# Non-Parametric Conserving Smoothing Method of Area, Volume and Mass for Piecewise Linear Curves and Surfaces

**Prepared by**: Bayan Noufal

**Supervisor**: Dr.Ahmad Khamayseh

M.Sc.Thesis

Submitted to the Department of Mathematics at Palestine Polytechnic University as a partial fulfillment of the requirement for the degree of Master of science.

Hebron-Palestine

The program of Graduated Studies
Department of mathematics
Deanship of Graduate Studies and Scientific Research

# Non-Parametric Conserving Smoothing Method of Area, Volume and Mass for Piecewise Linear Curves and Surfaces

**Prepared by: Bayan Noufal**

**Supervisor**
**Dr.Ahmad Khamayseh**

2019

# Declaration

I declare that the master thesis entitled ”Non-Parametric Conserving Smoothing Method of Area, Volume and Mass for Piecewise Linear Curves and Surfaces” is my own work, and here by certify that unless stated, all work contained within this thesis is my own independent research and has not been submitted for the award of any other degree at any institution, except where due acknowledgment is made in the text.

*Bayan Noufal*

*Signature:*........................................ *Date:*........................................

# Dedication

*To my parents,*
*To my dear husband Mohammad and*
*To Fatima Rasheed.*

*Bayan Noufal*

# Acknowledgment

I am very grateful to my supervisor Dr.Ahmad Khamayseh for all his help and encouragement.

I am very grateful to my internal examiner Dr.Nureddin Rabie for his valuable suggestion on this thesis.

I thank my external examiner Dr.Yousef Manasrah for his useful comments and advice.

Also, my thanks to the members of the department of mathematics at Palestine Polytechnic University.

## Abstract

The interest in curve/surface smoothing has increased in many fields, such as computer graphics and computational fluid mechanics, due to it's various applications in these fields. Different approaches have been used to smooth surfaces such as Laplacian approach which is a very popular approach. However, in many simulations, area, volume, mass and sometimes shape have to be preserved. In this work, we present several non-parametric smoothing algorithms which preserve area, volume and mass of curve or surface grids depending on Laplacian algorithm. A new application of area conserving smoothing method of planar polygonal regions in the space will be presented.

# Contents

# Introduction

In computer graphics, triangular meshes are very popular. Creating triangular meshes of high complexity using three dimension scanning is not difficult, but how to effectively process such meshes remains a challenging problem. For example, three dimension scanning systems commonly give rise to noisy meshes due the factors related to measurement [11]. This problem is one of the problems smoothing method has solved. The basic idea of surface smoothing is to minimize the surface energy or to remove the high angular deviation from surfaces. Some smoothing algorithms [10,13] are based on the Laplacian approach which has unwanted result "Shrinkage effect". To avoid this effect, another algorithms such as [15,11] which doesn't generate shrinkage have been developed.

In this work, we present a modified approach which is an area conserving smoothing of a polygon in the space. The thesis is organized as follows:

**Chapter 1:** Presents vectors and some basic geometric concepts.

**Chapter 2:** Presents smoothing and some smoothing algorithms (Laplacian smoothing and Taubin smoothing).

**Chapter 3:** Presents two algorithms of area conserving smoothing and a modified method of conserving the area of a polygon in the space.

**Chapter 4:** Introduces two algorithms of volume conserving smoothing.

**Chapter 5:** Introduces the Trapezoidal sub grid undulations removal, a method aims to preserve mass during smoothing in two dimensions and three dimensions.

# Chapter 1

# Vectors and Basic Geometric Objects

## 1.1 Vectors in space

Vectors are used to express the geometric objects such as lines, planes, shapes and solids.

In this section, a brief definitions and some operations on vectors will be performed.

**Definition 1.1.** *[3] A **nonzero vector** is an arrow directed from an initial point $P$ to a terminal point $Q$ with $P$ and $Q$ are distinct. The vector denoted by $\overrightarrow{PQ}$.*

**Definition 1.2.** *A vector $\boldsymbol{v}$ has a **magnitude** which is the length of the arrow, i.e the length of the line segment between $P$ and $Q$ denoted by $||\boldsymbol{v}||$, and it has a **direction** which is indicated by the arrow. If a vector has an initial point at the origin $(0, 0, ..., 0)$ and it's terminal is $\boldsymbol{v}$= $(v_1, v_2, ..., v_n)$, then we simply write $\overrightarrow{\boldsymbol{v}}$.*

**Definition 1.3.** *[1] The **metric**(distance) is a function $d : \boldsymbol{X} \times \boldsymbol{X} \to [0, \infty)$, where $\boldsymbol{X}$ is a non empty set, which satisfies the following axioms:*

1. *$d(X, Y) \geq 0$, $\forall X, Y \in \boldsymbol{X}$,* *(Non negativity property)*

2. *If $d(X, Y) = 0$, then $X = Y$,* *(Identity property)*

3. *$d(X, Y) = d(Y, X)$, $\forall X, Y \in \boldsymbol{X}$,* *(Symmetry property)*

*4. $d(X,Z) \leq d(X,Y) + d(Y,Z)$, $\forall X,Y,Z \in \boldsymbol{X}$,* *(Triangle inequality)*

*A set $\boldsymbol{X}$ with a metric function is called a **metric space**.*

**Definition 1.4.** *[2] Let $\boldsymbol{X} = \mathbb{R}^n$, $X = (x_1, x_2, ..., x_n)$, $Y = (y_1, y_2, ..., y_n)$ be two elements in $\boldsymbol{X}$, and the function $d : \boldsymbol{X} \times \boldsymbol{X} \to [0, \infty)$, then the **Euclidean metric** $(\boldsymbol{X}, d)$ is a metric space given by:*

$$\begin{aligned} d(X,Y) &= ||X - Y||_2 \\ &= \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \end{aligned}$$

**Definition 1.5.** *If $\boldsymbol{v} = \overrightarrow{PQ}$ is a vector in $\mathbb{R}^n$ with the initial point $P = (p_1, p_2, ..., p_n)$ and the terminal point $Q = (q_1, q_2, ..., q_n)$, then the **magnitude** of $\overrightarrow{PQ}$ is:*

$$||\overrightarrow{PQ}|| = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + ... + (q_n - p_n)^2}$$

**Definition 1.6.** *[3] Two vectors are said to be **equal** if they have the same magnitude and the same direction.*

**Definition 1.7.** *[3] If $\boldsymbol{v} = (v_1, v_2, ..., v_n)$ is a nonzero vector and $k$ is a scalar, then the scalar multiple $k\boldsymbol{v}$ is the vector $(kv_1, kv_2, ..., kv_n)$, it has the same direction as $\boldsymbol{v}$ if $k > 0$ and the opposite direction if $k < 0$. The magnitude of $k\boldsymbol{v}$ is $|k|||\boldsymbol{v}||$.*

**Definition 1.8.** *[3]**Vectors Addition***
*Let $\boldsymbol{u} = (u_1, u_2, ..., u_n)$ and $\boldsymbol{v} = (v_1, v_2, ..., v_n)$ be two vectors, then $\boldsymbol{u} + \boldsymbol{v}$ is define as:*

$$\boldsymbol{u} + \boldsymbol{v} = (u_1 + v_1, u_2 + v_2, ..., u_n + v_n)$$

For geometric representation of **u** and **v** consider Figure 1.1.
In the case, $\mathbf{u} - \mathbf{v} = \mathbf{u} + (-\mathbf{v})$

**Definition 1.9.** *[3] Two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ are **parallel** if one of them is a scalar multiple of the other.*

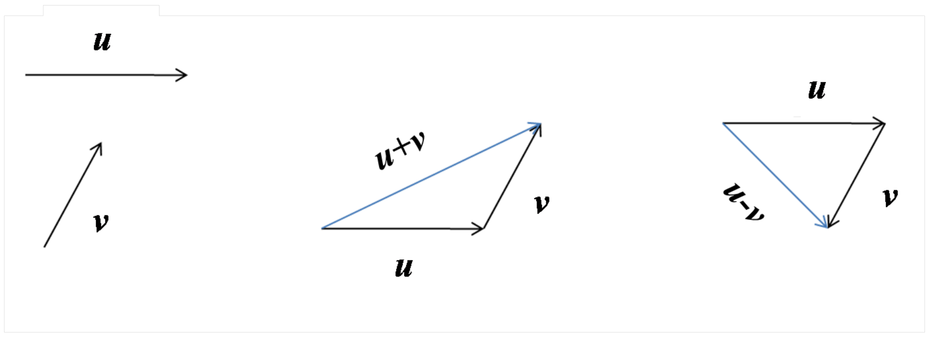


Figure 1.1: Vector Addition.

**Definition 1.10.** *[5]**Cross Product***
*Let **u** and **v** be two vectors in* $\mathbb{R}^n$*, then the **cross product** of **u** and **v** is the vector defined as:*

$$\boldsymbol{u} \times \boldsymbol{v} = (||\boldsymbol{u}||||\boldsymbol{v}|| \sin\theta)\hat{\boldsymbol{n}}$$

*Where:*
$\hat{\boldsymbol{n}}$*: The unit normal vector.*
$\theta$*: The angle between **u** and **v**,* $0 \leq \theta \leq 180$*.*

**Note 1.1.** *[5]*

- *The way we choose* $\theta$ *from **u** to **v** or the converse, determines the direction of* $n$*.*
- *The magnitude of the cross product gives the area determined by **u** and **v**.*
- *The direction of the cross product is the normal vector to both **u** and **v** determined by the right-hand rule.*
- *Cross product can be calculated by determinant, let **u** and **v** be two vectors with components relative to a cartesian coordinate system, i.e*

$$\boldsymbol{u} = u_1 i + u_2 j + u_3 k, \;\; \boldsymbol{v} = v_1 i + v_2 j + v_3 k$$

*Where* $i$*,*$j$ *and* $k$ *are unit vectors in the* $x$*,* $y$ *and* $z$ *directions respectively, then:*

$$\boldsymbol{u} \times \boldsymbol{v} = \begin{vmatrix} i & j & k \\ u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \end{vmatrix}$$

- $||\boldsymbol{u} \times \boldsymbol{v}|| = ||\boldsymbol{u}||||\boldsymbol{v}|| \sin\theta$.

**Definition 1.11.** *[5]**Dot Product***
*Let $\boldsymbol{u}$ and $\boldsymbol{v}$ be two vectors in $\mathbb{R}^n$, then the **dot product** of $\boldsymbol{u}$ and $\boldsymbol{v}$ denoted by $\boldsymbol{u} \cdot \boldsymbol{v}$ is given by:*

- $\boldsymbol{u} \cdot \boldsymbol{v} = ||\boldsymbol{u}||||\boldsymbol{v}|| \cos\theta, \quad 0 \leq \theta \leq 180.$
  *$\theta$ is the angle between $\boldsymbol{u}$ and $\boldsymbol{v}$.*
- $\boldsymbol{u} \cdot \boldsymbol{v} = \sum_{i=1}^{n} u_i v_i.$

**Note 1.2.** *[3]*

- *Let $\boldsymbol{u}$ and $\boldsymbol{v}$ be two vectors in $\mathbb{R}^n$, then $\boldsymbol{u}$ and $\boldsymbol{v}$ are perpendicular if and only if $\boldsymbol{u} \cdot \boldsymbol{v} = 0$.*
- *$\boldsymbol{u} \cdot \boldsymbol{v} > 0$ when $0 \leq \theta < 90$.*
- *$\boldsymbol{u} \cdot \boldsymbol{v} < 0$ when $90 < \theta < 180$.*
- *If $\boldsymbol{u} = (u_1, u_2)$, then the normal vector $\boldsymbol{u}^{\perp} = (-u_2, u_1)$ since $\boldsymbol{u} \cdot \boldsymbol{u}^{\perp} = u_1(-u_2) + u_2 u_1 = 0$.*
- $||\boldsymbol{u} \times \boldsymbol{v}||^2 = (\boldsymbol{u} \cdot \boldsymbol{u})(\boldsymbol{v} \cdot \boldsymbol{v}) - (\boldsymbol{u} \cdot \boldsymbol{v})^2.$

**Definition 1.12.** *[3] The **normalization** of vector $\boldsymbol{v}$ is a unit vector with the same direction of $\boldsymbol{v}$ and it's computed by:$\dfrac{\boldsymbol{v}}{||\boldsymbol{v}||}$.*

By Euclidean metric, if $\overrightarrow{PQ}$ is a vector in $\mathbb{R}^n$ with the initial point $P = (p_1, p_2, ..., p_n)$ and the terminal point $Q = (q_1, q_2, ..., q_n)$ the magnitude of $\overrightarrow{PQ}$ is:

$$||\overrightarrow{PQ}|| = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + ... + (q_n - p_n)^2} \ .$$

Consider $P$ is the origin, then $\mathbf{v} = \overrightarrow{PQ}$ has a magnitude:

$$||\mathbf{v}|| = \sqrt{q_1^2 + q_2^2 + ... + q_n^2} \ .$$

Vector $\mathbf{v}$ in this case indicates a point $Q$ in the Euclidean space, i.e a **point** $Q$ is a position vector such that $\mathbf{v}$ is directed from the origin to the point $Q = (q_1, q_2, ..., q_n)$.

**Definition 1.13.** *Let $\boldsymbol{X}$ be a vector space over a field $\boldsymbol{K}$. Then a function $\langle .,. \rangle : \boldsymbol{X} \times \boldsymbol{X} \to \boldsymbol{K}$ with the properties:*

- $\langle X, X \rangle \geqslant 0,$ *(Non negativity property)*
- $\langle X, X \rangle = 0$ *if and only if* $X = 0,$ *(Definiteness property)*
- $\langle X, Y \rangle = \overline{\langle Y, X \rangle},$ *(Symmetry property)*
- $\langle \alpha X + \beta Y, Z \rangle = \alpha \langle X, Z \rangle + \beta \langle Y, Z \rangle,$ *(Linearity property)*
  *For all $X, Y, Z \in \boldsymbol{X}$ and $\alpha, \beta \in \boldsymbol{K}$ is called an inner product space on $\boldsymbol{X}$.*

**Note 1.3.** *Every inner product space is a metric space with the Euclidean metric defined by $d(X,Y) = \sqrt{\langle X - Y, X - Y \rangle}$, but the converse is not always true, take $L_1$ space.*

## 1.2 Basic geometric objects

## Lines in the space

**Definition 1.14.** *[7] A **line** is a straight geometric object that is infinitely long and has zero thin.*

In the plane, a line is determined by a point $(x_1, y_1)$ and the slope $m$ of the line by the equation:

$$y - y_1 = m(x - x_1)$$

In the space, we can determine line by one of the followings [6]:

1. Point and direction.
   A vector equation of the line $L$ through the point $P(x_0, y_0, z_0)$ and parallel to vector $\mathbf{v}$ is:
   $$L(t) = P + t\mathbf{v}, \qquad -\infty < t < \infty$$
   where:
   $t$ is a parameter takes different values for each point. See Figure 1.2.

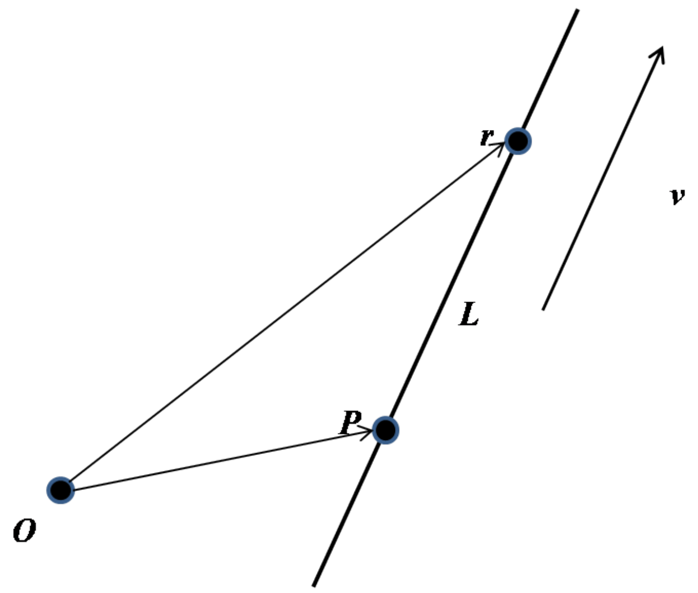


Figure 1.2: Line L given by a point and a vector.

2. Two points [14].

   A vector equation of a line $L$ passes through two points $P$ and $Q$ is defined as:

$$L(t) = Q + t(P - Q), \qquad -\infty < t < \infty$$

   where:

   $t$ is a parameter takes different values for each point.

**Note 1.4.** *[1]*

- ***Line segment*** *is a part of a line that is bounded by two endpoints, that contains every point between the ends.*
- *If the line has one endpoint at one side and extend at the other it is called* ***ray****,* $0 \leq t < \infty$.

## 1.3 Planes in the space

**Definition 1.15.** *[7]* ***Plane*** *is a flat surface extending infinitely in all directions.*

We can determine a plane by [6]:

A point and a normal vector to the plane. Let $P$ be a point in the plane and $\mathbf{n}$ is the normal vector to the plane, then the **plane** is the set of all points $R$ such that $(P - R) \cdot \mathbf{n} = 0$.

Now, let $P = (x_0, y_0, z_0)$, $R = (x, y, z)$ in $\mathbb{R}^3$, $\mathbf{n} = ai + bj + ck$.
The equation $(P - R) \cdot \mathbf{n} = 0$ is equivalent to:

$$(ai + bj + ck) \cdot ((x - x_0)i + (y - y_0)j + (z - z_0)k) = 0$$

$$= a(x - x_0) + b(y - y_0) + c(z - z_0) = 0$$

where $i \cdot i = |i|^2 = 1$

$$ax + by + cz = ax_0 + by_0 + cz_0 = d$$

The equation of the plane becomes:

$$ax + by + cz = d$$

# Chapter 2

# Smoothing Algorithms

Curves or surfaces obtained from physical simulations are frequently jagged such as Potts model. When a metal solidifies from the melton state, millions of tiny crystals start to grow. These crystals form the grains in the solid metal see Figure 2.1. Potts model simulations of metallic grain growth describes the interface between these grains as a series of stair steps. The jagged stair step interface might produce incorrect results in subsequence simulations. To avoid such results it has to be smoothed.

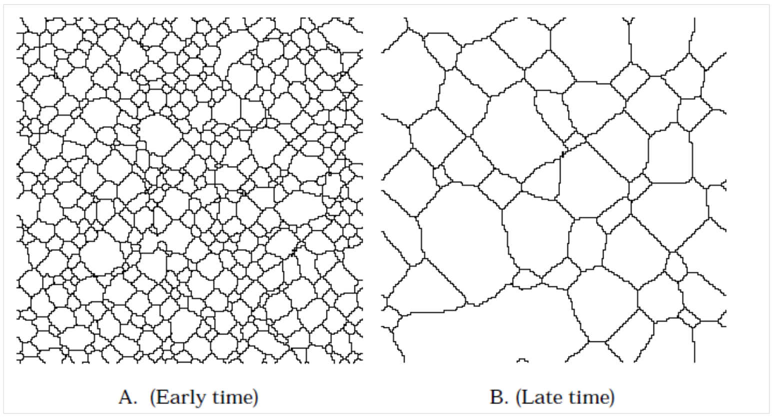


Figure 2.1: Example of grain growth at two different times.

## 2.1 Surface grid smoothing

**Definition 2.1.** *[15] A* ***mesh*** *is defined as the association of vertices(nodes, points), edges and faces that defines the shape of a polyhedral object in solid modeling as in Figure 2.2. The faces in the mesh usually consists of triangles(triangle mesh), quadrilaterals or simple convex(non-self intesating) polygons, it may also be composed of general concave* *polygons(has at least one reflex interior angle) or polygons with holes. The surface of the mesh is a* ***surface grid****.*

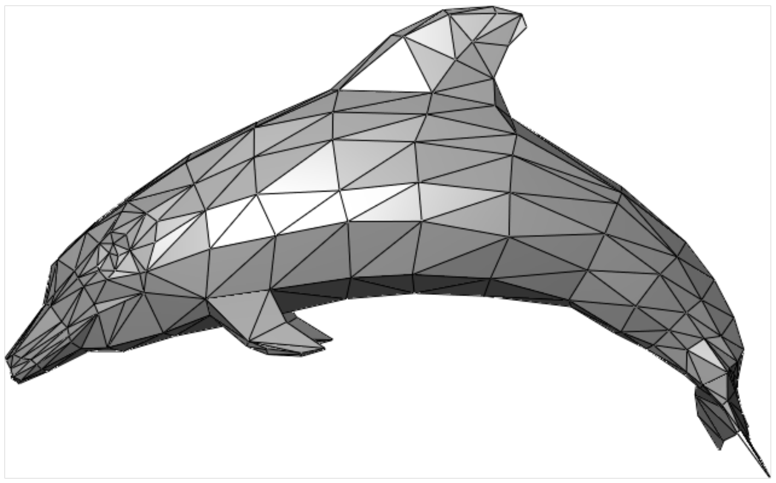


Figure 2.2: Dolphin triangle mesh.

**Definition 2.2.** *[9]* ***Smoothing*** *a surface grid is defined as a process attempts to remove noise with minimal damage caused to geometric features of the object (i.e to capture patterns in the data), by moving it's vertices without changing the connectivity of the edges or removing or adding vertices to the polygon, see [9].*

To achieve smoothing, **three conditions** have to be satisfied [10]:

1. Adjacent facets of the surface grid have normals adjusted to vary more gradually.

2. Nodes densities are equidistributed on the surface.

3. The aspect ratios of facets are improved.

There are several approaches to surface grid smoothing namely **parametric**, **nonparametric** and **curvature** based smoothing.

1. Parametric approach maps from a parametric space to the surface and smoothing the grid in the parametric space, but the mapping to the parametric space is not

always possible, such as a lot of surfaces generated by physical simulations. Parametric approach has another drawback, that is parametric smoothing preserves the shape of the surface and doesn't always preserve the volume enclosed by the surface grid.

2. Curvature is the reciprocal of the radius of the circle passing through $P$ and other points $N$ and $Q$ on the curve infinitesimally close to $P$ [5], curvature removes the jaggedness of the surface, but it has drawbacks, it doesn't preserve volume and it need a sophisticated partial differential equation solution.

In this work, We will use the non-parametric area and volume conserving smoothing of curve grids.

## 2.2 Representation of piecewise linear curve

In this section, we will discuss the representation of piecewise linear curve using vertices and edges in the neighborhood of a vertex.

1. A **polygon** is the region of a plane bounded by a finite collection of line segments forming a simple closed curve [13].

2. For a closed polygon in 2-D or 3-D, it is represented as an ordered list of vertices $\mathbf{X} = \{X_i : 1 \leq i \leq n\}$ where $X_i = (x_i, y_i)$ for curves in 2-D and $X_i = (x_i, y_i, z_i)$ in 3-D.

3. For an open polygon, it is represented as a list of edges $E = \{e_k : k \in \mathbb{N}, 1 \leq k \leq n_E\}$ where $e_k = (i_1^k, i_2^k)$, $i_1^k$ and $i_2^k$ are non-repeated indices of vertices, where $n_E$ is the number of edges in the polygon.

4. The neighborhood of a vertex $X_i$ in a graph is the set $i^*$ of all vertices adjacent to $X_i$ including $X_i$ itself. If vertex $X_j$ belongs to the neighborhood $i^*$, $X_i$ is called a neighborhood of $X_i$, where $1 \leq i, j \leq n$, $n$ is the number of the vertices of the polygon, see [8].

## 2.3 Laplacian smoothing

The most common mesh smoothing algorithms is Laplacian smoothing. This algorithm moves vertices of the gird iteratively to the center of mass of it's neighboors. Laplacian smoothing has two types: Firstly, The traditional locally Laplacian smoothing. Secondly, Global Laplacian smoothing.

## Traditional local Laplacian smoothing

Laplacian smoothing has an operator called **Laplacian Operator** [8], the discrete Laplacian operator can be approximated at each vertex by:

$$L(X_i) \quad = \quad \sum_{j \in i^*} w_{ij}(X_j - X_i) \tag{2.1}$$

where:
$i^*$: The set of indices of neighborhood vertices of the vertex $X_i$.
$w_{ij}$: The weight of edge $(i, j)$.

**Note 2.1.** *[9]*

1. *For each vector $\boldsymbol{x}_i$, we have:*
$$\sum_{j \in i^*} w_{ij} = 1$$
2. *Several weighting schemes used like edge length scheme $w_{ij} = \frac{1}{|e_{ij}|}$ and cotangent scheme $w_{ij} = \cot \alpha_i + \cot \beta_j$, where $\alpha_i$ and $\beta_j$ are the opposite of the edge $e_{ij}$ in the two triangles that shares $e_{ij}$.*

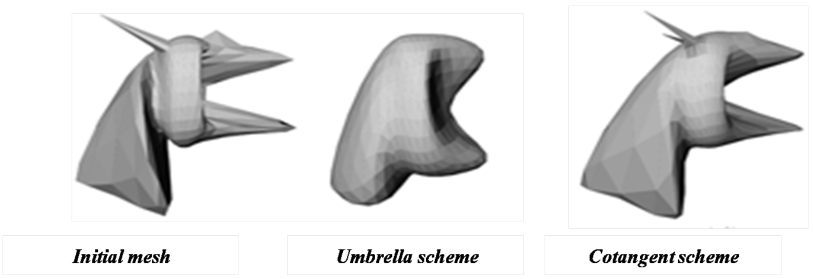


Figure 2.3: [4] Weighting Schemes.

3. *A simple choice of weight (uniform umbrella) $w_{ij} = \frac{1}{d_i}$ where $d_i$ is the number of neighbors of the vertex $X_i$. For example, for the polygon curve each vertex has two neighbors, so $w_{ij} = \frac{1}{2}$, see Figure 2.4.*

4. *In the continuous case, the Laplacian operator is defined as:*
$$\Delta f = \bigtriangledown^2 f = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2}, \quad \textit{where } \bigtriangledown f \textit{ is the deviation gradient.}$$

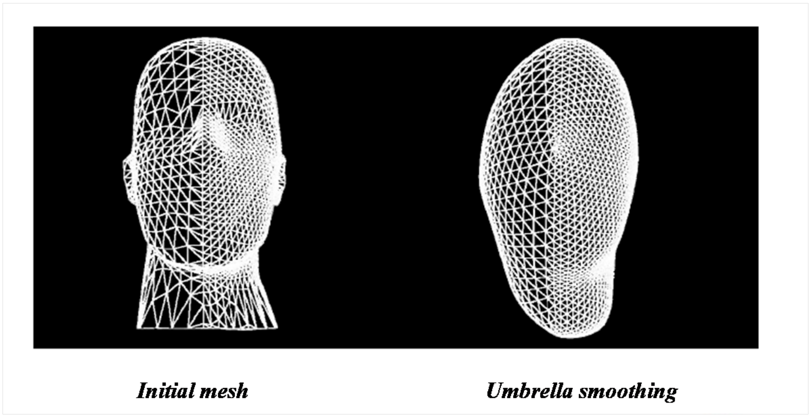


Figure 2.4: [4] Uniform Umbrella.

5. *The only difference between the operators is how they calculate the weights, see [9].*

Now, the equation (2.1) can be written in a matrix form as:

$$L(X_i) = -(I - W)X = -kX$$

where:
I: The identity matrix.
W: The weight matrix, $W = \begin{cases} w_{ij}, & \text{if vertex } j \text{ is a neighbor of vertex } i \\ 0, & \text{otherwise} \end{cases}$

After calculating Laplacian operator, we move each vertex by the formula:

$$X_i' = X_i + \lambda L(X_i) \tag{2.2}$$

The equation (2.2) can be written in matrix form as:

$$X' = (I - \lambda k)X, \quad 0 < \lambda < 1$$

where $\lambda$ is a scalar factor controls smoothing speed.

Next, iterate for several times, for further smoothing increase the number of iterations. The iterative step is expressed as:

$$X^N = (1 - \lambda k)^N X$$

where $N$ is the number of iterations.

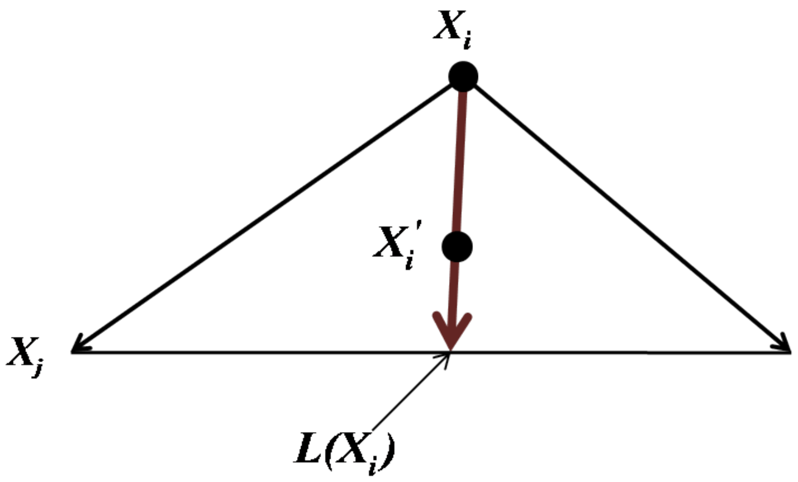


Figure 2.5: Discrete Laplacian.

## Global Laplacian smoothing

Instead of moving vertices locally and iteratively in the Traditional Laplacian smoothing, vertices can be removed globally, which is given in Global Laplacian smoothing. The smooth condition of a vertex $X_i$ is that $L(X_i) = 0$ where $L(X_i)$ as in (2.1), which means that vertex $X_i$ lies in the weighted average of it's 1-ring neighbors, but if we choose $w_{ij} = \frac{1}{di}$ where $d_i$ = number of elements in $i^*$, then $X_i$ lies in the center of gravity of it's 1-ring neighbors. Vertices in (2.2) forms a linear system which can be represented in a matrix form as:

$$LX = 0$$

where:

L: $n \times n$ matrix with elements $L_{ij} = \begin{cases} 1, & i = j \\ -w_{ij}, & \text{if vertex } j \text{ is a neighbor of vertex } i \\ 0, & \text{otherwise} \end{cases}$

X: $n \times 1$ column vector of the corresponding vertices.

$L$ has $(n-1)$ rank for a connected mesh surface which means $X$ will not be uniquely determined unless a vertex is given.

If we know the geometry of some vertices in the mesh, we can find the geometry for the rest of the vertices by solving the linear system. By the choice $w_{ij} = \frac{1}{d_{ij}}$ each vertex $X_j$ will be as close as possible to the weighted center, which means that the vertices will be distributed in a fair way. Thus, the mesh will be approximated in a global way and usually smoothed.

Global Laplacian smoothing works under constrants such as feature constrants which aims to keep the features of the original mesh surface, but gives undesired distortion and shrinkage in region of non feature vertices as shown in Figure 2.6 due to exceeded relaxation on the vertices caused by Laplacian operator. To minimize such relaxation, we fix all the triangle barycenters in positions during smoothing which gives a new extra constrants called Barycenter constrants as shown in Figure 2.7.

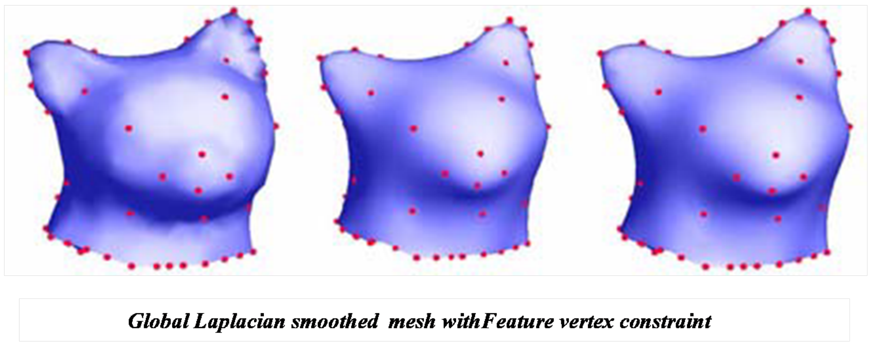

Global Laplacian smoothed mesh withFeature vertex constraint

Figure 2.6: [9] Global Laplacian using Feature Constraint.

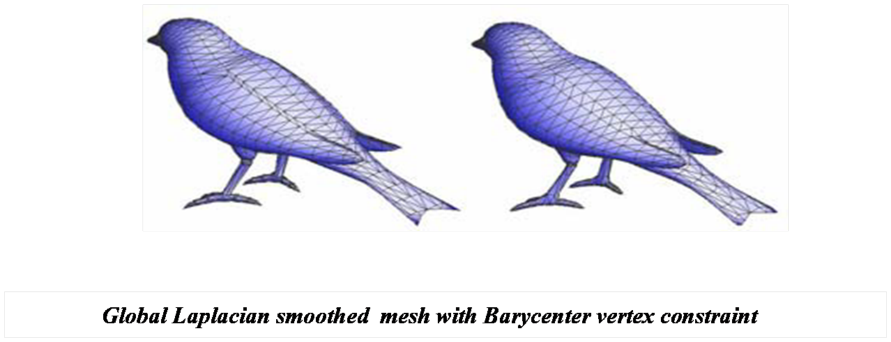

Global Laplacian smoothed mesh with Barycenter vertex constraint

Figure 2.7: [9] Global Laplacian using Barycenters Constraint.

In the process of iterating Laplacian algorithm, the shape will finally collapses to a point, this effect is called shrinkage. Preventing a mesh from shrinking has became one of the major problems in mesh smoothing. So several approaches have been developed such as Taubin and Novel volume constrained smoothing method.

## 2.4 Taubin smoothing

Taubin smoothing consists of two consecutive Laplacian smoothing steps. After a first Laplacian smoothing step with a positive scale $\lambda$ is applied to the all vertices of the piecewise linear shapes, a second Laplacian smoothing step is applied to all the vertices with a negative scale factor $\mu$ where $0 < \lambda < -\mu$, see [16].

**Algorithm 1.** *[16] Taubin smoothing algorithm:*

```
DO $i = 1, ..., N$.
$\Delta X \leftarrow$ Laplacian operator $\lambda, \mu$.
$X \leftarrow X + \lambda \Delta X$.
$X' \leftarrow X + \mu \Delta X$.
```

**Note 2.2.** *[16] The two steps have to be repeated, alternating the positive and negative scale factors several times in order to produce a significant smoothing effect. Taubin suggested a way to choose all of the parameter, First, we choose the pass-band frequency $K_{PB}$ (Taubin suggests to choose $k_{PB} = 0.1$), the number of iterations $N$ and $\lambda$, then $\mu = \dfrac{\lambda}{K_{PB}^{-1}}$.*

# Chapter 3

# Non-Parametric Area Conserving Smoothing

Curves or surfaces obtained from physics-based simulations are oftenly nonsmooth which may be in appropriate for later simulations. However, one desire that any smoothing operation be area conservative with respect to an enclosed region.

This chapter develops a non-parametric area conserving smoothing approach which rapidly deforms a ”stair stepped” closed curve into a smoothed curve, both of the curves have the same enclosed area. This approach allows deformation of the shape. The degree of deformation will depend on the number of sweeps (iterations) performed.

## 3.1 Area conserving smoothing of a closed plane curve by single-node relaxations method

Suppose $\gamma = \{X_0, X_1, ..., X_{n-1}, X_n = X_0\}$ is a closed non-self intersecting curve in $\mathbb{R}^2$ consisting of **n** line segments and encloses region $R$ with area $A$, see Figure 3.1. The signed area defined by the points $X_0, X_1, ..., X_n$, taken in a counter clockwise direction.

Our aim is to find a smoothing operation that can be applied locally at each point $X_j$ such that it's position changes slightly depending on the adjacent data points (say$\{X_{j-m}, X_{j-m+1}, ..., X_{j+m}\}$, $m$ is small) and does not change the area.

The smoothing operation may not depend on all points in the neighborhood $\{X_{j-m}, X_{j-m+1}, ..., X_{j+m}\}$, i.e, we may only need to change the position of one or more

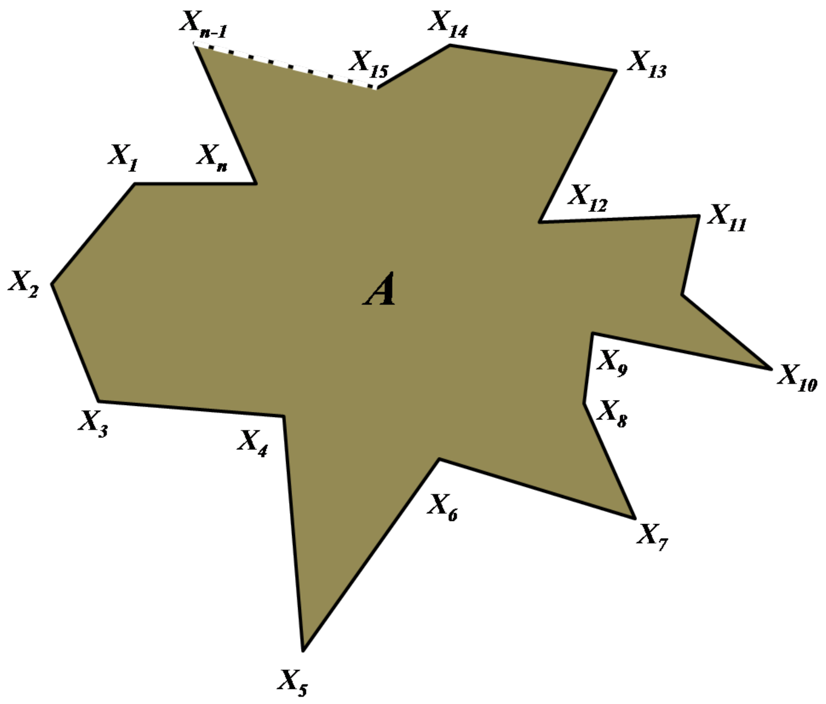


Figure 3.1: Closed curve $\gamma$ surrounds an area $A$.

points in the neighborhood .

Applying the local smoothing operation in each local neighborhood in the curve in some order, this is called a **sweep**. The number of sweeps needed in the smoothing operation through the curve has to be as small as possible.

Now, back to Figure 3.1, choose the points $X_0$, $X_1$and $X_2$ of curve $\gamma$, where $X_0 = (x_0, y_0)$, $X_1 = (x_1, y_1)$ and $X_2 = (x_2, y_2)$. These points form the triangle $\Delta$ $X_0X_1X_2$, see Figure 3.2 .

In Figure 3.2, the area of triangle $\Delta$ $X_0X_1X_2$ is given by the relation [3]:

$$\begin{aligned} A_1 &= \frac{1}{2}||(X_2 - X_0) \times (X_1 - X_0)|| \\ &= \frac{1}{2}\begin{vmatrix} i & j & k \\ x_2 - x_0 & y_2 - y_0 & 0 \\ x_1 - x_0 & y_1 - y_0 & 0 \end{vmatrix} \end{aligned}$$

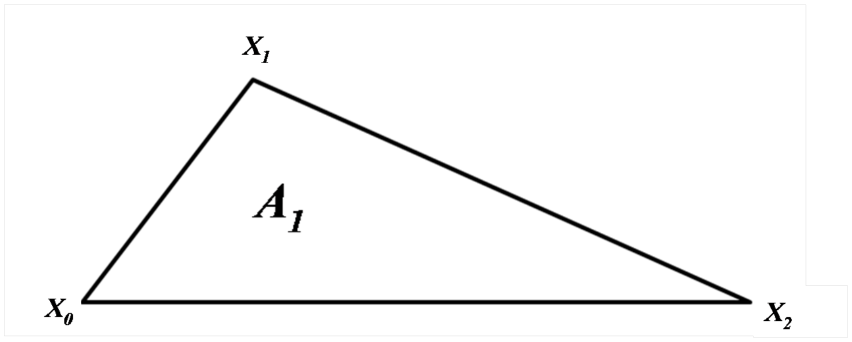


Figure 3.2: The triangle $\Delta\ X_0X_1X_2$ encloses area $A_1$.

where:

$$\begin{aligned}(X_2 - X_0) &= ((x_2 - x_0), (y_2 - y_0))\\ (X_1 - X_0) &= ((x_1 - x_0), (y_1 - y_0)\end{aligned}$$

Thus,

$$\begin{aligned}A_1 &= \frac{1}{2}[(x_2 - x_0)(y_1 - y_0) - (x_1 - x_0)(y_2 - y_0)]\\ &= \frac{1}{2}[((x_1 - x_0), (y_1 - y_0)) \cdot (-(y_2 - y_0), (x_2 - x_0)]\end{aligned}$$

but,

$$\begin{aligned}(X_2 - X_0)^{\perp} &= (X_2 - X_0)e^{i\frac{\pi}{2}}, \quad i = \sqrt{-1}\\ &= [(x_2 - x_0) + i(y_2 - y_0)]i\\ &= [-(y_2 - y_0), (x_2 - x_0)]\end{aligned}$$

Then,

$$A_1 = \frac{1}{2}(X_1 - X_0) \cdot (X_2 - X_0)^{\perp}$$

Now, we aim to change the position of the point $X_1$ such that the area $A_1$ does not change, this can be accomplished by moving $X_1$ parallel to the line segment $\overline{X_0X_2}$. Also the projection of the line segments $\overline{X_0X_1}$ and $\overline{X_1X_2}$ onto $\overline{X_0X_2}$ will be equal if we move $X_1$ such that it's projection onto $\overline{X_0X_2}$ is in the middle distance between $X_0$ and $X_2$. See Figure 3.3.

The one-point smoothing operation will be as follows:

$$A_1 = \frac{1}{2}(X_2 - X_0)^{\perp} \cdot (X_1 - X_0)$$

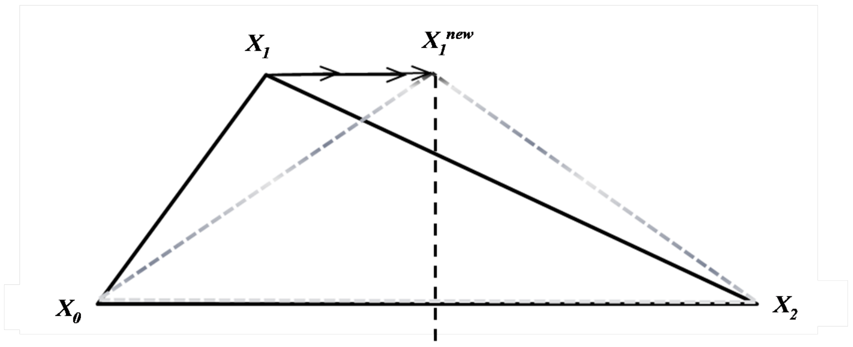


Figure 3.3: One-point smoothing operation: Movement of $X_1$ parallel to $\overline{X_0X_2}$.

This is the signed area of the triangle $\Delta\ X_0X_1X_2$.

Now, we figure the height h of $X_1$ above the base $\overline{X_0X_2}$ from $A_1 = \frac{1}{2}||X_2 - X_0|| \cdot h$, then, $h = \frac{2A_1}{||X_2 - X_0||}$. The unit normal to the base line $\overline{X_0X_2}$ is $\hat{\mathbf{n}} = \frac{(X_2-X_0)\perp}{||(X_2-X_0)\perp||}$. Thus, the new position of $X_1$, $X_1^{new} = \frac{1}{2}(X_0 + X_2) + h\hat{\mathbf{n}}$. By this way we complete through the nodes in sequential order.

**Algorithm 2.** *[10] Area conserving smoothing of a closed plane curve area by single-node relaxations method.*

- *Choose some $i$ from $0$ to $n-1$, $n$ is the number of vertices of the curve.*
- *Apply smoothing operation on neighborhood $\{X_i, X_{i+1}, X_{i+2}\}$.*
- *Compute local area.*

$$\begin{aligned} A_{i,i+2,i+1} &= \frac{1}{2}(X_{i+2} - X_i)^{\perp} \cdot (X_{i+1} - X_i) \\ \hat{\boldsymbol{n}} &= \frac{(X_{i+2} - X_i)^{\perp}}{||(X_{i+2} - X_i)^{\perp||}} \\ h &= \frac{2A_{i,i+2,i+1}}{||(X_{i+2} - X_i)||} \\ X_{i+1} &= \frac{1}{2}(X_i + X_{i+2}) + h\hat{n}. \end{aligned}$$

- *Sweep for $i = 0, 1, ..., n-1$.*

**Remark 3.1.** *[10] If $\gamma$ is an open curve i.e $X_n \neq X_0$, we don't relax the endpoints.*

**Remark 3.2.** *[10] The previous algorithm has a drawback since it does not work for all shapes. In Figure 3.4, consider the direction of $\overline{X_0X_2}$ to be the direction tangential to $\gamma$ and the orthogonal direction to $\overline{X_0X_2}$ to be the normal direction, applying the one-point smoothing operation smooths only in the tangential direction as explained previously, which means smoothing in the normal direction using single node method is not possible since it doesn't change the positions of nodes (i.e the shape will remain the same). Figure 3.4 shows a star-shaped region cant' be smoothed by Algorithm* 1.

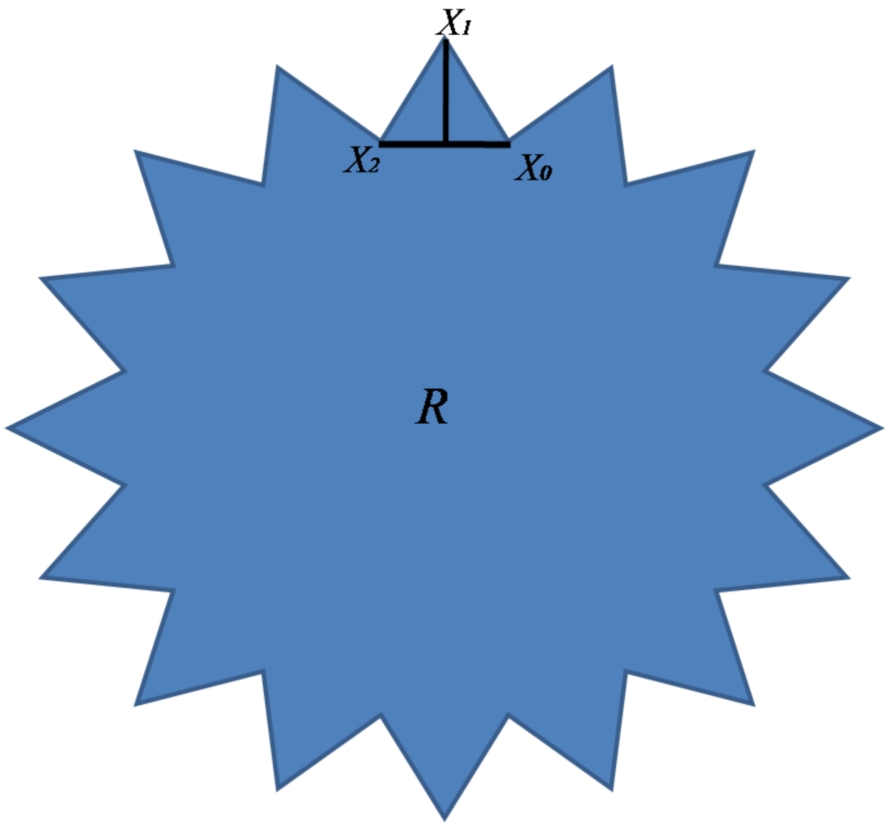


Figure 3.4: Star-shaped region unchanging under algorithm 1.

## 3.2 Area conserving smoothing of a closed plane curve by edge relaxations method

Single-node smoothing doesn't smooth all shapes such as star shaped regions as in Figure 3.4. A new local smoothing operation developed to include normal smoothing called **Area conserving smoothing by edge relaxation**.

Now, consider four points along $\gamma$ call $X_0$, $X_1$, $X_2$ and $X_3$ as in Figure 3.5. We take $\overrightarrow{X_0X_3}$ to be the direction tangential to the curve. Changing $X_1$ and $X_2$ positions simultaneously under area conservation condition will allow smoothing in the normal direction.

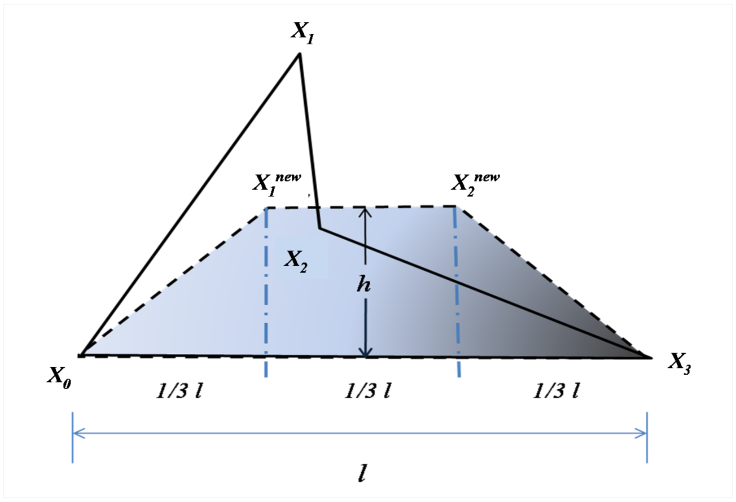


Figure 3.5: Two points smoothing. $\overline{X_1X_2}$ moved to be parallel to $\overline{X_0X_3}$.

In Figure 3.5, move $X_1$ and $X_2$ such that the projection of $X_1$ onto $\overline{X_0X_3}$ equals one third of $\overline{X_0X_3}$ length and the projection of $X_2$ equals two thirds of $\overline{X_0X_3}$ length ,their projections are equally spaced and both $X_1$ and $X_2$ have the same distance $h$ from $\overline{X_0X_3}$ where $h$ is taken to conserve area. The quadrilateral area $A_{0123}$ must be conserved. The shape becomes trapezoid with two bases $\overline{X_0X_3}$ with length $l$ and $\overline{X_1^{new}X_2^{new}}$ with length $\frac{1}{3}l$ and so the quadrilateral area is:

$$\begin{aligned} A_{0123} &= \frac{1}{2}h(l + \frac{1}{3}l) \\ A_{0123} &= \frac{2}{3}hl \end{aligned}$$

thus,

$$\begin{aligned} h &= \frac{3}{2}\frac{A_{0123}}{l} \\ X_1^{new} &= X_0 + \frac{1}{3}(X_3 - X_0) + h\hat{n} \\ &= \frac{2}{3}X_0 + \frac{1}{3}X_3 + h\hat{n} \end{aligned}$$

This smoothing is called a smoothing operation on the edge $\overline{X_1X_2}$. Next, we smooth on the edge $\overline{X_2X_3}$.

**Algorithm 3.** *[10] Conserving smoothing of a closed plane curve area by edge relaxations method.*

- *Choose some $i$ from $0$ to $n-1$, $n$ is the number of vertices of the curve.*
- *Apply smoothing operation on neighborhood $\{X_i, X_{i+1}, X_{i+2}, X_{i+3}\}$.*
- *Compute local area.*

$$\begin{aligned} A_{i,i+3,i+2,i+1} &= \frac{1}{2}(X_{i+3} - X_i)^{\perp} \cdot (X_{i+2} - X_i) + \frac{1}{2}(X_{i+2} - X_i)^{\perp} \cdot (X_{i+1} - X_i) \\ \boldsymbol{\hat{n}} &= \frac{(X_{i+3} - X_i)^{\perp}}{||(X_{i+3} - X_i)^{\perp}||} \\ h &= \frac{3}{2}\frac{A_{i,i+3,i+2,i+1}}{||(X_{i+3} - X_i)||} \\ X_{i+1} &= \frac{2}{3}X_i + \frac{1}{3}X_{i+3} + h\hat{n} \\ X_{i+2} &= \frac{1}{3}X_i + \frac{2}{3}X_{i+3} + h\hat{n}. \end{aligned}$$

- *Sweep for $i = 0, 1, ..., n-1$.*

Figure 3.6, shows the result of applying algorithm 3 with 10 sweeps.

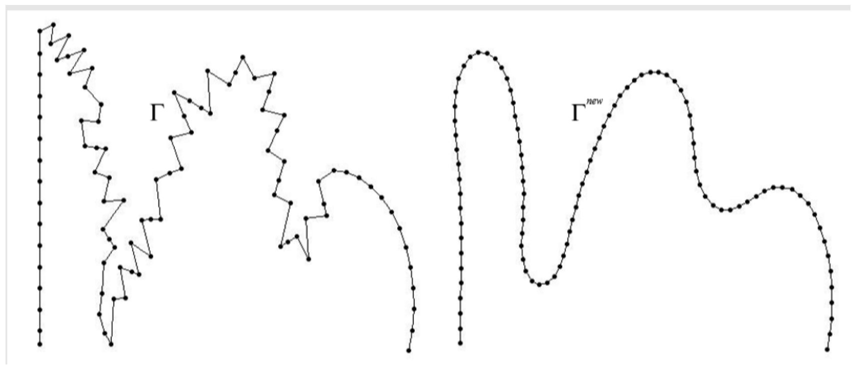


Figure 3.6: [10] An open curve before and after applying algorithm 3 with 10 sweeps.

## 3.3 Area conserving smoothing of a polygon in the space

Consider the triangle $X_0X_1X_2$ in the space as in Figure 3.7.

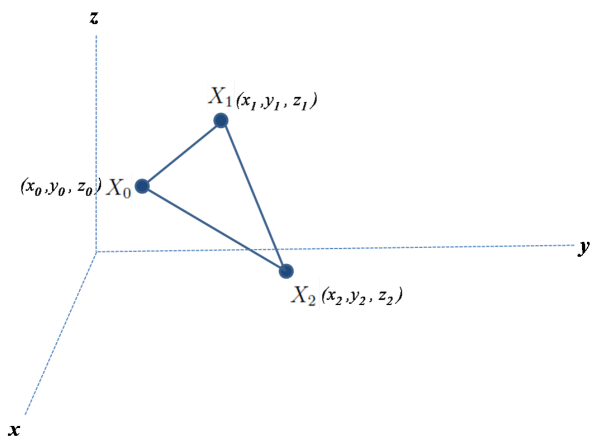


Figure 3.7: Triangle in the space.

The parametric equation of the triangle is given by:

$$T(s,t) = (X_1 - X_0)s + (X_2 - X_0)t + X_0$$

where $0 \leq s,t \leq 1$ and $0 \leq s + t \leq 1$.

**Proposition 3.1.** *[13] Every triangle is convex.*

*Proof.* The line joins two points $a$ and $b$ is given by:

$$L(\tau) = (1-\tau)a + \tau b, \quad 0 \leq \tau \leq 1$$

Let $a$ and $b$ be any two arbitrary points inside $\Delta X_0X_1X_2$ then,

$$a = (X_1 - X_0)s_1 + (X_2 - X_0)t_1 + X_0$$

and

$$b = (X_1 - X_0)s_2 + (X_2 - X_0)t_2 + X_0$$

where $0 \leq s_1, s_2, t_1, t_2 \leq 1$ and $0 \leq s_1 + t_1 \leq 1$ and $0 \leq s_2 + t_2 \leq 1$

To show that the line $L(\tau)$ lies inside the triangle $T(s,t)$, we have to show that for any arbitrary point $c$ lies on $L(\tau)$, $c$ also lies inside $T(s,t)$.

Now, $c \in L(\tau)$, then:

$$\begin{aligned} c &= (1-\tau)a + \tau b, \quad for \;\; some \;\; 0 \leq \tau \leq 1 \\ &= (1-\tau)[(X_1 - X_0)s_1 + (X_2 - X_0)t_1 + X_0] + \tau[(X_1 - X_0)s_2 + (X_2 - X_0)t_2 + X_0] \\ &= (X_1 - X_0)[(1-\tau)s_1 + \tau s_2] + (X_2 - X_0)[(1-\tau)t_1 + \tau t_2] + X_0 \end{aligned}$$

Let $s^* = (1-\tau)s_1 + \tau s_2$ and $t^* = (1-\tau)t_1 + \tau t_2$, for $c$ to lie inside $T(s,t)$ $s^*$ and $t^*$ must satisfy:

1. $0 < s^*, t^* < 1$ and this is true since $0 < \tau, s_1, s_2, t_1, t_2 < 1$.
2. $0 < s^* + t^* < 1$, this also true since:

$$s^* + t^* = (1-\tau)s_1 + \tau s_2 + (1-\tau)t_1 + \tau t_2 = (1-\tau)(s_1+t_1) + \tau(s_2+t_2) < (1-\tau) + \tau = 1$$

Thus, $T(s,t)$ is convex. □

We want to smooth $\Delta X_0X_1X_2$ without changing it's area. The idea is to map the triangle from $\mathbb{R}^3$ to $\mathbb{R}^2$ and to smooth in $\mathbb{R}^2$, then mapping the smoothed triangle back to $\mathbb{R}^3$.

The equation of the plane containing $\Delta X_0X_1X_2$ is given by:

$$(X - X_c) \cdot \mathbf{n} = 0$$

where: $\mathbf{n} = (X_1 - X_0) \times (X_2 - X_0)$ is the normal vector.
$X_c$ is any point in the plane.
(We choose $X_c$ to be not $X_0, X_1$ or $X_2$).

We will decompose each $X_i, i = 0, 1, 2$ into an orthonormal basis $\hat{\mathbf{n}} - \hat{\mathbf{u}} - \hat{\mathbf{v}}$ which is:

1. $\hat{\mathbf{n}}$ is the unit normal vector to the plane, i.e $\hat{\mathbf{n}} = \dfrac{\mathbf{n}}{||\mathbf{n}||}$.
2. $\mathbf{u} = X_1 - X_0$

   $$\hat{\mathbf{u}} = \frac{X_1 - X_0}{||X_1 - X_0||}$$
3. $\mathbf{v} = \hat{\mathbf{n}} \times \hat{\mathbf{u}}$

   $$\hat{\mathbf{v}} = \frac{\hat{\mathbf{n}} \times \hat{\mathbf{u}}}{||\hat{\mathbf{n}} \times \hat{\mathbf{u}}||}$$

Now, the decomposition is:

$$\begin{aligned} X_i &= (X_i \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} + (X_i \cdot \hat{\mathbf{u}})\hat{\mathbf{u}} + (X_i \cdot \hat{\mathbf{v}})\hat{\mathbf{v}} \\ &= (X_c \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} + (X_i \cdot \hat{\mathbf{u}})\hat{\mathbf{u}} + (X_i \cdot \hat{\mathbf{v}})\hat{\mathbf{v}}, \quad i = 0, 1, 2 \end{aligned}$$

Since $(X_i - X_c) \cdot \hat{\mathbf{n}} = 0$.
The mapping of $X_i$ from $\mathbb{R}^3$ to $\mathbb{R}^2$ is given by:

$$X_i \rightarrow (X_i \cdot \hat{\mathbf{u}}, X_i \cdot \hat{\mathbf{v}}) = X_i^{new} = X_i' = (x_i', y_i'), \quad i = 0, 1, 2$$

The mapping of $\Delta X_0X_1X_2$ is $\Delta X_0'X_1'X_2'$.
Now, single node algorithm is used to smooth $\Delta X_0'X_1'X_2'$ at $X_1'$.

$$X_1'' = \frac{1}{2}(X_0' + X_2') + h\hat{\mathbf{n}} = (x_1'', y_1'')$$

where:

$$\begin{aligned} h &= \frac{2A_{012}}{||(X_2' - X_0')^{\perp}||} \\ \hat{\mathbf{n}} &= \frac{(X_2' - X_0')^{\perp}}{||(X_2' - X_0')^{\perp}||} \\ A_{012} &= \frac{1}{2}(X_2' - X_0')^{\perp} \cdot (X_1' - X_0') \end{aligned}$$

After smoothing at $X_1'$, the new triangle is $X_0'X_1''X_2'$.
Now, we map points $X_0'$, $X_1''$ and $X_2'$ back to $\mathbb{R}^3$ by:

$$\begin{aligned}
X_0' &= (X_c \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} + x_0'\hat{\mathbf{u}} + y_0'\hat{\mathbf{v}} \\
X_1'' &= (X_c \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} + x_1''\hat{\mathbf{u}} + y_1''\hat{\mathbf{v}} \\
X_2' &= (X_c \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} + x_2'\hat{\mathbf{u}} + y_2'\hat{\mathbf{v}}
\end{aligned}$$

Consider the polygon $X_0X_1...X_m$ as in Figure 3.8. All points lie on the plane $X_c \cdot \mathbf{n}$.

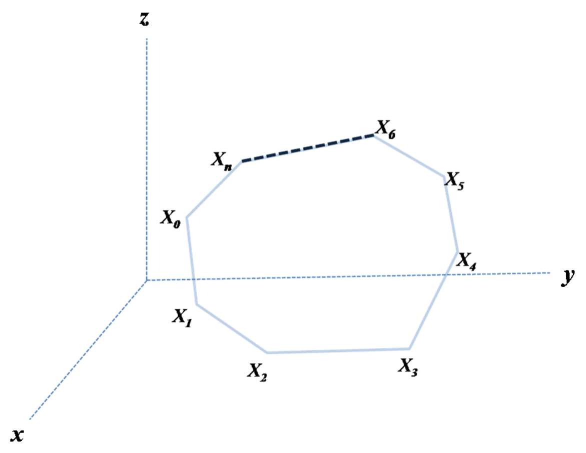


Figure 3.8: Polygon in the space.

$X_c$ is another point in the plane where $\mathbf{n}$ is the normal to the plane. It's determined by any two vectors in the plane, for example take $X_1 - X_0$ and $X_m - X_0$.

$$\begin{aligned}
\mathbf{n} &= (X_m - X_0) \times (X_1 - X_0) \\
&= \begin{vmatrix} i & j & k \\ x_m - x_0 & y_m - y_0 & z_m - z_0 \\ x_1 - x_0 & y_1 - y_0 & z_1 - z_0 \end{vmatrix}
\end{aligned}$$

1. Find the orthonormal basis $\hat{\mathbf{n}} - \hat{\mathbf{u}} - \hat{\mathbf{v}}$ where:

$$\begin{aligned}
\mathbf{n} &= (X_m - X_0) \times (X_1 - X_0) \\
\hat{\mathbf{n}} &= \frac{\mathbf{n}}{||\mathbf{n}||} \\
\mathbf{u} &= \frac{X_1 - X_0}{||X_1 - X_0||} \\
\hat{\mathbf{u}} &= \frac{\mathbf{u}}{||\mathbf{u}||} \\
\mathbf{v} &= \hat{\mathbf{n}} \times \hat{\mathbf{v}} \\
\hat{\mathbf{v}} &= \frac{\hat{\mathbf{n}} \times \hat{\mathbf{v}}}{||\hat{\mathbf{n}} \times \hat{\mathbf{v}}||}
\end{aligned}$$

2. Each $X_i, \quad i = 0, 1, ..., m$ is decomposed into the $\hat{\mathbf{n}} - \hat{\mathbf{u}} - \hat{\mathbf{v}}$ basis such that:

$$X_i = (X_c \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} + (X_i \cdot \hat{\mathbf{u}})\hat{\mathbf{u}} + (X_i \cdot \hat{\mathbf{v}})\hat{\mathbf{v}}$$

3. The mapping of $X_i$ from $\mathbb{R}^3$ to $\mathbb{R}^2$ is given by:

$$X_i \to (X_i \cdot \hat{\mathbf{u}}, X_i \cdot \hat{\mathbf{v}}) = X_i^{new} = X_i' = (x_i', y_i'), \quad i = 0, 1, ..., m$$

   The new polygon in $\mathbb{R}^2$ is $X_0'X_1'...X_m'$.

4. One of the algorithm (2 and 3) is used to conserve the area. The resulting shape $X_0''X_1''...X_m''$.

5. Now, map $X_0''X_1''...X_m''$ back to $\mathbb{R}^3$ by:

$$X_i''' = (X_c \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} + x_i''\hat{\mathbf{u}} + y_i''\hat{\mathbf{v}}, \quad i = 0, 1, ..., m$$

   where $X_i'' = (x_i'', y_i'')$.

**Example 3.1.** *Let $X_0X_1X_2$ be a triangle, $X_0 = (1, 1, 1)$, $X_1 = (3, 3, 2)$, $X_2 = (2, 3, -1)$, apply area conserving smoothing of a triangle in the space.*

**Solution:**

1. The area of $\Delta X_0X_1X_2$ is:

$$A = \frac{1}{2}||(X_2 - X_0) \times (X_1 - X_0)|| = \frac{\sqrt{65}}{2} \ .$$

2. $X_0X_2 = (1, 2, -2)$ and $X_0X_1 = (2, 2, 1)$.

$$\begin{aligned}
\mathbf{n} &= X_0X_1 \times X_0X_2 \\
&= \begin{vmatrix} i & j & k \\ 2 & 2 & 1 \\ 1 & 2 & -2 \end{vmatrix} \\
&= i(-4-2) - j(-4-1) + k(4-2) \\
&= -6i + 5j + 2k\ .
\end{aligned}$$

$\Rightarrow \mathbf{n} = (-6, 5, 2)$.
$X_c = (0, 1, -2)$ lies on the plane, then the plane equation is: $6x + 5y + 2z - 1 = 0$.

$$\hat{\mathbf{n}} = \frac{\mathbf{n}}{||\mathbf{n}||} = \frac{-6i + 5j + +2k}{\sqrt{(-6)^2 + 5^2 + 2^2}} = \frac{-6i + 5j + 2k}{\sqrt{65}} = (\frac{-6}{\sqrt{65}}, \frac{5}{\sqrt{65}}, \frac{2}{\sqrt{65}}).$$

Let $\mathbf{u} = X_0X_1 \Rightarrow \hat{\mathbf{u}} = \frac{\mathbf{u}}{||\mathbf{u}||} = (\frac{2}{3}, \frac{2}{3}, \frac{1}{3})$.

Let $\mathbf{v} = \hat{\mathbf{n}} \times \hat{\mathbf{u}} = \begin{vmatrix} i & j & k \\ \frac{-6}{\sqrt{65}} & \frac{5}{\sqrt{65}} & \frac{2}{\sqrt{65}} \\ \frac{2}{3} & \frac{2}{3} & \frac{1}{3} \end{vmatrix} = \frac{1}{3\sqrt{65}}i + \frac{10}{3\sqrt{65}}j - \frac{22}{3\sqrt{65}}k$

$\Rightarrow \hat{\mathbf{v}} = \frac{\mathbf{v}}{||\mathbf{v}||} = (\frac{1}{3\sqrt{65}}, \frac{10}{3\sqrt{65}}, \frac{-22}{3\sqrt{65}})$ .

The orthonormal basis is $\hat{\mathbf{n}} - \hat{\mathbf{u}} - \hat{\mathbf{v}}$ where:
$\hat{\mathbf{n}} = (\frac{-6}{\sqrt{65}}, \frac{5}{\sqrt{65}}, \frac{2}{\sqrt{65}})$, $\hat{\mathbf{u}} = (\frac{2}{3}, \frac{2}{3}, \frac{1}{3})$, $\hat{\mathbf{v}} = (\frac{1}{3\sqrt{65}}, \frac{10}{3\sqrt{65}}, \frac{-22}{3\sqrt{65}})$ .

3. We decompose each $x_i$, $i = 0, 1, 2$ into $\hat{\mathbf{n}} - \hat{\mathbf{u}} - \hat{\mathbf{v}}$:
$X_0 = (X_0 \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} + (X_0 \cdot \hat{\mathbf{u}})\hat{\mathbf{u}} + (X_0 \cdot \hat{\mathbf{v}})\hat{\mathbf{v}}$
$= \frac{1}{\sqrt{65}}\hat{\mathbf{n}} + \frac{5}{3}\hat{\mathbf{u}} + \frac{-11}{3\sqrt{65}}\hat{\mathbf{v}}$ .

$X_1 = (X_1 \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} + (X_1 \cdot \hat{\mathbf{u}})\hat{\mathbf{u}} + (X_1 \cdot \hat{\mathbf{v}})\hat{\mathbf{v}}$
$= \frac{1}{\sqrt{65}}\hat{\mathbf{n}} + \frac{14}{3}\hat{\mathbf{u}} + \frac{-11}{3\sqrt{65}}\hat{\mathbf{v}}$ .

$X_2 = (X_2 \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} + (X_2 \cdot \hat{\mathbf{u}})\hat{\mathbf{u}} + (X_2 \cdot \hat{\mathbf{v}})\hat{\mathbf{v}}$

$$= \frac{1}{\sqrt{65}}\hat{\mathbf{n}} + 3\hat{\mathbf{u}} + \frac{18}{\sqrt{65}}\hat{\mathbf{v}} \ .$$

4. The mapping of $\Delta X_0X_1X_2$ is:
   $X_i \to (X_i \cdot \hat{\mathbf{u}}, X_i \cdot \hat{\mathbf{v}}) = (x'_i, y'_i)$
   $X'_0 = (\frac{5}{3}, \frac{-11}{3\sqrt{65}}) = (x'_0, y'_0)$
   $X'_1 = (\frac{14}{3}, \frac{-11}{3\sqrt{65}}) = (x'_1, y'_1)$
   $X'_2 = (3, \frac{18}{\sqrt{65}}) = (x'_2, y'_2)$
   The area of $\Delta X'_0X'_1X'_2$ is $\frac{\sqrt{65}}{2}$.

5. Smoothing $\Delta X'_0X'_1X'_2$ at $X'_1$ using single-node relaxation:
   $X'_2 - X'_0 = (3, \frac{18}{\sqrt{65}}) - (\frac{5}{3}, \frac{-11}{3\sqrt{65}}) = (\frac{4}{3}, \frac{\sqrt{65}}{3})$
   $(X'_2 - X'_0)^{\perp} = (\frac{-\sqrt{65}}{3}, \frac{4}{3})$
   $||(X'_2 - X'_0)^{\perp}|| = \sqrt{\frac{65}{9} + \frac{16}{9}} = 3$
   $A_{012} = \frac{1}{2}(\frac{-\sqrt{65}}{3}, \frac{4}{3}) \cdot (3, 0) = \frac{-\sqrt{65}}{2}$
   $h = \frac{2 \cdot \frac{\sqrt{65}}{2}}{3} = \frac{\sqrt{65}}{3}$
   $\mathbf{n} = (\frac{-\sqrt{65}}{3}, \frac{4}{3}) \times \frac{1}{3} = (\frac{-\sqrt{65}}{9}, \frac{4}{9})$
   $X''_1 = \frac{1}{2}(X'_0 + X''_2) + h\mathbf{n}$
   $X''_1 = \frac{1}{2}\left[(\frac{5}{3}, \frac{-11}{3\sqrt{65}}) + (3, \frac{18}{\sqrt{65}})\right] + \frac{\sqrt{65}}{3}(\frac{-\sqrt{65}}{9}, \frac{4}{9}) = (\frac{-4}{54}, \frac{907}{54\sqrt{65}}) = (x''_1, y''_1)$
   The area of $\Delta X'_0X''_1X'_2$ is $\frac{\sqrt{65}}{2}$.

6. We now map $\Delta X'_0X''_1X_2$ back to $\mathbb{R}^3$
   $X_0 \to (X_c \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} + x'_0\hat{\mathbf{u}} + y'_0\hat{\mathbf{v}} = (\frac{-6}{65}, \frac{5}{65}, \frac{2}{65}) + (\frac{10}{9}, \frac{10}{9}, \frac{5}{9}) + (\frac{-11}{585}, \frac{-110}{585}, \frac{242}{585})$
   $= (1, 1, 1)$

$$X_1^{new} = (\frac{-6}{65}, \frac{5}{65}, \frac{2}{65}) + \frac{-4}{54}(\frac{2}{3}, \frac{2}{3}, \frac{1}{3}) + \frac{907}{54\sqrt{65}}(\frac{1}{3\sqrt{65}}, \frac{10}{3\sqrt{65}}, \frac{-22}{3\sqrt{65}}) \backsimeq (-0.056, 0.889, -1.889)$$

$$X_2 \to \frac{1}{\sqrt{65}}\hat{\mathbf{n}} + 3\left(\frac{2}{3}, \frac{2}{3}, \frac{1}{3}\right) + \frac{18}{\sqrt{65}}\left(\frac{1}{3\sqrt{65}}, \frac{10}{3\sqrt{65}}, \frac{-22}{3\sqrt{65}}\right) = (2, 3, -1)$$

The final triangle $\Delta X_0 X_1^{new} X_2$ is:
$(1, 1, 1), (-0.056, 0.889, -1.889), (2, 3, -1)$.

The area of $\Delta X_0 X_1^{new} X_2$ is $\frac{\sqrt{65}}{2}$, thus the area of the triangle is conserved.

# Chapter 4

# Non-parametric Volume Conserving Smoothing of Surface Grids

In this chapter, volume is targeted,we want to smooth without changing the volume. Consider a closed surface $\alpha$ subdivided into triangular facets ***i.e*** $\alpha = \cup\, \tau_i$ where $\tau_i$ are planar triangular facets , Smoothing operation will be performed in sweep over a small neighborhoods across the surface in such away the amount of volume surrounded by the triangular facets is conserved.

## 4.1 Volume conserving smoothing of a surface by single-node relaxations

The method used in this section is basically simple, we firstly choose a single node, then we change it's position depending on the adjacent nodes in the neighborhood.

To see this, consider a closed surface $\alpha$ as in Figure 4.1 , the position of node $X$ will be changed depending on the data of nodes $X^{(1)}$, $X^{(2)}$, ..., $X^{(n)}$. Suppose that the node $X$ is moved by $dX^s$ to the position $X^s$,***i.e*** $X^s = X + dX^s$ depending only on the data of the adjacent nodes in the local neighborhood. Such a motion will change the volume enclosed by the surface, so we in addition move $X^s$ by some multiple of wisely chosen direction $h\hat{\mathbf{n}}$, so

$$X^s = X + dX^s + h\hat{\mathbf{n}}$$

let $dX = dX^s + h\hat{\mathbf{n}}$, hence

$$X^s \equiv X + dX$$

$dX$ is the total displacement $X$ went through.

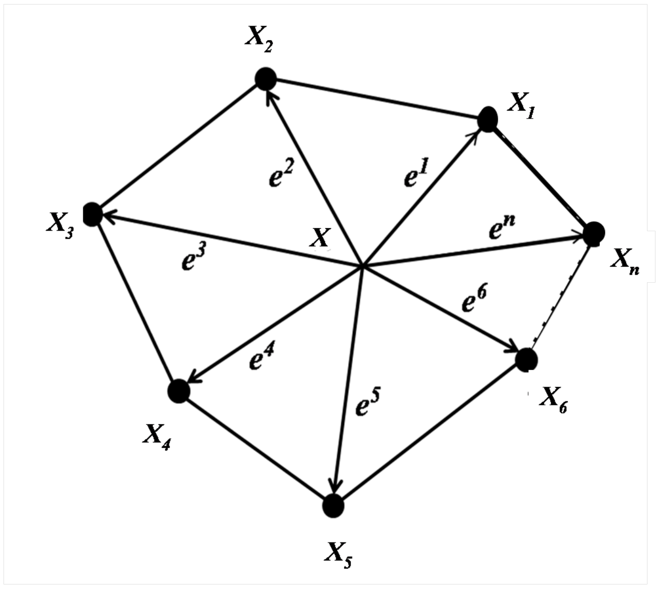


Figure 4.1: Triangular faceted surface.

Figure 4.1 shows a neighborhood cut from a closed surface, this neighborhood forms a polyhedron with $X$ in the center surrounded by n neighbors at $X^{(1)}$, $X^{(2)}$, ..., $X^{(n)}$ when viewed from outside.

To find $h$, divide the polyhedron into n tetrahedrons with $X$ is the common vertex among them all.

$$Tetrahedron\ volume = \frac{1}{6}(e^{(j)} \times e^{(j+1)}) \cdot h$$

hence, the change in the volume of the polyhedron becomes:

$$\begin{aligned} 6dV &= \sum_{j=1}^{n}(dX^s + h\hat{\mathbf{n}}) \cdot e^{(j)} \times e^{(j+1)} \\ &= (dX^s + h\hat{\mathbf{n}}) \cdot \sum_{j=1}^{n} e^{(j)} \times e^{(j+1)} \end{aligned}$$

Volume is conserved, so $dV$=0 which implies:

$$h = \frac{-dX^s \cdot \sum_{j=1}^{n} e^{(j)} \times e^{(j+1)}}{\hat{\mathbf{n}} \cdot \sum_{j=1}^{n} e^{(j)} \times e^{(j+1)}}$$

$\hat{\mathbf{n}}$ has to minimize the norm of $h\hat{\mathbf{n}}$. Since the normal to the undistributed surface of $X$ is the normalized sum of area vectors of all triangles incident on $X$, $\hat{\mathbf{n}}$ considered to be the normal to the undistributed surface of $X$.
thus,

$$\hat{\mathbf{n}} = \frac{\sum_{j=1}^{n} e^{(j)} \times e^{(j+1)}}{||\sum_{j=1}^{n} e^{(j)} \times e^{(j+1)}||}$$

then our minimal corrective movement

$$h\hat{\mathbf{n}} = -(dX^s \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}$$

We use Laplacian smoothing since it yields $X^s$ depending on the data of the nearest neighborhood.

$$X^s = X + dX^s \equiv \frac{\sum_{j=1}^{n} X^{(j)}}{n}$$

where: $X^{(j)}$ is the position of the $j^{th}$ adjacent vertex, and $n$ is the number of adjacent vertices.

**Algorithm 4.** *[10] Volume conserving smoothing of a surface grid by single node relaxation method*

- *Choose some $X$.*
- *Number all adjacent nodes from 1 to $n$, i.e $\{X^{(1)}, X^{(2)}, ..., X^{(n)}\}$.*
- $h = \dfrac{-dX^s \cdot \sum_{j=1}^{n} e^{(j)} \times e^{(j+1)}}{\hat{\boldsymbol{n}} \cdot \sum_{j=1}^{n} e^{(j)} \times e^{(j+1)}}$.
- $\hat{\boldsymbol{n}} = \dfrac{\sum_{j=1}^{n} e^{(j)} \times e^{(j+1)}}{||\sum_{j=1}^{n} e^{(j)} \times e^{(j+1)}||}$.
- $dX^s = \dfrac{\sum_{j=1}^{n} X^{(j)}}{n} - X$.
- $X^s = X + dX^s - (dX^s \cdot \hat{\boldsymbol{n}})\hat{\boldsymbol{n}}$.
- *Sweep until done.*

The previous algorithm has a drawback, again the smoothing in the normal direction which does not change the star-shaped polyhedra.

## 4.2 Volume conversing smoothing of a surface using edge relaxations

To solve the previous problem (smoothing in the normal direction), we will smooth using two adjacent nodes (neighbours), i.e we relax edges on the surface.

In Figure 4.2, choose the two nodes $X_1$ and $X_2$, we relax on the edge $\overline{X_1X_2}$ depending on the position of the surrounding nodes.

In Figure 4.2, call the nodes surrounding $X_1$ $(X_1^{(1)}, X_1^{(2)}, ..., X_1^{(n_1)})$, and the nodes surrounding $X_2$ $(X_2^{(1)}, X_2^{(2)}, ..., X_2^{(n_2)})$.

here,

$$X_1 = X_2^{(1)}$$
$$X_2 = X_1^{(1)}$$

Call the edge connecting $X_i$ with $X_i^{(j)}$ by $e_i^{(j)}$ i.e $e_i^{(j)} = X_i^{(j)} - X_i$.
We define:

$$A_i = \sum_{j=1}^{n_i} e_i^{(j)} \times e_i^{(j+1)}, \quad i = 1, 2$$

We want to move $X_i$ to $X_i^s$ where $X_i^s = X_i + dX_i^s, \quad i = 1, 2$, considering the addition of $(h\hat{\mathbf{n}})$ such that $||h\hat{\mathbf{n}}||$ is minimal and the volume is conserved, by some smoothing algorithm the total displacement for $X_1$ and $X_2$ is:

$$dX_i \;=\; dX_i^s + h\hat{\mathbf{n}}, \quad i = 1, 2 \tag{4.1}$$

Consider Figure 4.3, when moving $X_1$ to $X_1 + dX_1$, the triangles $\{X_1, X_1^{(j)}, X_1^{(j+1)} : 1 \leq j \leq n_1\}$ positions change to the final positions $\{X_1 + dx_1, X_1^{(j)}, X_1^{(j+1)} : 1 \leq j \leq n_1\}$, which causes change in volume, the volume change is equal to the volume of the tetrahedra $\{X_1, X_1 + dX_1, X_1^{(j)}, X_1^{(j+1)} : 1 \leq j \leq n_1\}$.

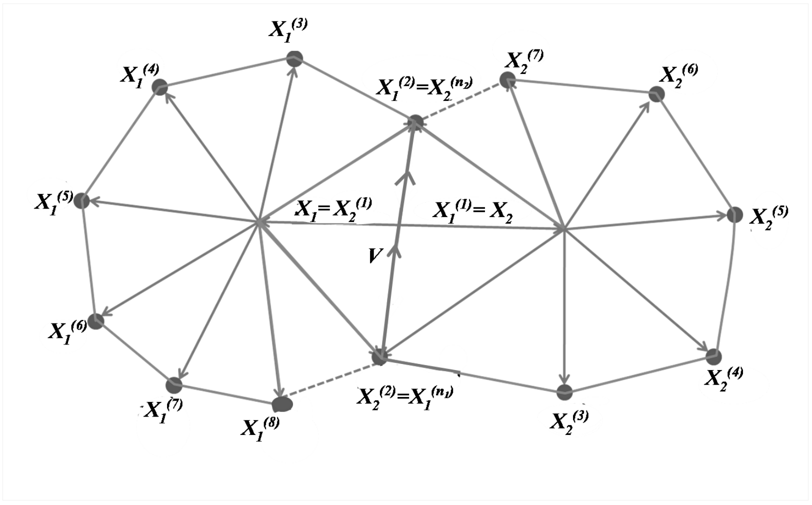


Figure 4.2: Triangular faceted surface containing edge $\overline{X_1X_2}$

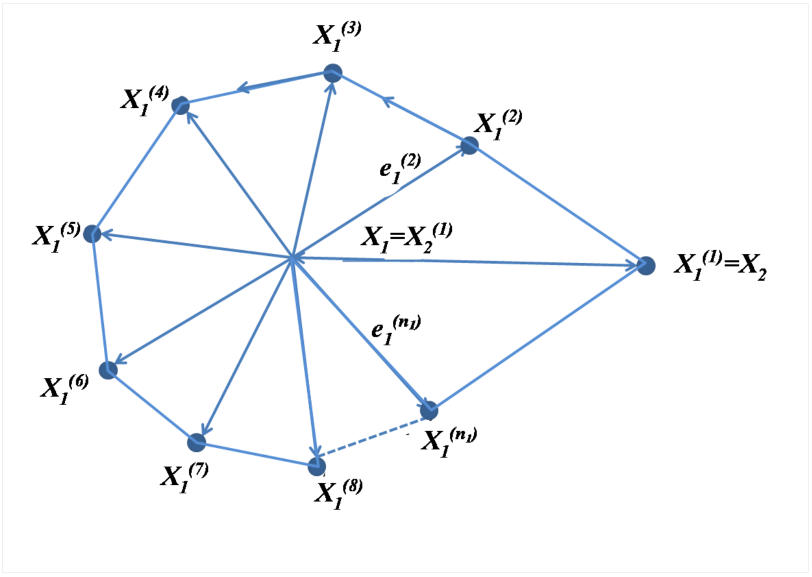


Figure 4.3: Nodes edges surrounding $X_1$

The change in volume in Figure 4.3 is called $d\mathbf{V}_1$, thus

$$\begin{aligned} 6d\mathbf{V}_1 &= \sum_{j=1}^{n_1} dX_1 \cdot e_1^{(j)} \times e_1^{(j+1)} \\ &= dX_1 \cdot A_1 \end{aligned}$$

Now, consider Figure 4.4, again moving $X_2$ to $X_2+dX_2$, changes the volume taking into account that the node at $X_1 = X_2^{(1)}$ has already moved in the previous to $X_1 + dX_1$.

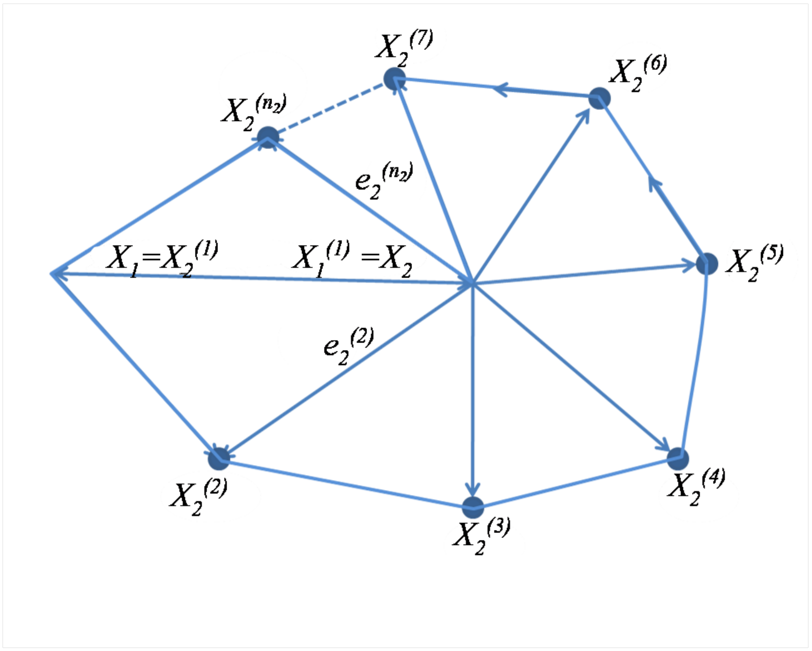


Figure 4.4: Edges $e_2^{(j)}$ connecting nodes to $X_2$.

Now,

$$e_2^{(1)} = X_2^{(1)} - X_2 = X_1 - X_2$$

Call the new $e_2^{(1)}$ by $\widetilde{e_2^{(1)}}$.
thus,

$$\begin{aligned}\widetilde{e_2^{(1)}} &= X_1 + dX_1 - X_2\\ &= e_2^{(1)} + dX_1\\ \widetilde{e_2^{(j)}} &= e_2^{(j)}, \quad 2 \le j \le n_2\end{aligned}$$

The volume change in Figure 4.4, $d\mathbf{V}_2$

$$\begin{aligned}6d\mathbf{V}_2 &= \sum_{j=1}^{n_2} dX_2 \cdot \widetilde{e_2^{(j)}} \times \widetilde{e_2^{(j+1)}}\\ &= dX_2 \cdot \widetilde{e_2^{(1)}} \times \widetilde{e_2^{(2)}} + dX_2 \cdot \widetilde{e_2^{(2)}} \times \widetilde{e_2^{(3)}} + \ldots + dX_2 \cdot \widetilde{e_2^{(n_2)}} \times \widetilde{e_2^{(1)}} \qquad (4.2)\end{aligned}$$

**Note 4.1.** *1.* $dX_2 \cdot \widetilde{e_2^{(1)}} \times \widetilde{e_2^{(2)}} = dX_2 \cdot (e_2^{(1)} + dX_1) \times e_2^{(2)}$
$= dX_2 \cdot e_2^{(1)} \times e_2^{(2)} + dX_2 \cdot dX_1 \times e_2^{(2)}$

*2. Similarly,* $dX_2 \cdot \widetilde{e_2^{(n_2)}} \times \widetilde{e_2^{(1)}} = dX_2 \cdot e_2^{(n_2)} \times e_2^{(1)} + dX_2 \cdot e_2^{(n_2)} \times dX_1$

*3.* $dX_2 \cdot dX_1 \times e_2^{(2)} + dX_2 \cdot e_2^{(n_2)} \times dX_1 = dX_2 \cdot (dX_1 \times e_2^{(2)} + e_2^{(n_2)} \times dX_1)$
$= dX_2 \cdot (-[e_2^{(2)} \times dX_1] + e_2^{(n_2)} \times dX_1)$
$= dX_2 \cdot (e_2^{(n_2)} - e_2^{(2)}) \times dX_1$

back to 4.2,

$$\begin{aligned}
6d\mathbf{V}_2 &= \sum_{j=1}^{n_2} dX_2 \cdot e_2^{(j)} \times e_2^{(j+1)} + dX_2 \cdot dX_1 \times e_2^{(2)} + dX_2 \cdot e_2^{(n_2)} \times dX_1 \\
&= dX_2 \cdot A_2 + dX_2 \cdot (e_2^{(n_2)} - e_2^{(2)}) \times dX_1 \\
&= dX_2 \cdot A_2 + dX_2 \cdot v \times dX_1
\end{aligned}$$

where,

$$v = e_2^{(n_2)} - e_2^{(2)} = e_1^{(2)} - e_1^{(n_1)} \quad From\ Figure\, 4.2$$

Since the volume is conserved, we have:

$$6d\mathbf{V} = 6d\mathbf{V}_1 + 6d\mathbf{V}_2 = 0$$

thus,

$$\begin{aligned}
0 &= dX_1 \cdot A_1 + dX_2 \cdot A_2 + dX_2 \cdot v \times dX_1 \\
0 &= (dX_1^s + h\hat{\mathbf{n}}) \cdot A_1 + (dX_2^s + h\hat{\mathbf{n}}) \cdot A_2 + (dX_2^s + h\hat{\mathbf{n}}) \cdot v \times (dX_1^s + h\hat{\mathbf{n}}) \quad From\ 4.1
\end{aligned}$$

then,

$$h = \frac{-dX_1^s \cdot A_1 + dx_2^s \cdot A_2 + dX_2^s \cdot v \times dX_1^s}{\hat{\mathbf{n}} \cdot (A_1 + A_2 + v \times (dX_1^s - dX_2^s))}$$

The value of $\hat{n}$ that minimizes $||h\hat{\mathbf{n}}||$ is:

$$\hat{\mathbf{n}} = \frac{A_1 + A_2 + v \times (dX_1^s - dX_2^s)}{||A_1 + A_2 + v \times (dX_1^s - dX_2^s)||}$$

We will use Laplacian smoothing of both $X_1$ and $X_2$. So,

$$X_1^s = \frac{1}{n_1}(X_2^s + \sum_{j=2}^{n_1} X_1^{(j)}) \qquad (4.3)$$

$$X_2^s = \frac{1}{n_2}(X_1^s + \sum_{j=2}^{n_2} X_2^{(j)}) \qquad (4.4)$$

To compute $X_1^s$, substitute (4.4) in (4.3) yields:

$$X_1^s = \frac{1}{n_1 n_2 - 1} \sum_{j=2}^{n_2} X_2^{(j)} + \frac{n_2}{n_1 n_2 - 1} \sum_{j=2}^{n_1} X_1^{(j)}$$

Then we compute $X_2^s$ by (4.4).

For some problems, under relaxed Laplacian smoothing is used:

$$X_i^s \leftarrow (1-w)\ X_i + wX_i^s, \ \ i = 1, 2 \qquad (4.5)$$

where:

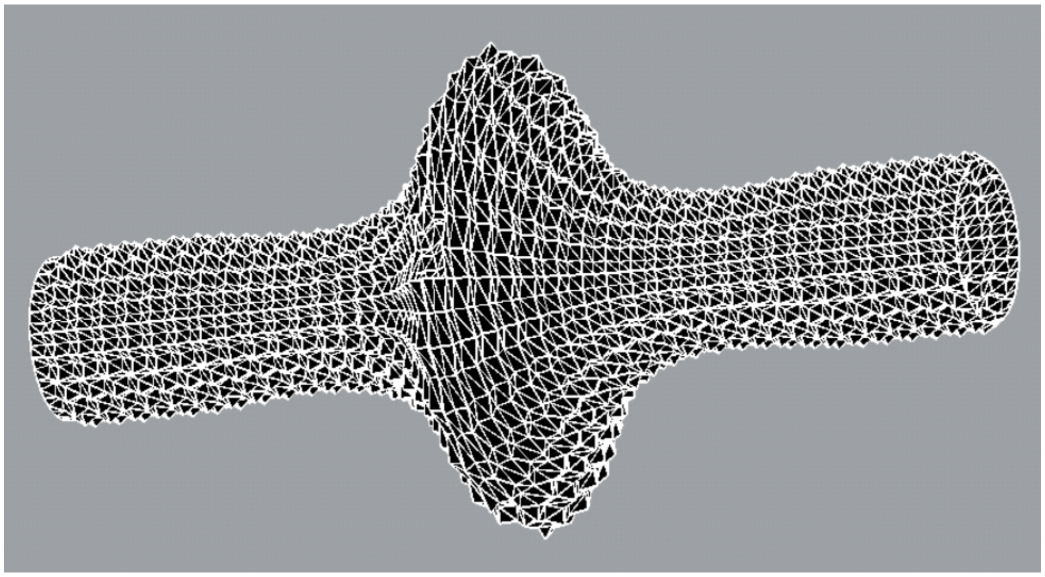
- $X_i^s$ on the right hand side are the positions yielded by (4.3) and (4.4). [Laplacian smoothing]
- $X_i^s$ on the left hand side are the positions yielded by underrelaxed Laplacian smoothing with $0 < w \leq 1$.

(4.5) slows down the smoothing in order to have a controlled surface deformation.

**Algorithm 5.** *[10] Volume conserving smoothing of a surface using edge relaxation. Using under relaxed Lapacian smoothing.*

- *Choose two nodes $X_1$ and $X_2$.*
- *Call the adjacent points for each as $\{X_i^{(j)}\}_{i=1,2}^{j=1,\dots,n_i}$.*
- $A_i = \sum_{j=1}^{n_i} e_i^{(j)} \times e_i^{(j+1)}, \quad i = 1, 2.$
- $v = e_2^{(n_2)} - e_2^{(2)}.$
- $X_1^s = \frac{1}{n_1 n_2 - 1}\left(\sum_{j=2}^{n_2} x_2^{(j)} + n_2 \sum_{j=2}^{n_1} x_1^{(j)}\right).$
- $X_2^s = \dfrac{1}{n_2}\left(X_1^s + \sum_{j=2}^{n_2} X_2^{(j)}\right).$
- $dX_i^s = w(X_i^s - X_i), \quad i = 1, 2, \quad 0 < w \leq 1.$
- $A = A_1 + A_2 + v \times (dX_1^s - dX_2^s).$
- *If $||A|| >$ ("a tiny number") then:*
  $\hat{\boldsymbol{n}} = A/||A||.$
  $h = -(dX_1^s \cdot A_1 + dX_2^s \cdot A_2 + dX_2^s \cdot v \times dX_1^s)/||A||.$
  $X_i = X_i + dX_i^s + h\hat{\boldsymbol{n}}, \quad i = 1, 2.$
- *Sweep until done.*

Figure 4.7 shows the result of applying algorithm 4 in smoothing a closed quadrilateral with 25 sweeps.



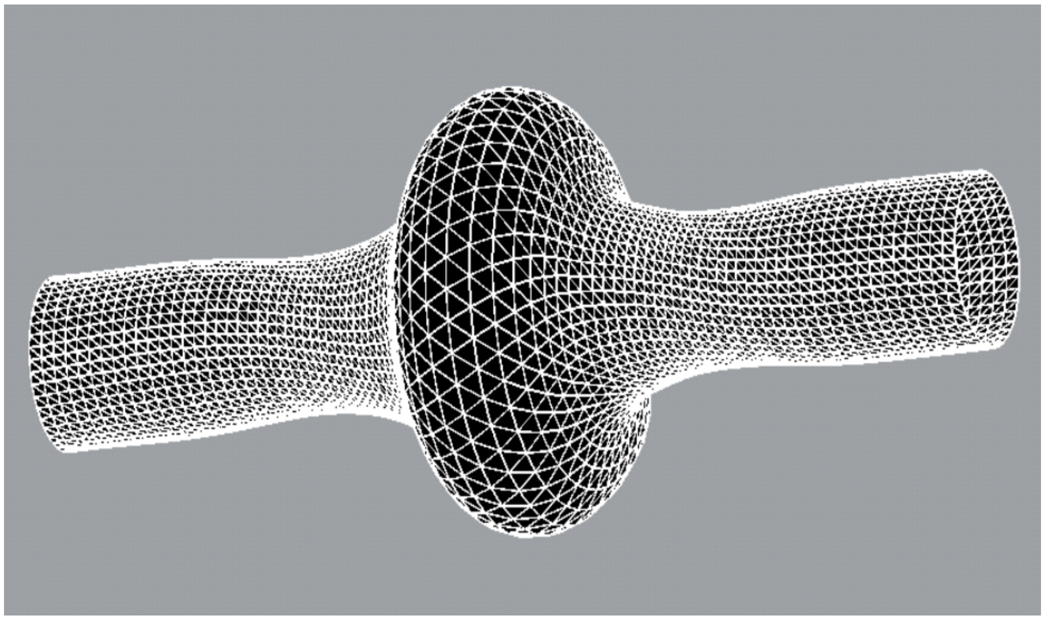
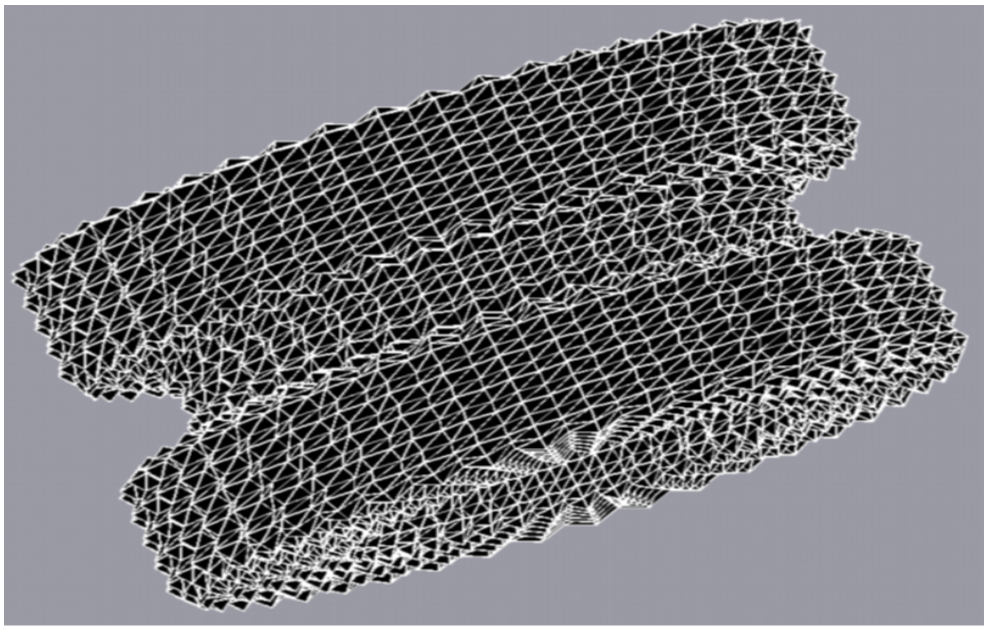


Figure 4.5: [10] Before and after smoothing an open triangular surface grid using 25 sweeps of algorithm 4.



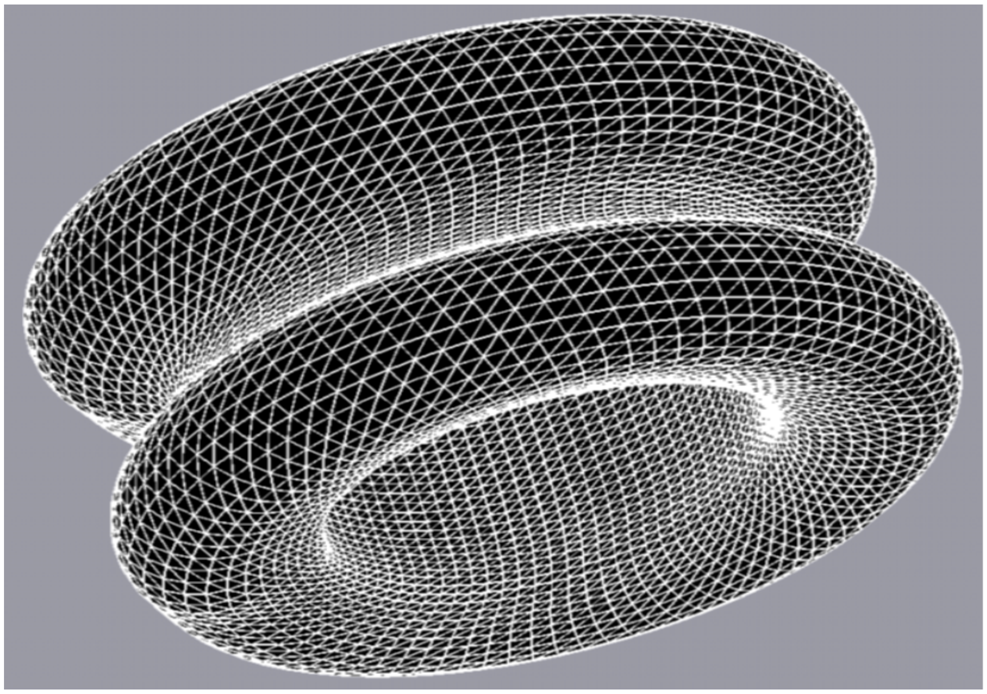
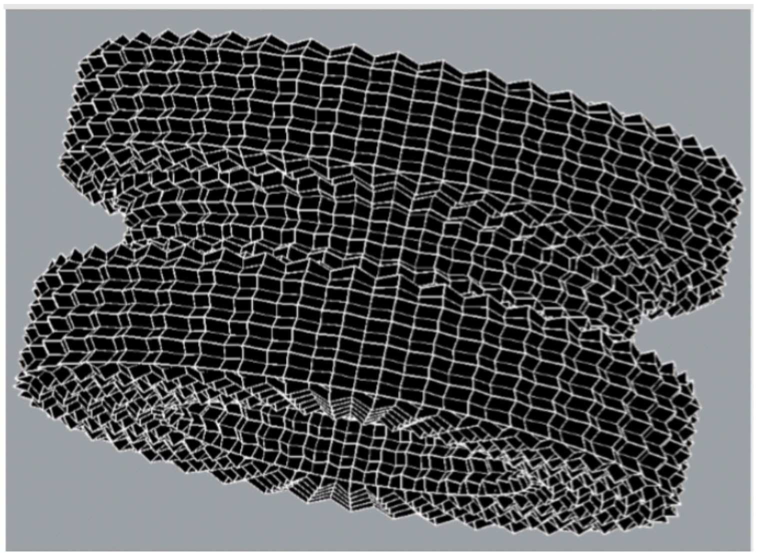


Figure 4.6: [10] Before and after smoothing an closed triangular surface grid using 25 sweeps of algorithm 4.



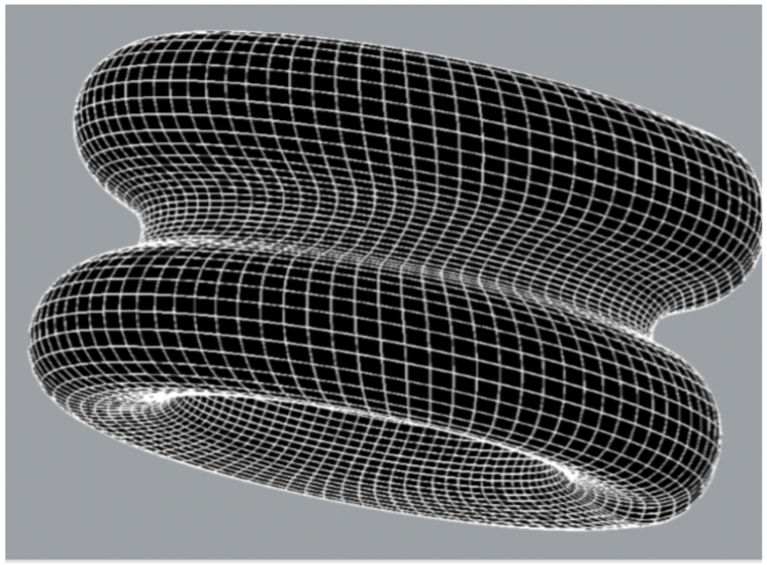


Figure 4.7: [10] Before and after smoothing a closed quadrilateral surface grid using 25 sweeps of algorithm 4.

# Chapter 5

# Mass conserving smoothing method

Smoothing mechanism has been used to solve a large number of applications, in this chapter we will use smoothing method to preserve mass in some applications like the incompressible fluid flow simulation with free surface. At the free surface, undulations appear on the surface because of the difference in a velocity from cell to cell, these undulations are usually smaller than a cell size and appear in computational simulations only since the tension and viscousity remove them.

This chapter illustrates a technique to conserve mass called Trapezoidal Sub-grid Undulations Removal (TSUR), this technique keeps a constant volume during the repositioning of vertices. TSUR adds robustness and linearity which are concerned to computational time. This chapter contains a review of some concepts, an illustration of a two different versions implemented by TSUR, planar and three dimensional.

## 5.1 Area of a triangle

Let $\mathbf{u}$ and $\mathbf{v}$ be two vectors in 3-D.
The parallelogram area (shaded area) equals to $|\mathbf{u}\times\mathbf{v}|$ where $|\mathbf{u}|$ is the base and $|\mathbf{v}||\sin\theta|$ is the height, i.e

$$\begin{aligned} Parallelogram\ area &= \mathbf{u}\times\mathbf{v} \\ &= \begin{vmatrix} i & j & k \\ u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \end{vmatrix} \\ &= i\begin{vmatrix} u_2 & u_3 \\ v_2 & v_3 \end{vmatrix} - j\begin{vmatrix} u_1 & u_3 \\ v_1 & v_3 \end{vmatrix} + k\begin{vmatrix} u_1 & u_2 \\ v_1 & v_2 \end{vmatrix} \\ &= (u_2v_3 - u_3v_2)i + (u_3v_1 - u_1v_3)j + (u_1v_2 - u_2v_1)k \end{aligned}$$

In the 2-D, Parallelogram area is the magnitude of $\mathbf{u} \times \mathbf{v}$, where:

$$\mathbf{u} \times \mathbf{v} = \begin{vmatrix} i & j & k \\ u_1 & u_2 & 0 \\ v_1 & v_2 & 0 \end{vmatrix} = (u_1 v_2 - u_2 v_1)k$$

$$\Rightarrow |\mathbf{u} \times \mathbf{v}| = (u_1 v_2 - u_2 v_1)$$

Now, the area of the triangle is one-half of parallelogram area as shown in Figure 5.1.

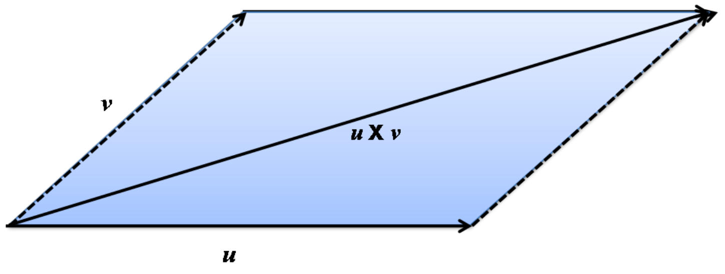


Figure 5.1: parallelogram area given by $\mathbf{u} \times \mathbf{v}$.

The cross product is a vector normal to the plane of the triangle, so the area of the triangle is:

$$A(T) = \frac{1}{2}(u_1 v_2 - u_2 v_1)$$

Substitute:

$$u_1 = x_j - x_i, \;\; u_2 = y_j - y_i$$

$$v_1 = x_k - x_i, \;\; v_2 = y_k - y_i$$

$$\Rightarrow A(T) = \frac{1}{2}\left[(x_j - x_i)(y_k - y_i) - (y_j - y_i)(x_k - x_i)\right]$$

This gives the area of triangle signed by the points $x_i$, $x_j$ and $x_k$ [3].

## 5.2 Trapezoidal sub-grid undulations removal in two dimensions

We will illustrate TSUR in 2-D, to see this consider four consecative points (particles in our application) $X_i$, $X_{i+1}$, $X_{i+2}$ and $X_{i+3}$ which forms quadrilateral as shown in Figure 5.2.

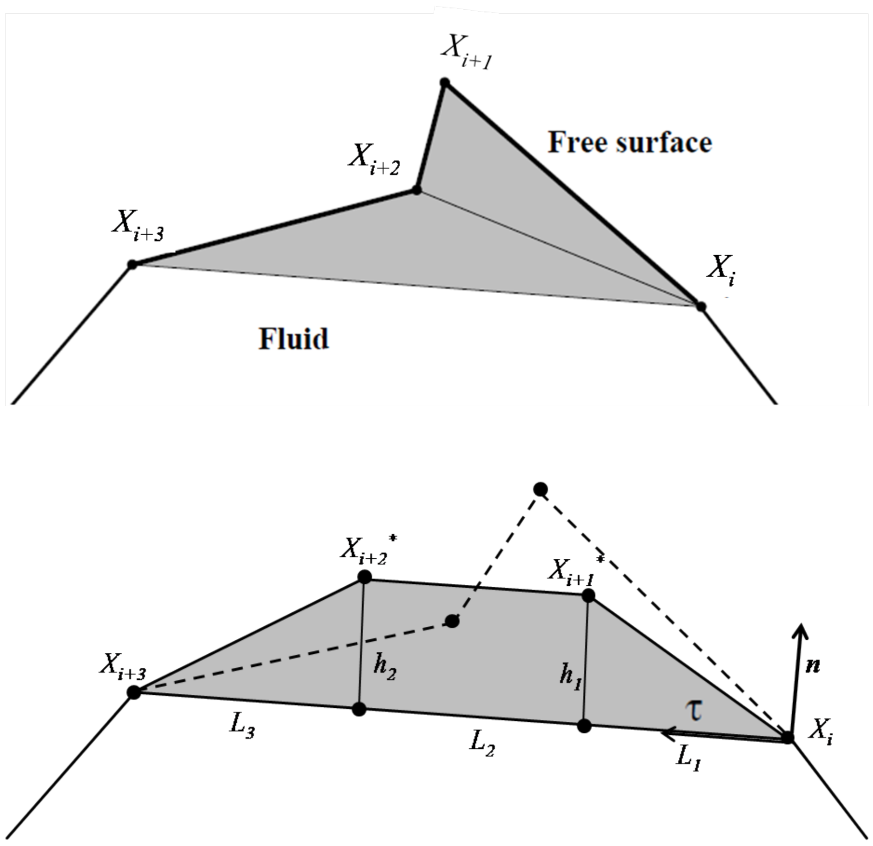


Figure 5.2: [12] Two-dimensional Trapezoidal Sub-gird Undulations Removal method.

We will reposite points $X_{i+1}$, $X_{i+2}$ such that the new shape (isosceles trapezium) has the same area shaded in Figure 5.2.

Also, $X_{i+1}$, $X_{i+2}$ moved to $X^*_{i+1}$ and $X^*_{i+2}$ respectively where $L_1 = L_2 = L_3 = L$ and $h_1 = h_2 = h$ as in Figure 5.2.

To compute quadilateral area in the two triangles where the area of the triangle defined by the points $X_i$, $X_j$ and $X_k$ taken a counter clockwise (we may take clockwise).

$$A_{ijk} = \frac{1}{2}\left[(x_j - x_i)(y_k - y_i) - (x_k - x_i)(y_j - y_i)\right]$$

where $X_i = (x_i, y_i)$.

To compute the area of trapezium, devide it into four triangles.

$$\begin{aligned} The\ total\ area &= A_1 + A_2 + A_3 + A_4 \\ &= 4A = 4(\frac{1}{2}lh) = 2lh \end{aligned}$$

And therefore $h = \frac{A}{2l}$.
The unit vector tangent to $X_{i+3} - X_i$ is:

$$\boldsymbol{\tau} = (\tau_x, \tau_y)^t = \frac{X_{i+3} - X_i}{||X_{i+3} - X_i||_2}$$

The outward unit normal is:

$$\mathbf{n} = (\tau_y, -\tau_x)^t$$

The new positions are:

$$\begin{aligned} X^*_{i+1} &= X_i + l\boldsymbol{\tau} + h\mathbf{n} \\ X^*_{i+2} &= X_i + 2l\boldsymbol{\tau} + h\mathbf{n} \end{aligned}$$

**Algorithm 6.** *[12]*

- *Perform smoothing operation on a neighborhood*
  $\{X_i, X_{i+1}, X_{i+2}, X_{i+3}\}, \quad i = 0, ..., n-1$
- $h = \frac{A}{2l}$
- $\boldsymbol{\tau} = \dfrac{X_{i+3} - X_i}{||X_{i+3} - X_i||_2}$
- $\boldsymbol{n} = (\tau_y, -\tau_x)^t$
- $X^*_{i+1} = X_i + l\boldsymbol{\tau} + h\boldsymbol{n}$
  $X^*_{i+2} = X_i + 2l\boldsymbol{\tau} + h\boldsymbol{n}$
- *Sweep until done.*

## 5.3 Trapezoidal sub-grid undulations removal in three dimensions

Consider Figure 5.3 which shows a vertex $V$ and it's corresponding star which is formed by the set of vertices $X_i$, $i = 0, 1, ..., n$, vertex $V$ is connected to the vertex $X_i$ by the edge $(V, X_i)$, $i = 0, 1, ..., n$ and vertices $X_i$ and $X_{i+1}$ are connected by the edge $(X_i, X_{i+1})$, $i = 0, 1, ..., n$, $(X_i, V, X_{i+1})$, $i = 0, 1, ..., n$ forms the faces.

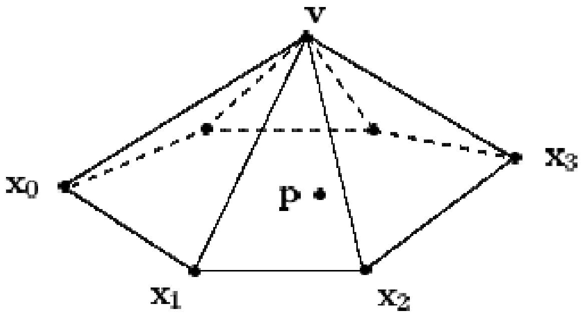


Figure 5.3: [12] Typical vertex and it's star.

TSUR in 3-D uses a balance procedure which preserves the local volume bounded by the star $(X_0, X_1, ..., X_n)$ and the faces $(X_i, V, X_{i+1})$, $i = 0, 1, ..., n$ or (alternatively) the volume bounded by the faces $(X_i, V, X_{i+1})$ and the faces $(X_i, P, X_{i+1})$, $i = 0, 1, ..., n$. Where $P$ is a point determined as in the following.

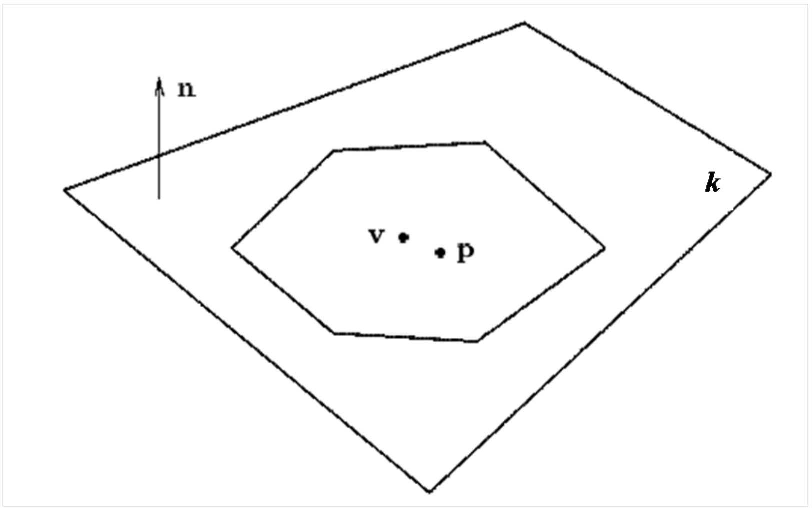


Figure 5.4: [12] The projection of the polyhedron into the plane K.

Figure 5.4 shows the projection of the polyhedron in Figure 5.3 into the plane $K$ which is defined by the point $P$ which is the average of the vertices (centroid of the star) $X_i$, $i = 0, 1, ..., n$,i.e

$$P = \frac{1}{n}\sum_{i=0}^{n} X_i \tag{5.1}$$

And the normal vector $\mathbf{n}$ which is computed by averaging the normal vectors of the faces $(X_i, P, X_{i+1})$, $i = 0, 1, ...n.$

$$\mathbf{n} = \frac{\sum_{i=0}^{n-1}(X_i - P) \times (X_{i+1} - P) + (X_n - P)(X_0 - P)}{||\sum_{i=0}^{n-1}(X_i - P) \times (X_{i+1} - P) + (X_n - P)(X_0 - P)||} \tag{5.2}$$

Now, each vertex(particle) of fluid will be moved depending on $P$ and $\mathbf{n}$ such that if $P$ is inside the projection of $X_0, X_1, ..., X_n$ on $K$, then the new position of $V$ is:

$$\begin{aligned} V_{new} &= P + h\mathbf{n} \\ h &= \frac{\mathbf{V}'}{\mathbf{V}_1} \end{aligned} \tag{5.3}$$

where:
$\mathbf{V}'$: The volume of the polyhedron $(V, X_0, X_1, ..., X_n, P)$.
$\mathbf{V}_1$:The volume of the unitary polyhedron $(P + \mathbf{n}, X_0, X_1, ..., X_n, P)$.

The volume of a polyhedron can be computed by the volume of a tetrahedron.

$$Volume = \frac{1}{6}\begin{vmatrix} x_1 - x_0 & x_2 - x_0 & x_3 - x_0 \\ y_1 - y_0 & y_2 - y_0 & y_3 - y_0 \\ z_1 - z_0 & z_2 - z_0 & z_3 - z_0 \end{vmatrix}$$

**Note 5.1.** *[12] $h = \frac{V'}{V_1}$ preserves the local volume since the volume of the polyhedron $(P + h\boldsymbol{n}, X_0, X_1, ..., X_n, P)$ is equal to $h$ times the volume of $(P + \boldsymbol{n}, X_0, X_1, ..., X_n, P)$.*

The previous step is called **vertex balance procedure** [12], Figure 5.5 shows this step.

Vertex balance procedure makes the mesh more homogeneous, but it doesn't remove the sub-gird undulation. So, another step called **Undulation removal procedure** is

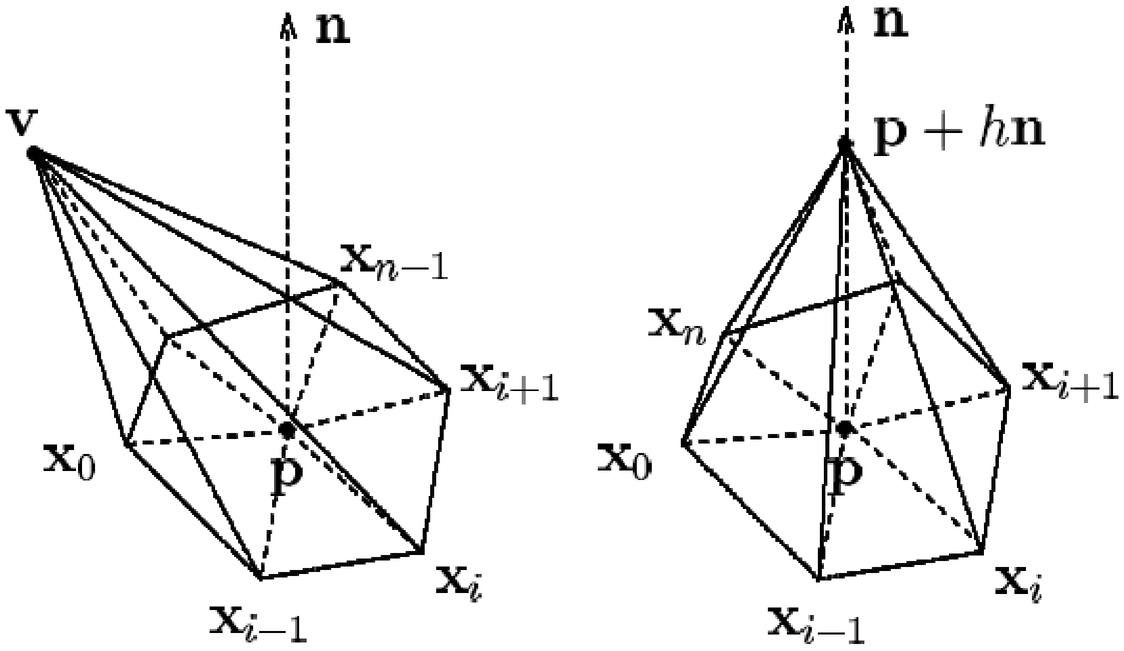


Figure 5.5: [12] A vertex with it's star, before and after the vertex balance procedure has been applied.

applied [12]. Although vertex balance procedure doesn't remove the undulation, it increases the robustness of the undulation removal procedure.

In Figure 5.6, Let $e$ be the edge connecting the vertices $V_1$ and $V_2$. Let $\mathbf{n}_1$ and $\mathbf{n}_2$ be the normals at $V_1$ and $V_2$ respectively computed as in (5.2).

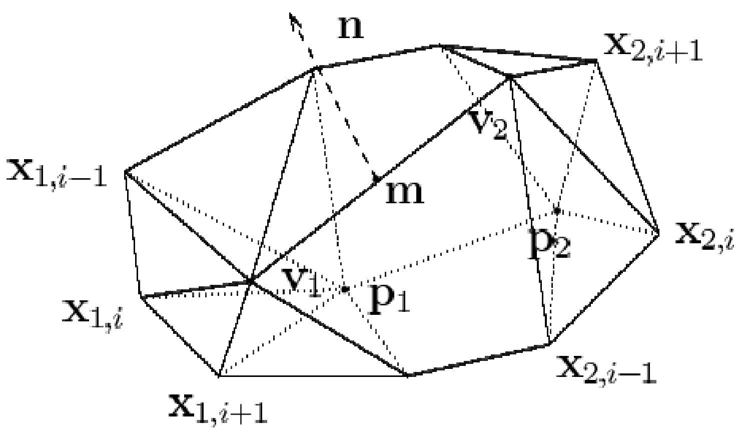


Figure 5.6: [12] Undulation removal procedure, showing an edge and it's star.

Then, the normal $n$ for the edge $e$ will be:

$$\mathbf{n} = \frac{\mathbf{n}_1 + \mathbf{n}_2}{||\mathbf{n}_1 + \mathbf{n}_2||}$$

Also, let $m$ be the average of the vertices $V_1$ and $V_2$ i.e,

$$m = \frac{1}{2}(V_1 + V_2)$$

The positions of vertices $V_1$ and $V_2$ will be changed as follows:

1. The height $h_1$ and $h_2$ corresponding to $V_1$ and $V_2$ respectively will be in the direction of $n$ and given by the inner product $h_1 =< V_1 - m, \mathbf{n} >$ and $h_2 =< V_2 - m, \mathbf{n} >$.

2. The points $p_1$ and $p_2$ are computed by:
   $p_1 = V_1 - h_1\mathbf{n}$ and $p_2 = V_2 - h_2\mathbf{n}$.
   The previous two steps gives two polyhedra $(V_1, X_{1,0}, X_{1,1}, ..., X_{1,n}, p_1)$ with volume $\mathbf{V}_1$ where $X_{1,0}, x_{1,1}, ..., x_{1,n}$ is the star of $V_1$ and $(V_2, X_{2,0}, X_{2,1}, ..., X_{2,m}, p_2)$ with volume $\mathbf{V}_2$ where $X_{2,0}, X_{2,1}, ..., X_{2,m}$ is the star of $V_2$.

3. We have to find the new height $h$ in the direction of $n$ such that the local volume is preserved.
   Let $\mathbf{V} = \mathbf{V}_1 + \mathbf{V}_2$, then by linearity (5.3):

   $$h = \frac{\mathbf{V}}{\mathbf{V}_a}$$

   Where: $\mathbf{V}_a$ is the volume of the polyhedron $(p_1, x_{1,0}, ..., x_{1,n}, p_1{+}n, p_2{+}n, x_{2,0}, ..., x_{2,n}, p_2)$
   So,

   $$h = \frac{\mathbf{V}_1 + \mathbf{V}_2}{\mathbf{V}_1|h_1 + \mathbf{V}_2|h_2}$$

4. The new positions of the vertices $V_1$ and $V_2$ are given by:
   $V_1 = p_1 + h\mathbf{n}$ and $V_2 = p_2 + h\mathbf{n}$.

5. The previous steps are applied to all edges of the mesh.

**Note 5.2.** *The number of times we apply the undulation removal procedure depend on the problem and the grid resolution.*

## Applications of TSUR

1. **Planer Flow**
   TSUR in 2-D is used to remove the undulation results in a free surface of a fluid during filling a container. Figure 5.7 shows the undulations in the dashed line (without TSUR in 2-D). While these undulations are removed after applying TSUR in 2-D which is represented by the dotted line [12].

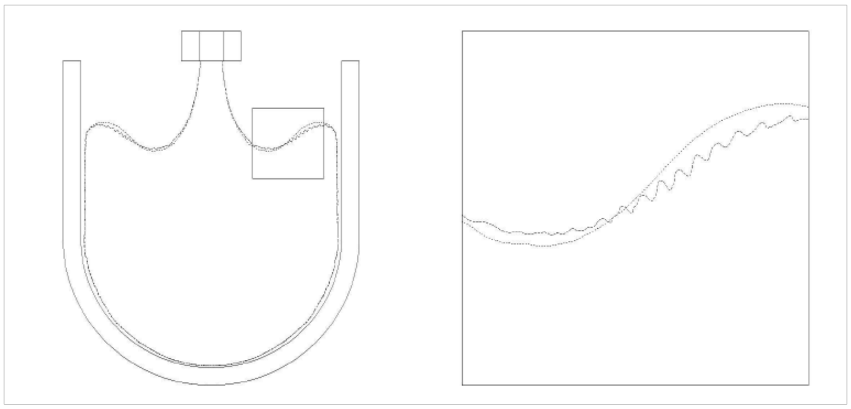


Figure 5.7: [12] Comparison of a simulation without TSUR against a simulation with TSUR.

2. **3-D Flow**
   TSUR in 3-D is used to remove the undulation appears on the pendant drop. The parameters of the drop taken are :$Re = 0.125$, $Fr = 0.25$ and $We = 0.02$. Figure 5.8 Shows the drop without applying TSUR in 3-D and Figure 5.9 shows the effect of TSUR in 3-D.

   As volume is conserved, the following table shows the volume after applying TSUR in 3-D several times, given error less than $10^{-3}$ [12].

| Number of Times | 0 | 15 | 45 |
|---|---|---|---|
| Volume | 0.504041595 | 0.504073084 | 0.50411593 |
| Figure | 5.a | 5.b | 5.c |

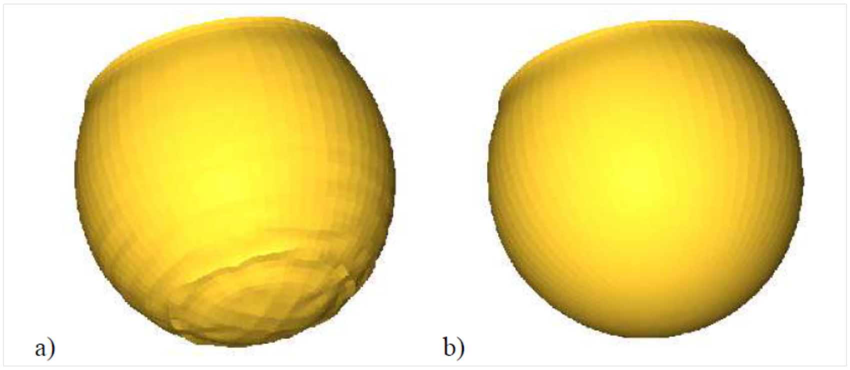


Figure 5.8: [12] Pendent drop; a) without TSUR, b) with TSUR.

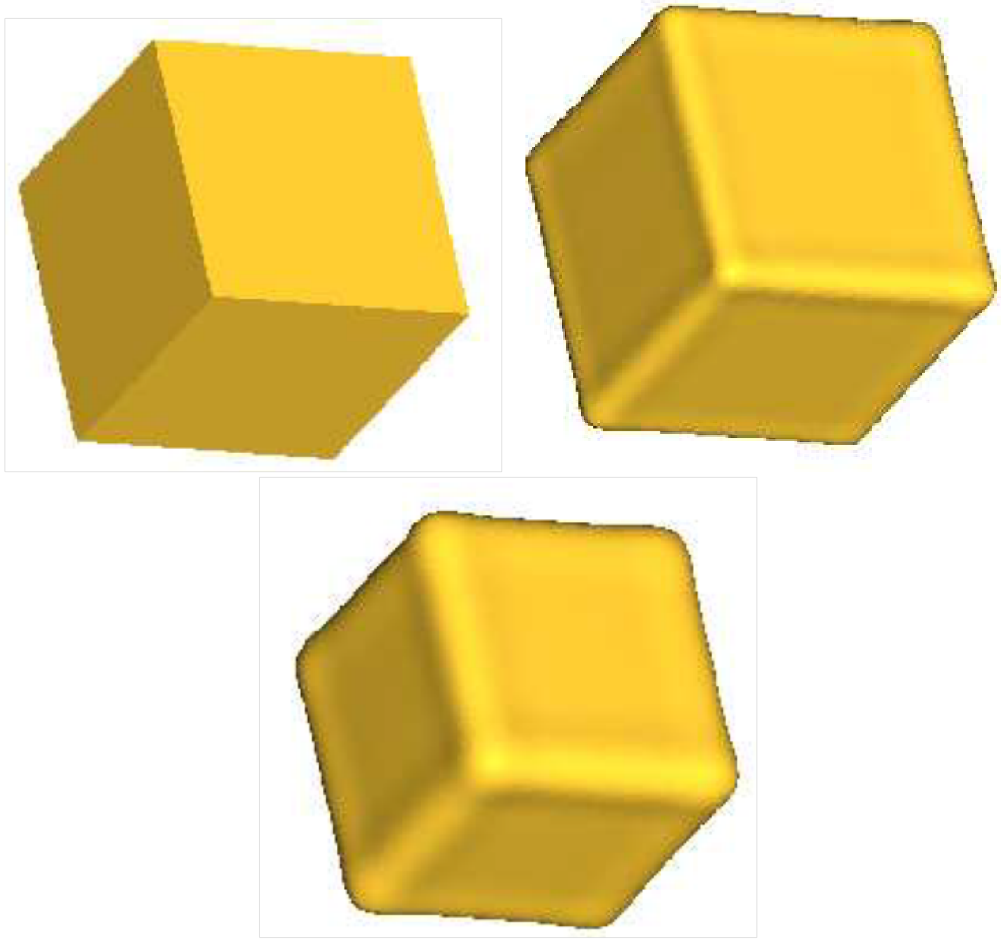


Figure 5.9: [12] TSUR-3D applied in a cube.

# Bibliography

[1] J.M.Aarts, *Plane and Solid Geometry,* Springer,LLC,2008.

[2] R. Burden, J. Douglas Fairs, *Numerical Analysis,* Ninth Edition, Brooks/ Cole Inc, 2010.

[3] M. Corral, *Vector Calculus,* 2008.

[4] M.Desbrun, M.Meyer, P.Schrder and A.Barr, *Implicit Fairing of Arbitrary Meshes Using Diffusion and Curvature Flow,* In Proceedings of SIGGRAPH 1999, PP.317324, 1999.

[5] I. Faux, M. Pratt, *Computational Geometry for Design and Manufacture: Mathematics and Applications,* Ellis Horwood Limited, 1985.

[6] T.Finney, *Calculus and Analytic Geometry,* Eleventh Edition, Wiley, 2008.

[7] D.Greenberg, L.Jordan and A.Gloag *CK-12's Basic Geometric Concepts,* 2012.

[8] A. Hajdu and I. Pitas, *Piecewise Linear Curves Representation and Compression using Graph Theory and a Line Segment Alphabet,* IEEE Transcation on Image Processing, 2008.

[9] Z.Ji, L.Liu and G.Wang, *A Global Laplacian Smoothing Approach with Feature Preservation,* IEEE Computer Society, 2005.

[10] A. Kuprat, A. Khamayseh, D. George and L. Larkey, *Non- Parametric Volume Conserving Smoothing,* J. Comput.Phys.,1-10, 1998.

[11] X.Liu, H.Bao, H.Y.Shum and Q.Peng, *A Novel Volume Constrained Smoothing Method for Meshes,* Graphical Models 64, 169-182, 2002.

[12] L.G. Nonato, N. Mangievacchi, F.Sousa, A. Castelo and J. Cuminato, *A Mass-Conserving Smooth Method,* Mecanica Computational Vol.XXIII, 1897-1909, 2004.

[13] J.O'Rourke, *Computational Geometry in C,* Second Edition,Cambridge University Press, 1997.

[14] G.Robinson, *Vector Geometry,* Dover Publication Inc, 1962.

[15] G.Taubin, *Dual Mesh Resampling,* Watson Research Center, 2001.

[16] G.Taubin, *Linear Anistrotropic Mesh Filtering,* IBM Research Report. Thomas J.Watson Research Center, 2001.