003)

[8] Haftka, R.T., Grandhi, R.V.: Structural shape optimization—a survey. Computer Methods in Applied Mechanics and Engineering **57**(1), 91–106 (1986)

[9] Hong, Z., Peeters, D., Turteltaub, S.: An enhanced curvature-constrained design method for manufacturable variable stiffness composite laminates. Composite Structures **287**, 115331 (2022)

[10] Høghøj, L.C., Nørhave, D.R., Alexandersen, J., Sigmund, O., Andreasen, C.S.: Topology optimization of two fluid heat exchangers. Inter-

national Journal of Heat and Mass Transfer **163**, 120543 (December 2020). https://doi.org/10.1016/j.ijheatmasstransfer.2020.120543

[11] Jameson, A.: Aerodynamic shape optimization using the adjoint method. Tech. rep., Von Karman Institute, Brussels (February 6 2003), lecture, Department of Aeronautics & Astronautics, Stanford University

[12] Knabner, P., Angermann, L.: Numerical Methods for Elliptic and Parabolic Partial Differential Equations. Springer (2012)

[13] Lipschutz, M.M.: Schaum's Outline of Differential Geometry. McGraw-Hill (1969)

[14] Liu, D., Xu, G.: Angle deficit approximation of gaussian curvature and its convergence over quadrilateral meshes. Journal of Computational and Applied Mathematics **206**(2), 1032–1046 (2007)

[15] Meyer, M., Desbrun, M., Schröder, P., Barr, A.H.: Discrete differential-geometry operators for triangulated 2-manifolds. Visualization and Mathematics III pp. 35–57 (2003)

[16] Okazaki, Y., Sakuma, I., Mori, K.: Shape optimization of stress concentration-free lattice for self-expandable nitinol stent-grafts. Medical Engineering & Physics **34**(2), 203–210 (2012). https://doi.org/10.1016/j.medengphy.2011.06.013

[17] OpenCFD Ltd.: OpenFOAM: The Open Source CFD Toolbox – User Guide (Dec 2023), `https://develop.openfoam.com`, licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License

[18] Panozzo, D., Puppo, E., Rocca, L.: Efficient multi-scale curvature and crease estimation. Proceedings of Computer Graphics, Computer Vision and Mathematics pp. 1–6 (2010)

[19] Polthier, K.: Polyhedral Surfaces of Constant Mean Curvature. Ph.D. thesis, Technische Universität Berlin (2002), habilitation thesis

[20] Rusinkiewicz, S.: Estimating curvatures and their derivatives on triangle meshes. Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT) pp. 486–493 (2004)

[21] Schmitt, O., Steinmann, P.: On curvature approximation in 2d and 3d parameter–free shape optimization. Computational Mechanics **59**(6), 933–946 (2016). https://doi.org/10.1007/s00466-016-1319-z

[22] Stankiewicz, G., Dev, C., Steinmann, P.: Geometrically nonlinear design of compliant mechanisms: Topology and shape optimization with stress and curvature constraints. Computer Methods in Applied Mechanics and Engineering **397**, 115229 (2022). https://doi.org/10.1016/j.cma.2022.115229

[23] Tajima, M., Yamada, T.: Topology optimization with geometric constraints for additive manufacturing based on coupled fictitious physical model. Computer Methods in Applied Mechanics and Engineering **398**, 115270 (2023)

[24] Taubin, G.: Estimating the tensor of curvature of a surface from a polyhedral approximation. Proceedings of the IEEE International Conference on Computer Vision (ICCV) pp. 902–907 (1995)

[25] Wu, Z.: On the optimization problem of fillets and holes in plates with curvature constraints. Structural and Multidisciplinary Optimization **35**(5), 499–506 (2008). https://doi.org/10.1007/s00158-008-0350-0

[26] Xiong, Y., Li, G., Han, G.: Mean laplace–beltrami operator for quadrilateral meshes. Journal of Computer-Aided Design & Computer Graphics **23**(5), 829–836 (2011)

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Μηχανολόγων Μηχανικών
Τομέας Ρευστών
Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής
& Βελτιστοποίησης

Υπολογισμός Καμπυλότητας σε Επιφανειακά Πλέγματα για τη Βελτιστοποίηση μέσω Συζυγούς Μεθόδου με Περιορισμούς

Διπλωματική Εργασία

Άννα Χοτζάλλη

Επιβλέπων: Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

Αθήνα, 2025

Εισαγωγή

Η βελτιστοποίηση στο μηχανολογικό σχεδιασμό είναι ευρέως εφαρμοσμένη σε διάφορους τομείς της μηχανολογίας, όπως στην αεροναυπηγική και στην αυτοκινητοβιομηχανία, οι οποίοι συνήθως απαιτούν την επίλυση υπολογιστικά απαιτητικών εξισώσεων ρευστοδυναμικής (CFD). Η βελτιστοποίηση κατασκευών έχει εξελιχθεί τα τελευταία χρόνια σε βελτιστοποίηση μορφής και τοπολογίας, καθότι προσφέρουν τη δυνατότητα μορφοποίησης της γεωμετρίας των κατασκευών ώστε να ανταποκρίνονται στις μηχανολογικές απαιτήσεις. Ένας από τους διαφόρους περιορισμούς που επιβάλλονται κατά τη βελτιστοποίηση κατασκευών είναι ο περιορισμός της καμπυλότητας, ο οποίος έχει εξέχουσα σημασία καθότι συμβάλλει στην εξασφάλιση της αντοχής, της κατασκευασιμότητας και άλλων ιδιοτήτων σχετικών με την καμπυλότητα των επιφανειών στις βελτιωμένες κατασκευές. Ιδίως η εξασφάλιση της αντοχής και της κατασκευασιμότητας μέσω της καμπυλότητας έχουν ιδιαίτερη σημασία σε πολλές μηχανολογικές εφαρμογές, όπως στη βελτιστοποίηση τεμαχίων που παράγονται μέσω προσθετικών ή κοπτικών κατεργασιών, καθότι μία βελτιωμένη λύση που δεν κατασκευάζεται ή δεν πληρεί βασικές μηχανολογικές προϋποθέσεις είναι είσονος σημασίας. Δεδομένης της αναγκαιότητας επιβολής περιορισμών καμπυλότητας σε CFD προβλήματα βελτιστοποίησης μορφής και τοπολογίας, κρίνεται απαραίτητη η δυνατότητα ακριβούς προσδιορισμού της καμπυλότητας των εξωτερικών επιφανειών των εκάστοτε γεωμετριών, ώστε να μπορούν να επιβληθούν ως περιορισμοί, πράγμα το οποίο πραγματεύεται η παρούσα διπλωματική.

Η καμπυλότητα αποτελεί μία μετρική που τόσο σε δισδιάστατες καμπύλες όσο και σε επιφάνειες χαρακτηρίζει κάθε σημείο ξεχωριστά και αποκαλύπτει τον τρόπο με τον οποίο η γειτονική περιοχή της καμπύλης ή της επιφάνειας αναπτύσσεται κατά τη μία ή περισσότερες εφαπτομενικές κατευθύνσεις. Στην περίπτωση των επιφανειών, αποτελεί αναλλοίωτη μετρική, που σημαίνει ότι παραμένει σταθερή μεταξύ των διαφόρων μετασχηματισμών του πλέγματος της επιφανείας, καθότι ορίζεται μέσω άλλων αναλλοίωτων μεγεθών του επιφανειακού πλέγματος. Συγκεκριμένα, ένα επιφανειακό πλέγμα χαρακτηρίζεται από δύο αναλλοίωτες ποσότητες, την πρώτη I και δεύτερη II θεμελιώδη μορφή, των οποίων η φυσική σημασία συνδέεται άμεσα με τοπικά γεωμετρικά χαρακτηριστικά της επιφάνειας, καθώς μέσω αυτών προσδιορίζεται η κάθετη καμπυλότητα κάθε καμπύλης που διέρχεται από το συγκεκριμένο σημείο της επιφάνειας. Η κάθετη καμπυλότητα μίας καμπύλης ορίζεται ως το μέτρο του διανύσματος της καμπυλότητας που είναι προβεβλημένο στο κάθετο διάνυσμα. Εφόσον στην επιφάνεια υπάρχουν πάνω από μία εφαπτόμενες κατευθύνσεις για κάθε σημείο, η ολική καμπυλότητα σε ένα σημείο λαμβάνει υπόψη μόνο δύο εξ αυτών των κατευθύνσεων, αυτών που αντιστοιχούν στις καμπύλες με τη μεγαλύτερη και τη μικρότερη καμπυλότητα στο συγκεκριμένο σημείο, κ_1 και κ_2 αντίστοιχα. Οι καμπυλότητες αυτές είναι οι κύριες καμπυλότητες και είναι επίσης αναλλοίωτες. Δεδομένου ότι δεν υπάρχει άμεσος τύπος υπολογισμού των κύριων καμπυλοτήτων για οποιοδήποτε τύπο πλέγματος, δύο νέες μετρικές της καμπυλότητας εισάγονται, η μέση καμπυλότητα H και καμπυλότητα Gauss K, των οποίων ο τρόπος υπολογισμού είναι εφαρμόσιμος τόσο στα δομημένα όσο και στα μη-δομημένα πλέγματα. Οι μέση και Gauss καμπυλότητες σχετίζονται με τις κύριες καμπυλότητες

ως εξής:

$$H = \frac{\kappa_1 + \kappa_2}{2}, \quad K = \kappa_1 \kappa_2 \tag{1}$$

Στη παρούσα εργασία, εξετάζονται οι μέθοδοι υπολογισμού της μέσης και Gauss καμπυλότητας σε οποιοδήποτε τύπο πλέγματος. Η μέθοδος για τη μέση καμπυλότητα υιοθετείται από την αντίστοιχη στην ανάλυση των πεπερασμένων στοιχείων και προσαρμόζεται κατάλληλα στην περίπτωση της ανάλυσης μέσω πεπερασμένων όγκων, η οποία είναι η κατά κόρων χρησιμοποιούμενη στην περίπτωση των CFD προβλημάτων βελτιστοποίησης μορφής και τοπολογίας. Η μέθοδος για τη καμπυλότητα Gauss είναι γενική για κάθε διαφορική γεωμετρία. Οι μέθοδοι εξετάζονται ως προς την ακρίβεια τους να αποτυπώνουν την πραγματική καμπυλότητα επιφανειών. Η διερεύνηση ανέδειξε την εμφάνιση σημαντικών σφαλμάτων στον υπολογισμό της καμπυλότητας και δημιούργησε την ανάγκη δημιουργίας μίας νέας διόρθωσης των μεθόδων υπολογισμού που θα προσφέρει μεγαλύτερη ακρίβεια από τις υπάρχουσες. Στην παρούσα εργασία αποτυπώνεται η μαθηματική θεμελίωση, ο προγραμματισμός και η εφαρμογή της νέας μεθόδου σε διαφόρων ειδών επιφανειακά πλέγματα. Η μέθοδος συγκρίνεται ως προς την ακρίβεια της με ευρέως χρησιμοποιούμενα λογισμικά σε περιπτώσεις δομημένων και μη-δομημένων επιφανειακών πλεγμάτων. Επιπλέον, αποτυπώνεται η μορφή της τελικής συνάρτησης της καμπυλότητας θα χρησιμοποιηθεί ως περιορισμός στη βελτιστοποίηση μορφής ή τοπολογίας.

Υπολογισμός της μέσης καμπυλότητας σε επιφανειακά πλέγματα

Η μέση καμπυλότητα H σε ένα σημείο μιας επιφάνεια δίνεται ως:

$$\Delta_s \vec{r} = \nabla_s^2 \vec{r} = 2H\underline{\vec{N}} \tag{2}$$

όπου $\underline{\vec{N}}$ το αδιάστατο κάθετο διάνυσμα στο συγκεκριμένο σημείο. Ο *Laplace - Beltrami* τελεστής Δ_s είναι η γενίκευση του *Laplacian* τελεστή για συναρτήσεις ορισμένες πάνω σε επιφάνεια.

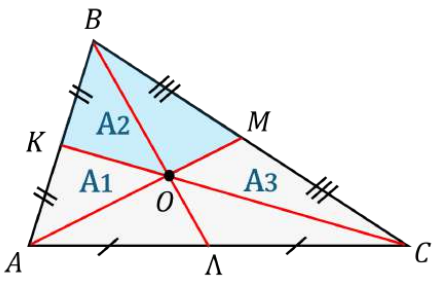
Συνεπώς, ο προσδιορισμός της μέσης καμπυλότητας σε κάθε σημείο της επιφάνειας διαμορφώνεται στον υπολογισμό του τελεστή *Laplace - Beltrami* του διανύσματος θέσης καθώς και του κάθετου διανύσματος του σημείου. Ακολουθώντας για την FVM έκφραση του πλέγματος την αντίστοιχη μεθοδολογία με αυτή στην περίπτωση της *FEM*, αποδεικνύεται ότι ο τελεστής *Laplace - Beltrami* μίας επιφανείας μπορεί να προσεγγιστεί μέσω του διαφορικού τελεστή στη λογική των πεπερασμένων όγκων.

Για τον υπολογισμό τόσο του διαφορικού τελεστή όσο και του κάθετου διανύσματος σε κάθε σημείο απαιτείται πρώτα η μετατροπή της έκφρασης των πεπερασμένων όγκων από κεντροκυψελική σε κεντροκομβική. Η μετατροπή γίνεται μέσω κατάλληλου ορισμού του πεπερασμένου όγκου ανά κόμβο λαμβάνοντας συνεισφορά από στοιχεία που τον

απαρτίζουν. Υπάρχουν τρεις διαφορετικοί τρόποι ορισμού των $VCFV$, που βασίζονται σε τρεις διαφορετικούς τρόπους απόδοσης της συνεισφοράς των γειτονικών στοιχείων στον εξεταζόμενο κόμβο:

- Βαρυκεντρικός: Βασίζεται στο διαμοιρασμό του εμβαδού του εκάστοτε στοιχείου σε κάθε κορυφή με βάση τα εμβαδά που σχηματίζουν οι διάμεσοι, διερχόμενοι από το βαρύκεντρο, με τα μέσα των πλευρών του τριγώνου, Σχήμα 1.
- *Voronoi* ή περικεντρικός: Βασίζεται στο διαμοιρασμό του εμβαδού του εκάστοτε στοιχείου σε κάθε κορυφή με βάση τα εμβαδά που σχηματίζουν οι μεσοκάθετοι, διερχόμενες από το περίκεντρο, με τα μέσα των πλευρών του τριγώνου, Σχήμα 2α. Στην περίπτωση του αμβλυγωνίου, το περίκεντρο βγαίνει εκτός τριγώνου, με αποτέλεσμα το άθροισμα των υποεμβαδών να είναι μεγαλύτερο από το εμβαδόν του τριγώνου, Σχήμα 2β.
- Διορθωμένος *Voronoi*: Παρόμοιος με το *Voronoi*, μόνο που στην περίπτωση του αμβλυγωνίου, με σκοπό το άθροισμα των υποεμβαδών να δίνει το εμβαδών του τριγώνου, εφαρμόζεται μία διόρθωση στην οποία το αμβλυγώνιο προσεγγίζεται ως ορθογώνιο, οπότε το περίκεντρο τοποθετείται στο κέντρο της πλευράς απέναντι από την αμβλεία γωνία, Σχήμα 3.

Βάσει του ορισμού του εμβαδού κάθε $VCFV$, το κάθετο διάνυσμα που αντιστοιχεί σε κάθε υποεμβαδό προκύπτει ως ο λόγος του υποεμβαδού προς το συνολικό εμβαδόν, πολλαπλασιασμένος με το ολικό κάθετο διάνυσμα.

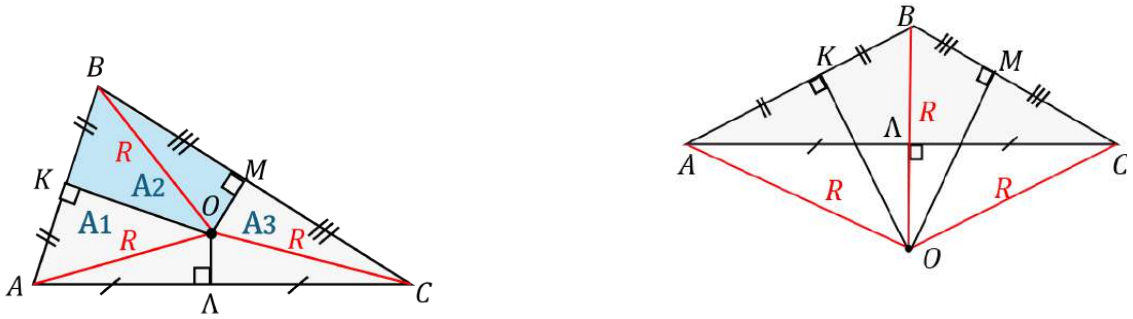


Σχήμα 1: *Ορισμός του υποεμβαδού που αντιστοιχεί σε κάθε κορυφή με βάση το βαρύκεντρο του τριγώνου* $\triangle ABC$*, όπου* $A = 1$*,* $B = 2$ *και* $C = 3$*.*

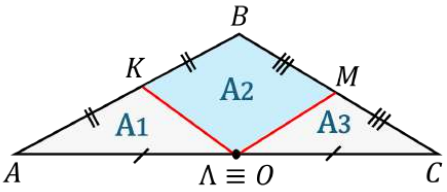
Για τον υπολογισμό του διαφορικού του διανύσματος θέσης δύο προσεγγίσεις εξετάζονται, μέσω της χρήσης των κάθετων διανυσμάτων των πλευρών του τριγώνου και μέσω της χρήσης της έκφρασης της συνεφαπτομένης, που δίνεται ως εξής:

$$\nabla_s^2 \vec{r}\Big|_o = \frac{1}{2} \sum_{i=1}^{C_g} (\vec{r}_{p_i} - \vec{r}_{p_o}) \left[\cot k_i + \cot l_i\right] \tag{3}$$

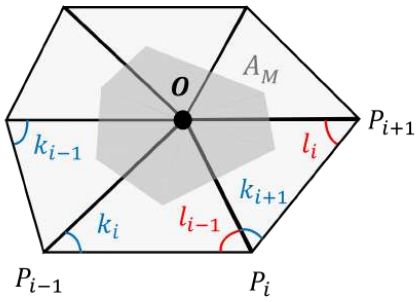
όπου C_g είναι ο αριθμός των ακμών που διέρχονται από τον κόμβο O και $\cot k_i$ και $\cot l_i$ είναι οι εφαπτόμενες των γωνιών που αντιστοιχούν στην ακμή i στα δύο τρίγωνα που τη μοιράζονται, Σχήμα 4.



Σχήμα 2: *Ορισμός του υποεμβαδού που αντιστοιχεί σε κάθε κορυφή με βάση το περίκεντρο ενός (α) οξυγώνιου και (β) ενός αμβλυγώνιου $\triangle ABC$, όπου $A = 1$, $B = 2$ και $C = 3$.*



Σχήμα 3: *Ορισμός του υποεμβαδού που αντιστοιχεί σε κάθε κορυφή με βάση το διορθωμένο περίκεντρο ενός αμβλυγωνίου $\triangle ABC$, όπου $A = 1$, $B = 2$ και $C = 3$.*

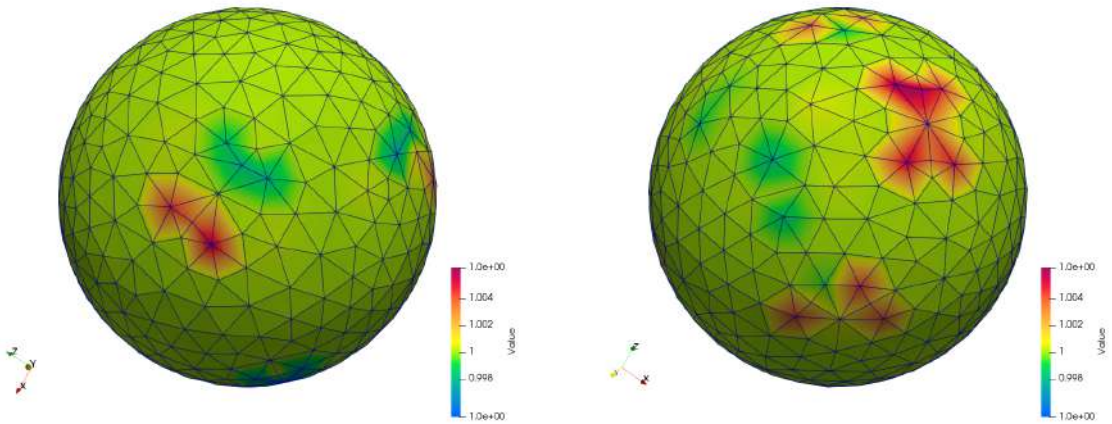


Σχήμα 4: *Εμβαδόν Voronoi για το κόμβο O ενός επιφανειακού πλέγματος.*

Οι δύο προσεγγίσεις δίνουν πανομοιότυπα αποτελέσματα, με τη δεύτερη να είναι υπολογιστικά πιο αποδοτική, για αυτό και επιλέγεται μεταξύ των δύο για τη συνέχεια.

Μεταξύ των τριών διαφορετικών ορισμών των $VCFV$, η διορθωμένη Voronoi είχε την μεγαλύτερη ακρίβεια στον υπολογισμό της μέσης καμπυλότητας στις διάφορες εφαρμογές, όπως στη σφαίρα. Το μέγιστο σφάλμα σε αυτήν τη περίπτωση ήταν της τάξης του 20%, το οποίο κρίθηκε μη ικανοποιητικό για τη χρήση της υπολογισμένης καμπυλότητας ως περιορισμό βελτιστοποίησης. Για το λόγο αυτό, κρίθηκε απαραίτητη η εισαγω-

γή μίας νέα μεθόδου, η οποία αποκαλείται Εξομαλυμένη Γεωμετρικά-Προσαρμοσμένη (SGAC) Μέθοδος Voronoi. Η μέθοδος βασίζεται στην υπάρχουσα Διορθωμένη Μέθοδο Voronoi, εφαρμόζοντας επιπλέον διορθώσεις που λαμβάνουν υπόψη τη γεωμετρία του αμβλυγωνίου για τον διαμοιρασμό των υποεμβαδών, σε αντίθεση με τη Διορθωμένη Μέθοδο Voronoi που αποδίδει σταθερό ποσοστό εμβαδού σε κάθε κορυφή σε κάθε αμβλυγώνιο. Παράλληλα, η διόρθωση αντισταθμίζει τα μέρη των υποεμβαδών που βγαίνουν εκτός του αμβλυγωνίου στην αρχική μέθοδο Voronoi. Επιπλέον, η μέθοδος εξομαλύνει τη μετάβαση από τον κανονικό ορισμό Voronoi για τα οξυγώνια και τα ορθογώνια με τον νέο διορθωμένο ορισμό για το αμβλυγώνιο. Η νέα μέθοδος δοκιμάστηκε σε δύο σφαιρικά πλέγματα, ένα αραιό (Πίνακας 1) και ένα πυκνό (Πίνακας 2), με το σχετικό απόλυτο σφάλμα (RMAE) να μειώνεται κατά μία τάξη μεγέθους σε σχέση με την υπάρχουσα Διορθωμένη Μέθοδο Voronoi. Παρόμοια συμπεράσματα αντλήθηκαν από τη σύγκριση των μεθόδων στο σφαιρικό πλέγμα μέσω του σφάλματος της μέγιστης και της ελάχιστης τιμής της μέσης καμπυλότητας, Πίνακας 3. Στο Σχήμα 5 φαίνεται η κατανομή της μέσης καμπυλότητας με τη χρήση της νέας μεθόδου, όπου παρατηρείται μικρή απόκλιση στα σημεία που περιτριγυρίζονται από τουλάχιστον ένα αμβλυγώνιο.



(α′) *Πρώτη προοπτική θέαση κόμβων με απόκλιση μέσης καμπυλότητας.*

(β′) *Δεύτερη προοπτική θέαση κόμβων με απόκλιση μέσης καμπυλότητας.*

Σχήμα 5: *Προοπτικές θέασης κόμβων με απόκλιση μέσης καμπυλότητας από την αναμενόμενη τιμή με τη χρήση της SGAC Μεθόδου Voronoi.*

VCFV Definition for Obtuse Triangles	**Relative Error**
Corrected Voronoi	0.223%
GAC Voronoi	**0.025%**

Πίνακας 1: *RMAE της GAC Μεθόδου Voronoi σε σύγκριση με την υπάρχουσα Διορθωμένη Μέθοδο Voronoi σε σφαιρικό επιφανειακό πλέγμα.*

VCFV Definition for Obtuse Triangles	Relative Error
Corrected Voronoi	0.059%
GAC Voronoi	**0.0066%**

Πίνακας 2: *RMAE της GAC Μεθόδου Voronoi σε σύγκριση με την υπάρχουσα Διορθωμένη Μέθοδο Voronoi σε πυκνό σφαιρικό επιφανειακό πλέγμα.*

VCFV Definition for obtuse triangles	Min Value of Mean Curvature	Error	Max Value of Mean Curvature	Error
Corrected Voronoi	0.941683572258	5.832%	1.222835972957	22.286%
SGAC Voronoi	0.995123547932	**0.488%**	1.006219135819	**0.622%**

Πίνακας 3: *Σύγκριση της SGAC Μεθόδου Voronoi με την υπάρχουσα Διορθωμένη Μέθοδο Voronoi του VCFV σε σφαιρικό επιφανειακό πλέγμα.*

Υπολογισμός της Gauss καμπυλότητας σε επιφανειακά πλέγματα

Ο υπολογισμός της Gauss καμπυλότητας σε τριγωνικά πλέγματα βασίζεται στο Gauss-Bonnet θεώρημα. Όταν εκφράζεται σε κάθε κόμβο p, η καμπυλότητα Gauss K δίνεται ως:

$$K(p) = \frac{1}{A_M}\left(2\pi - \sum_{j=1}^{N_t}\theta_j\right) \tag{4}$$

όπου A_M το εμβαδόν του VCFV όπως δόθηκε και στη μέση καμπυλότητα, θ_j η γωνία κάθε τριγώνου που εφάπτεται στον κόμβο p και έχει ως κορυφή των ίδιο κόμβο και N_t ο αριθμός των γειτονικών τριγώνων του p.

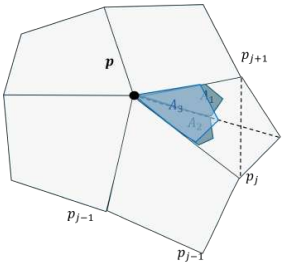
Η ακρίβεια της Gauss καμπυλότητας, όμοια με τη μέση, εξαρτάται σε μεγάλο βαθμό από τον ορισμό VCFV. Η νέα SGAC Μέθοδος Voronoi εφαρμόστηκε και σε αυτή την περίπτωση και αποδείχθηκε ως η πιο ακριβής από όσες δοκιμάστηκαν, Πίνακας 4.

VCFV Definition for obtuse triangles	Relative error
Barycentric	7.237%
Voronoi	0.833%
Corrected Voronoi	0.774%
GAC Voronoi	0.613%
SGAC Voronoi	0.616%

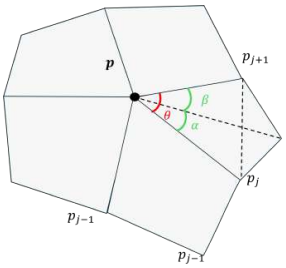
Πίνακας 4: *Σύγκριση RMAE για διαφορετικούς ορισμούς VCFV της καμπυλότητας Gaussσε επιφανειακό πλέγμα σφαίρας.*

Υπολογισμός των μετρικών καμπυλότητας σε τετραπλευρικά πλέγματα

Ο υπολογισμός των μετρικών της μέσης και Gauss καμπυλότητας στα τετραπλευρικά πλέγματα βασίζεται στην τριγωνοποίηση αυτών και την εφαρμογή των μεθόδων που αναλύθηκαν για τα τριγωνικά πλέγματα. Ωστόσο, διαφορετικοί τρόποι τριγωνοποίησης οδηγούν σε διαφορετικά αποτελέσματα καμπυλότητας. Προκειμένου να οριστεί ένας γενικός τρόπος υπολογισμού των καμπυλοτήτων που να λαμβάνει υπόψη, για κάθε κόμβο, τους δύο διαφορετικούς τρόπους τριγωνοποίησης των στοιχείων στα οποία ανήκει, ο μέσος όρος αυτών λαμβάνεται υπόψη για τον ορισμό του VCFV από τα τετραπλευρικά στοιχεία. Χρησιμοποιούνται οι ίδιοι τύποι με τα τριγωνικά στοιχεία, απλά στη θέση των βασικών όρων χρησιμοποιούνται οι μέσοι όροι από τους δύο τρόπους τριγωνοποίησης κάθε στοιχείου. Για τη μέση καμπυλότητα κάθε κόμβου, η μέση τιμή των υποεμβαδών από τις δύο τριγωνοποιήσεις κάθε γειτονικού στοιχείου χρησιμοποιείται για το εμβαδόν του VCFV, Σχήμα 6, καθώς και η έκφραση της συνεφαπτομένης με βάρη από τις δύο τριγωνοποιήσεις χρησιμοποιείται για το διαφορικό του διανύσματος θέσης, Σχήμα 8. Για τη καμπυλότητα Gauss κάθε κόμβου, χρησιμοποιείται η μέση τιμή των γωνιών που προκύπτουν από τις δύο τριγωνοποιήσεις σε κάθε γειτονικό στοιχείο, Σχήμα 7.



Σχήμα 6: *Υποεμβαδά Voronoi που προκύπτουν από δύο διαφορετικές τριγωνοποιήσεις κάθε στοιχείου.*

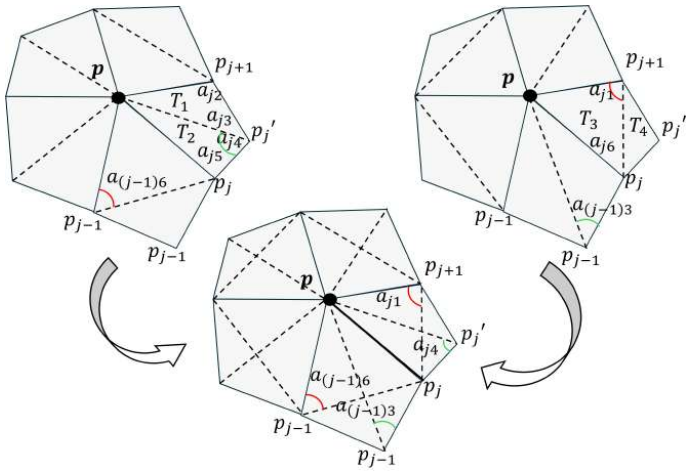


Σχήμα 7: *Εσωτερικές γωνίες που προκύπτουν από δύο διαφορετικές τριγωνοποιήσεις κάθε στοιχείου.*

Αυτή η μέθοδος εξασφαλίζει καλύτερη ακρίβεια υπολογισμού των καμπυλοτήτων σε κάθε τετραπλευρικό πλέγμα, με το σφάλμα ανά κόμβο να τείνει πιο κοντά προς το ελάχιστο εκ των δύο από τις δύο διαφορετικές τριγωνοποιήσεις κάθε γειτονικού στοιχείου του κόμβου.

Προτεινόμενη Μέθοδος Υπολογισμού Καμπυλότητας: Επικύρωση και Συγκριτική Αξιολόγηση

Μετά τον υπολογισμό της μέσης και Gauss καμπυλότητας σε κάθε σημείο, οι κύριες καμπυλότητες μπορούν να προσδιοριστούν. Η ολική καμπυλότητα πρέπει να λαμβάνει υπόψη τη μέγιστη και την ελάχιστη καμπυλότητα, ενώ διατηρεί την ιδιότητα μηδενισμού



Σχήμα 8: *Μέση τριγωνοποίηση κάθε στοιχείου του σημείου p [14].*

αν και μόνο αν και οι δύο κύριες καμπυλότητες μηδενίζονται, δηλαδή αν έχουμε επίπεδο. Ως εκ τούτου, η ολική καμπυλότητα ορίζεται ως $\kappa = \kappa_1^2 + \kappa_2^2$.

Η ακρίβεια απόδοσης της με βάση την προτεινόμενη μέθοδο συγκρίθηκε με αυτήν από ευρέως χρησιμοποιούμενα λογισμικά CFD [17], καθώς και με ξεχωριστά λογισμικά για την απεικόνιση και τη μετεπεξεργασία των υπολογισθέντων πεδίων ροής [1] και αναδείχθηκε ως η καλύτερη, Πίνακας 5.

Computation method	**RMSE**
Proposed Computation Method	**0.02977**
Well-known visualization software	2.2202
Libigl [18]	1.28
Meshlab [24]	2.7072
Trimesh2 [20]	1.0621
[Crane He Chen 2023] [4]	0.0372

Πίνακας 5: *Σύγκριση του RMSE της ολικής καμπυλότητας στην επιφάνεια τορού μεταξύ της προτεινόμενης μεθόδου υπολογισμού και προηγμένων λογισμικών καθώς και καθιερωμένων μεθόδων.*

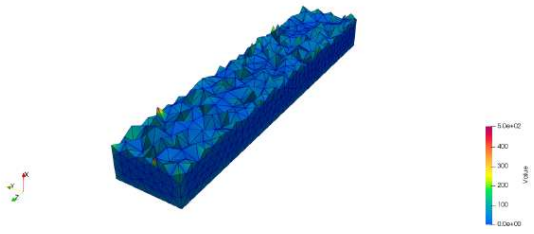
Η σύγκριση έγινε σε δομημένα και σε μη-δομημένα επιφανειακά πλέγματα, με τα τελευταία να παρουσιάζουν εμφανώς μικρότερη ακρίβεια σε σχέση με τα πρώτα λόγω της ύπαρξης ακανόνιστων δομών των τριγωνικών στοιχείων, ανομοιογένειας στην πυκνότητα των στοιχείων, κόμβων με ελάχιστο αριθμό γειτονικών στοιχείων ή άλλων ιδιαιτεροτήτων που δεν είναι εμφανείς στην περίπτωση των δομημένων. Σημαντικά μέγιστα σφάλματα αποδόθηκαν σε τοπικές ανωμαλίες του πλέγματος, για τις οποίες υπάρχει τρόπος πρόληψης μέσω κατάλληλης προ-επεξεργασίας του πλέγματος με βάση τους δείκτες ποιότητας του πλέγματος. Παράλληλα, δύναται αντιμετώπιση κάποιων ανωμαλιών κατά τη διάρκεια υπολογισμού της καμπυλότητας. Για παράδειγμα, έντονα

σφάλματα λόγω επιμήκυνσης των τριγώνων σε μία μόνο κατεύθυνση δύναται να αντιμετωπιστούν με τη χρήση διορθωτικών παραγόντων που ελαττώνουν την έντονη διαφορά σε σχέση με τις άλλες δύο κατευθύνσεις.

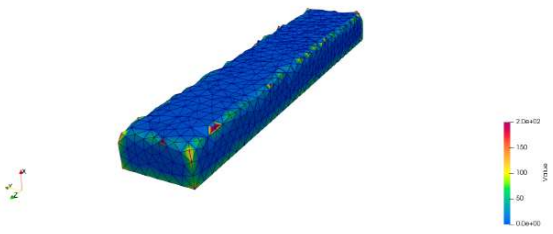
Στόχος της διαδικασίας υπολογισμού της συνολικής καμπυλότητας ανά κόμβο είναι η εξαγωγή ενός συνολικού μέτρου καμπυλότητας για την οριακή επιφάνεια της γεωμετρίας της δομής, ώστε να μπορεί να χρησιμοποιηθεί ως περιορισμός κατά τη διαδικασία βελτιστοποίησης μορφής ή τοπολογίας. Η χρήση ξεχωριστού περιορισμού για κάθε κόμβο θα καθιστούσε το πρόβλημα πολύ-περιορισμένο και θα επηρέαζε αρνητικά τη σύγκλιση της βελτιστοποίησης. Για αυτόν τον λόγο, υιοθετείται η συνάρτηση συσσώρευσης *p-norm*, η οποία επιτρέπει την ομαλή και διαφορίσιμη προσέγγιση της μέγιστης τιμής ενός πεδίου κατανεμημένων περιορισμών. Συγκεκριμένα, για $p = 3$ η συνολική καμπυλότητα δίνεται από τη σχέση:

$$f_3(\boldsymbol{\kappa}) = \left(\sum_{i=1}^{n} \kappa_i^3 \right)^{\frac{1}{3}} \tag{5}$$

Η παραπάνω μέθοδος εφαρμόστηκε σε πρόβλημα ελαχιστοποίησης τραχύτητας επιφάνειας. Μια αρχικά τραχιά επιφάνεια, Σχήμα 9α, εξομαλύνεται μέσω βελτιστοποίησης σχήματος, οδηγώντας σε πιο λεία γεωμετρία, Σχήμα 9β. Η αρχική επιφάνεια εμφάνιζε υψηλότερη συνολική καμπυλότητα ($\kappa_{\text{τοτ}} = 862.559$) σε σχέση με τη βελτιστοποιημένη ($\kappa_{\text{τοτ}} = 763.604$), γεγονός που αποτελεί σε αυτήν την εφαρμογή ένας δείκτης για την τραχύτητα. Παρατηρείται ότι, παρόλο που η διαφορά στις τιμές της συνολικής καμπυλότητας δεν είναι τόσο έντονη όσο υποδεικνύει η οπτική διαφορά, αυτό οφείλεται σε μικρές περιοχές με υψηλή τοπική καμπυλότητα που δεν εξομαλύνθηκαν επαρκώς. Η συνάρτηση p-norm ενισχύει την επίδραση αυτών των περιοχών, αποδεικνύοντας την αποτελεσματικότητά της στην ανάδειξη των πιο προβληματικών σημείων της επιφάνειας.



(α′) *Μέτρο συνολικής καμπυλότητας αρχικής επιφάνειας:* $\kappa_{\text{τοτ}} = \boldsymbol{862.559}$.



(β′) *Μέτρο συνολικής καμπυλότητας εξομαλυμένης επιφάνειας:* $\kappa_{\text{τοτ}} = \boldsymbol{763.604}$.

Σχήμα 9: *Συνολική καμπυλότητα επιφάνειας πριν και μετά από την εξομάλυνση τραχύτητας.*

Συμπεράσματα και Προτάσεις Μελλοντικής εργασίας

Η ανάλυση των μεθόδων υπολογισμού καμπυλότητας που προτείνονται στη βιβλιογραφία ανέδειξε σημαντικά ζητήματα ακρίβειας, ακόμη και για απλές γεωμετρίες, με τη βασική αιτία να είναι ο λανθασμένος υπολογισμός των επιφανειών που ανατίθενται στους κόμβους. Μία νέα μέθοδος εισήχθηκε για το σκοπό αυτό, η SGAC Μέθοδος Voronoi, που λαμβάνει υπόψη τη γεωμετρία των αμβλειών τριγώνων και βελτιώνει σημαντικά την ακρίβεια του υπολογισμού καμπυλότητας, μειώνοντας το μέγιστο σφάλμα κάτω από 1%. Η μέθοδος είναι εφαρμόσιμη τόσο σε δομημένα όσο και σε μη-δομημένα πλέγματα. Όσον αφορά την εφαρμογή της ολικής καμπυλότητας ως περιορισμό στη βελτιστοποίηση, θα πρέπει να ληφθούν υπόψη τα παρακάτω:

- Η καμπυλότητα, εκ φύσεως, είναι μέγεθος οριζόμενο σημειακά, ενώ σε λογισμικά για CFD προσομοιώσεις, για λόγους συνέπειας στην αποθήκευση δεδομένων με τα πεδία ροής που ορίζονται στις φάτσες, προτείνεται να υπολογίζεται αρχικά στους κόμβους με τις προτεινόμενες μεθόδους και στη συνέχεια να παρεμβάλλεται στις συνοριακές επιφάνειες.

- Λόγω της εξάρτησης της καμπυλότητας από τη γεωμετρία των τριγωνικών στοιχείων, συνιστάται η κανονικοποίηση των τιμών καμπυλότητας κάθε επιφανειακού στοιχείου ως εξής:

$$\kappa_{f_n} = \frac{\kappa_f S_f}{\sum S_f} \tag{6}$$

  όπου S_f είναι η επιφάνεια κάθε στοιχείου και $\sum S_f$ το άθροισμα όλων των επιφανειών. Αυτή η κανονικοποίηση εξασφαλίζει ότι τα μεγαλύτερα στοιχεία έχουν ανάλογα μεγαλύτερη επιρροή στη συνολική μέτρηση.

- Είναι απαραίτητη η επιλογή κατάλληλης συνάρτησης συσσώρευσης για τη μετατροπή των σημειακών περιορισμών καμπυλότητας σε έναν ενιαίο αριθμητικό περιορισμό, προκειμένου να επιτευχθεί καλύτερη σύγκλιση κατά τη βελτιστοποίηση. Προτείνεται η συνάρτηση συσσώρευσης τύπου $p\text{-}norm$, καθώς επιτρέπει την ομαλή και διαφορίσιμη προσέγγιση της μέγιστης τιμής ενός πεδίου περιορισμών.

Τονίστηκε επίσης η σημασία της ποιότητας του πλέγματος για την ακρίβεια στον υπολογισμό της καμπυλότητας, ιδίως σε μη-δομημένα πλέγματα όπου οι τοπικές ανωμαλίες είναι πιο συχνές. Σε περιπτώσεις όπου η βελτίωση της ποιότητας του πλέγματος δεν είναι εφικτή, προτείνονται μέτρα για τη διατήρηση της αξιοπιστίας της μεθόδου, όπως η επιπλέον επαλήθευση της καμπυλότητας σε κόμβους με μικρό αριθμό γειτόνων, αντλώντας πληροφορίες από γείτονες δευτέρου βαθμού, καθώς και η χρήση τοπικής προσέγγισης με δευτεροβάθμιες επιφάνειες για την αποφυγή απειρισμού της καμπυλότητας, π.χ. σε περιοχές όπου η επιφάνεια παρουσιάζει απότομες αλλαγές ή τσακίσματα.