

# Documentation for 2D Surface Characterization and Fitting Algorithms

Yu Xiong

## 1 Introduction

This document provides an overview of the structure, mathematical concepts, and theory implemented in the fitting algorithm code which was written by using C++, which consists of the following files:

- `ReadParseData.h`
- `ReadParseData.cpp`
- `Characterize2DSurface.h`
- `Characterize2DSurface.cpp`
- `FittingAlgorithms.h`
- `FittingAlgorithms.cpp`
- `main.cpp`

The code is designed to read and parse 2D surface data from a file, fit the surface to various geometric shapes (e.g., doubly curved shell, singly curved shell, flat panel), and characterize the fitted surfaces using differential geometry concepts. The fitting algorithms are based on the least squares method, and the surface characterization is done using the first and second fundamental forms of the surface.

The purpose of this code is to provide a tool for analyzing and characterizing 2D surfaces within a mathematical and geometric framework. In some SEA cases, when the mesh geometry is imported into Hypermesh, the mesh is coarsely defined, and the surface is not smooth, making it difficult to extract accurate geometric properties. Accurate radius, angle, and surface area measurements are important for SEA analysis and thick shell modeling. This code offers a solution by fitting the surface to a geometric shape and characterizing it using differential geometry concepts.

The code is structured in a modular way to facilitate code reuse and extensibility.

## 2 Code Structure

Figure 1 illustrates the structure of the code and the relationships between different classes and files.

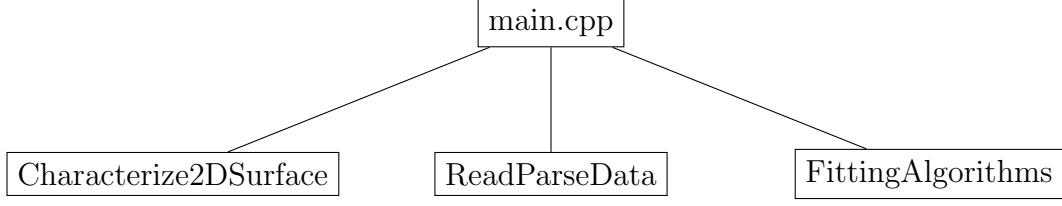


Figure 1: Code Structure Diagram

## 3 Mathematics and Theory

### 3.1 Least Squares Fitting

The least squares fitting method is used to find the best-fit curve or surface that minimizes the sum of the squared differences between the observed and predicted values. This method is fundamental in regression analysis.

For a set of points  $(x_i, y_i, z_i)$ , the goal is to fit a surface  $z = f(x, y)$ . This involves solving the system of equations:

$$A\mathbf{c} = \mathbf{z}$$

where  $A$  is the matrix of the polynomial terms,  $\mathbf{c}$  is the vector of coefficients, and  $\mathbf{z}$  is the vector of observed values. The solution is found using:

$$\mathbf{c} = (A^T A)^{-1} A^T \mathbf{z}$$

#### 3.1.1 Doubly Curved Shell (Elliptic Paraboloid)

A doubly curved shell is fitted using the general quadratic equation:

$$z = c + a_1x + a_2y + a_3x^2 + a_4xy + a_5y^2$$

The corresponding  $A$  matrix is:

$$A = \begin{pmatrix} 1 & x_1 & y_1 & x_1^2 & x_1y_1 & y_1^2 \\ 1 & x_2 & y_2 & x_2^2 & x_2y_2 & y_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & y_n & x_n^2 & x_ny_n & y_n^2 \end{pmatrix}$$

#### 3.1.2 Singly Curved Shell (Cylindrical Paraboloid)

A singly curved shell, which can be rotated along the x or y axis, is fitted using:

$$z = c_1(X - c_2)^2 + c_3Y + c_4$$

or

$$z = c_1(Y - c_2)^2 + c_3X + c_4$$

The corresponding  $A$  matrix for rotation along the y-axis is:

$$A = \begin{pmatrix} (X_1 - c_2)^2 & Y_1 & 1 \\ (X_2 - c_2)^2 & Y_2 & 1 \\ \vdots & \vdots & \vdots \\ (X_n - c_2)^2 & Y_n & 1 \end{pmatrix}$$

For rotation along the x-axis, the  $A$  matrix is:

$$A = \begin{pmatrix} (Y_1 - c_2)^2 & X_1 & 1 \\ (Y_2 - c_2)^2 & X_2 & 1 \\ \vdots & \vdots & \vdots \\ (Y_n - c_2)^2 & X_n & 1 \end{pmatrix}$$

### 3.1.3 Flat Panel

A flat panel is fitted using a simple linear equation:

$$z = a_0 + a_1x + a_2y$$

The corresponding  $A$  matrix is:

$$A = \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & y_n \end{pmatrix}$$

### 3.1.4 Closed-Form Singly Curved Shell (Cylinder)

A closed-form singly curved shell (cylinder) is fitted using:

$$(x - a)^2 + (y - b)^2 = R^2$$

The corresponding  $A$  matrix is:

$$A = \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & y_n \end{pmatrix}$$

and the vector  $\mathbf{b}$  is:

$$\mathbf{b} = \begin{pmatrix} -x_1^2 - y_1^2 \\ -x_2^2 - y_2^2 \\ \vdots \\ -x_n^2 - y_n^2 \end{pmatrix}$$

The solution for this fitting is obtained by solving:

$$\mathbf{c} = (A^T A)^{-1} A^T \mathbf{b}$$

### 3.1.5 Minimizing the Residual Sum of Squares (RSS)

The residual sum of squares (RSS) is given by:

$$RSS = \sum_{i=1}^n (z_i - f(x_i, y_i))^2$$

To minimize the RSS, we take the partial derivatives of the RSS with respect to each coefficient and set them to zero. This leads to a system of linear equations, which can be written in matrix form as:

$$A^T A \mathbf{c} = A^T \mathbf{z}$$

Solving this system gives the coefficients  $\mathbf{c}$  that minimize the RSS.

### 3.2 Differential Geometry for Surface Characterization

In differential geometry, the first and second fundamental forms of a surface are used to describe its curvature.

To compute these forms, we first need to find the partial derivatives of the surface function  $z = f(x, y)$ :

$$\begin{aligned} z_x &= \frac{\partial z}{\partial x} \\ z_y &= \frac{\partial z}{\partial y} \\ z_{xx} &= \frac{\partial^2 z}{\partial x^2} \\ z_{yy} &= \frac{\partial^2 z}{\partial y^2} \\ z_{xy} &= \frac{\partial^2 z}{\partial x \partial y} \end{aligned}$$

Using these derivatives, the first fundamental form is given by:

$$\begin{aligned} E &= 1 + z_x^2 \\ F &= z_x z_y \\ G &= 1 + z_y^2 \end{aligned}$$

The second fundamental form is:

$$\begin{aligned} L &= \frac{z_{xx}}{\sqrt{1 + z_x^2 + z_y^2}} \\ M &= \frac{z_{xy}}{\sqrt{1 + z_x^2 + z_y^2}} \\ N &= \frac{z_{yy}}{\sqrt{1 + z_x^2 + z_y^2}} \end{aligned}$$

The principal curvatures  $k_1$  and  $k_2$  are given by:

$$\begin{aligned} k_1 &= \frac{LG - 2FM + EN}{EG - F^2} \\ k_2 &= \frac{LG - EN}{EG - F^2} \end{aligned}$$

The radii of curvature are:

$$\begin{aligned} R_1 &= \frac{1}{|k_1|} \\ R_2 &= \frac{1}{|k_2|} \end{aligned}$$

The angles relative to the principal axes,  $\Theta_1$  and  $\Theta_2$ , are computed as:

$$\Theta_1 = \arccos \left( \sqrt{\frac{E}{E + G}} \right)$$

$$\Theta_2 = \arccos \left( \sqrt{\frac{G}{E+G}} \right)$$

### 3.2.1 Patch Surface Area Calculation

The surface area of a small patch on the surface is calculated using the formula:

$$\text{patch area} = \sqrt{EG - F^2} dX dY$$

where  $E$ ,  $F$ , and  $G$  are the coefficients from the first fundamental form, and  $dX$  and  $dY$  are the dimensions of the patch in the  $x$  and  $y$  directions, respectively.

This formula is implemented in the code to calculate the total surface area by summing up the areas of all small patches.

By using differential geometry theory to calculate the properties of the fitted surface, it computes the fundamental forms, curvatures, and other geometric characteristics by leveraging partial derivatives and curvature formulas derived from differential geometry. This approach provides a detailed and accurate description of the surface's local and global geometric properties.

## 4 Conclusion

This documentation provides an overview of the structure and mathematical concepts implemented in the provided C++ code for characterizing 2D surfaces and fitting them to various geometric shapes. The classes `ReadParseData`, `Characterize2DSurface`, and `FittingAlgorithms` are well-structured to handle reading data, performing surface fitting, and characterizing the fitted surfaces using differential geometry.

The code is designed to be modular, extensible, and reusable, making it easy to add new fitting algorithms or surface characterization methods in the future. By leveraging the least squares method and differential geometry theory, the code provides a powerful tool for analyzing and characterizing 2D surfaces in a mathematical and geometric framework.

## References

1. Manfredo P. do Carmo, *Differential Geometry of Curves and Surfaces*, Prentice Hall, 1976.
2. Barrett O'Neill, *Elementary Differential Geometry*, Academic Press, 2006.
3. Theodore Shifrin, *Differential Geometry: A First Course in Curves and Surfaces*, Online PDF.