



SURGE-2012



Project Report

An Algorithm for Fast and Automatic High Order Representation of Complex 3-D surfaces via Fourier Continuation Analysis

Submitted by:

Priya Goyal

Sophomore Undergraduate
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur

Under the guidance of:

Dr. Akash Anand
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur
Kanpur (U.P.)-208016, India

13th July, 2012



Summer Undergraduate Research Grant for Excellence, 2012 Indian Institute of Technology, Kanpur

An Algorithm for Fast and Automatic High Order Representation of Complex 3-D surfaces via Fourier Continuation Analysis

Priya Goyal
Sophomore Undergraduate
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur
Email: prigoyal@iitk.ac.in

Mentor:
Dr. Akash Anand
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur
Kanpur (U.P.)-208016, India

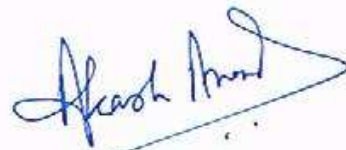
Abstract

Given a set of data points, creating the parameterizations of the full surface that represents this point cloud is a problem of great importance to engineers and scientists. Though there is software available for this purpose, the process requires a lot of human intervention particularly in the selection of intermediate projection surfaces. This makes the overall process of surface generation slow. This report presents an algorithm to automatically find the good projection surface and an efficient method of projecting the points orthogonally on it. This will make the process fast many times. The proposed algorithm can be applied to surfaces having some geometric singularities also, provided the necessary pre-processing of given data is done. For generating the projections surfaces automatically, we interpolate the given point cloud by quadratic surfaces and proper surface of revolution. We use Fourier Continuation method to find a smooth function for a discrete set of data points representing a smooth function. We demonstrate our approach by randomly generating a point cloud for a known surface and then show our implementation results on the fuselage and engines of Falcon aircraft for which the point cloud was given/generated using softwares like computer aided design (CAD).

CERTIFICATE

This is to certify that the project entitled "An Algorithm for fast and automatic high order parametrisation of complex 3-D surfaces via Fourier Continuation Analysis" submitted by Miss Priya Goyal as a part of Summer Undergraduate Research Grant for Excellence, 2012, offered by the Indian Institute of Technology, Kanpur, is a bonafide record of the work done by her under my guidance and supervision at the Indian Institute of Technology, Kanpur, from May 08, 2012 to July 12, 2012.

13th July, 2012



Signature of Supervisor
Dr. Akash Anand
Assistant Professor
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur

Candidate's Declaration

I hereby declare that this project report titled "**An Algorithm for fast and automatic high order parameterization of complex 3-D surfaces via Fourier Continuation Analysis**" submitted towards the completion of SURGE 2012 Program at IIT Kanpur is an authentic record of my work carried out under the guidance of "**Dr. Akash Anand**" at IIT Kanpur.

I affirm that the entire report has been written by me and any material taken from other sources has been appropriately acknowledged.

13th July, 2012



Student's signature

Name of the Student PRIYA GOYAL

ACKNOWLEDGEMENT

Working in SURGE 2012 was a great experience for me. I am very grateful to my mentor “**Dr. Akash Anand**” for his guidance and help throughout this project. The insightful discussions and motivation given by him made my first step into scientific career exciting.

In addition, I would like to thank Dr. Anurag Gupta, Mrs. Aparna Mitra and all the other members of SURGE 2012 organizing team for their care and help. SURGE was a true learning experience and fun filled program and I feel lucky to be a part of it.

Priya Goyal
Sophomore Undergraduate
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur

TABLE OF CONTENTS

CONTENT	PAGE NUMBER
Abstract.....	2
Certificate	3
Candidate's Declaration	4
Acknowledgement	5
1. Introduction	7
1.1 Brief Overview	7
2. Fourier Continuation Method.....	7
2.1 Method of Least Squares	8
2.2 Fourier Continuation Method for two dimensional functions	9
3. Parameterization by projection and continuation	9
Demonstration of Algorithm	10
4. Algorithm for finding the projection curve/surface automatically.....	12
1-Dimensional Functions	12
4.1 Why interpolate by Quadratic curves?	12
4.2 Single value Decomposition Method	12
4.3 Algorithm for parameterizing the quadratic equations.....	12
4.4 How to project the point cloud on projection curve orthogonally	13
Implementation results of above algorithm (for 1-Dimensional functions)	13
2-Dimensional Functions	15
Implementation results of our approach for 2-Dimensional Functions	16
5. Implementation results of our approach on Falcon aircraft	20
6. Conclusion and Future Work.....	25
6.1 Summary and Concluding Remarks	25
6.2 Future Directions	25
References	25

1 Introduction

This report presents an algorithm for high-order parametric representations of 3-D surfaces efficiently by making the intermediate process of selecting a good projection surface automatic. Until quite recently, the construction of surfaces involved manual work. The article [1] published gives a way to render full surface parameterizations for the given point cloud (thousands of points) for complex surfaces of engineering and scientific relevance. It “...renders both smooth and non-smooth portions of surface accurately, up to and including all points of geometric singularity like corners, edges, conical points etc...” [1] However, this method and software designed for this purpose require intensive human intervention, particularly, in the intermediate process of selection of projection surfaces, which makes the overall process of surface generation very slow.

“The problem of producing high-order parameterizations of surfaces is of great importance in wide range of field of science and engineering” which include medical imaging etc. [1]. In this report, we present an **automatic** and **efficient** method for creation of high-order surface representations for a given point cloud. We use the concept of Fourier Continuation [1] to find a smooth function that will represent a given point cloud (“...which should represent the *smooth portions* of a piecewise smooth function...” [1]) best, by defining a periodic function on larger domain. The point cloud for the engineering surfaces is either provided or can be extracted using a computer aided design (CAD) system. This point cloud, however, need to be pre-processed in order to satisfy the requirements of algorithm.

1.1 Brief Overview

This report has been organized in the following manner:

- Section 2 briefly explains the Fourier Continuation method which gives smooth functions for the point cloud representing a smooth portion of a surface. For this, we use Method of Least-squares which has been discussed briefly in Section 2.1.
- Section 3 presents an overall algorithm followed to find the surface parameterization for the given point cloud. The process has been demonstrated by randomly generating the point cloud for a known function of 1-Dimension.
- Section 4 presents our approach for finding the good projection surface automatically. An algorithm for finding the parametric form of this projection surface has been given in sub-section 4.3. Sub-section 4.4 gives an efficient algorithm for projecting the point cloud orthogonally on the automatically generated projection surface. This is followed by our implementation results for 1-Dimensional functions and 2-Dimensional function
- Section 5 shows the results of implementation of our approach on the fuselage and engines of a Falcon aircraft.

We have covered all conclusions and future directions in Section 6.

2 The Fourier Continuation Method

According to the article [1] “...this method enables creation of rapidly convergent Fourier series, from discrete point values of functions that are *smooth* but not necessarily periodic.” It gives certain least-squares interpolation by trigonometric polynomials (also called as truncated/finite Fourier series which is periodic with period b and the periodicity interval is greater than the domain of point cloud)

$$F(x) = \sum_{k=-m}^m c_k e^{\frac{i2\pi kx}{b}} \quad \text{where } m \leq N$$

of the point values (x_i, y_i) where $i = 1, 2, \dots, N$. Here c_k 's are the unknown Fourier modes to be determined and the total number of Fourier modes $(2m + 1)$ should be less than N (the size of point cloud) i.e. we solve the oversampled system of equations in order to determine c_k 's. Since the N data points may/are not equally spaced, so we use the Method of Least Squares discussed briefly in Section 2.1 [2] (Chapter 5, Section 5.3, page-236) to find them. For example, consider the function $f(x) = x$, defined in the interval $0 \leq x \leq 1$. Figure 2.1 shows the randomly generated set of data points in this interval

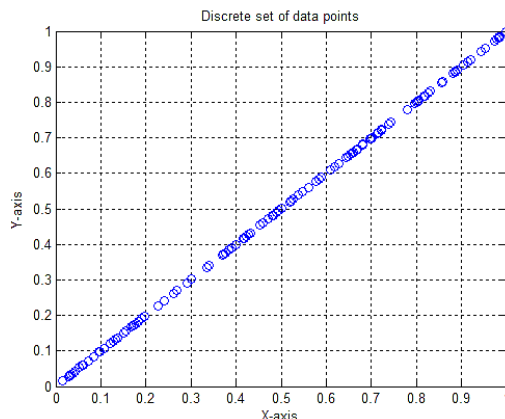


Figure 2.1: Randomly generated set of data points (here $N=128$) for the function $y=x$ in the interval $x \in [0,1]$

We want to interpolate this set of data points (representing a smooth curve) by truncated Fourier series. Clearly $f(x) = x$ is not a periodic function but the interpolating polynomial is periodic with period b . We observe that the truncated Fourier series of $f(x) = x$ oscillates around the points of discontinuity (also known as **Gibb's phenomenon**) in the periodic extension of f with period 1 (Figure 2.2). "To overcome this problem, the continuation method seeks a Fourier series of periodicity larger than the domain of definition of $f(x)$ " [1]. In our case, we choose the enlarged periodicity interval to be $0 \leq x \leq 2$. Figure 2.3 shows the enlarged periodicity interpolation.

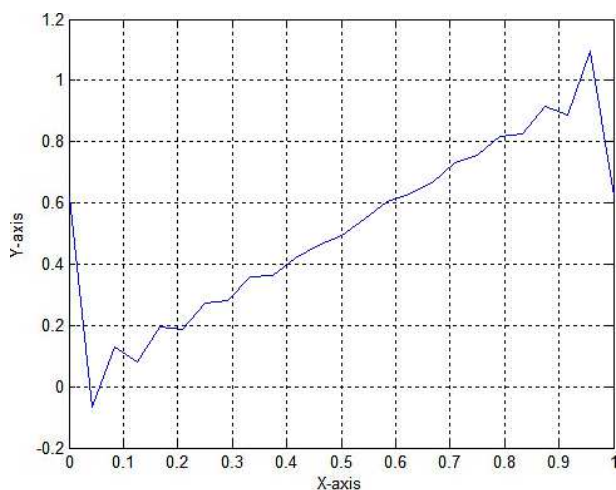


Figure 2.2: Oscillating finite Fourier series of $f(x) = x$ around the points of discontinuity 0 and 1.

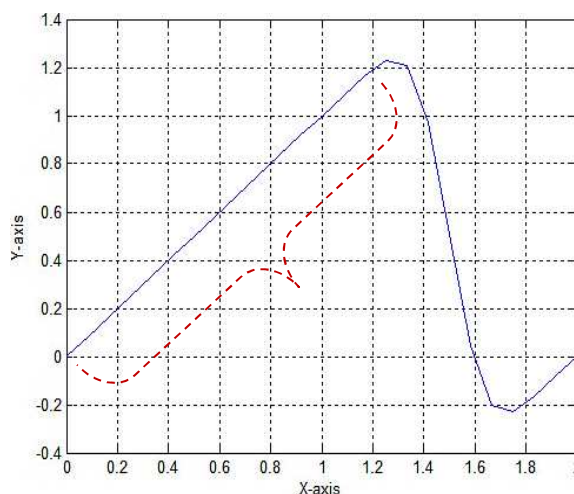


Figure 2.3: Interpolation of $f(x) = x$ with enlarged periodicity finite Fourier series.

2.1 Method of Least Squares

It is a standard method of finding the best fit curve to a given set of oversampled data points by minimizing the squares of offsets of points from their interpolating function. Suppose we are given N data points (x_i, y_i) where $i = 1, 2, \dots, N$ and an interpolating function f (in one-dimension)

$$f(x, a_1, a_2, \dots, a_m) \text{ Where } m \leq N$$

Here a_1, a_2, \dots, a_m are unknown's. To determine them, we minimize the square of offsets i.e.

$$\epsilon^2 = \sum_{i=1}^n [y_i - f(x_i, a_1, a_2, \dots, a_m)]^2 \dots \dots \dots (2.1.1)$$

We now differentiate (2.1.1) partially w.r.t. a_1, a_2, \dots, a_m each to get a system of m equations to solve say $Ax=b$. Here the unknown coefficients are given by vector x . The vector x is called the least- square solution. Given an inconsistent system of equations, we want to find x from R^m so that the error $\|b - Ax\|$ is minimum possible. This vector x is given by

$$x = (AA^T)^{-1}A^T b$$

2.2 Fourier Continuation Method for 2-Dimensional functions

Just like in 1-Dimensional case, finite/truncated Fourier series i.e. the interpolation function in 2-Dimension is given by

$$F(x, y) = \sum_{k=-m}^m \sum_{l=-n}^l c_k e^{\frac{i2\pi kx}{b_1}} c_l e^{\frac{i2\pi ly}{b_2}} \text{ where } m, n \leq N$$

the periodicity b_1 and b_2 are selected according to the domain of x and y respectively. Similarly, we can extend this approach to the higher dimension functions.

3 Parameterization by projection and continuation

In order to obtain the surface parameterization for the given point cloud, we follow the following algorithm given below and demonstrate it through an example:

- Extract the points from the given point cloud that represent a smooth surface. (e.g. See figure 3.1)
- For the extracted point cloud, project the points on the projection surface (we give an algorithm for finding the good projection surface automatically in Section 3.1).
- Find the point of orthogonal projection on projection surface for each point in the selected point cloud (Section 3.1.4).
- Then, we find the directed distance of each point from the point of projection using the following equation

$$d(t_j) = [(x_j, y_j) - P(t_j)] \cdot \overline{n(t_j)} \dots \dots \dots (3.1)$$

Where, $P(t_j)$ is the parametric form of the projection surface with parameter t , $\overline{n(t_j)}$ is the unit normal vector on the projection curve at the projection point t_j .

NOTE: For 2-Dimensional functions, equation (3.1) is

$$d(u_j, v_j) = [(x_j, y_j, z_j) - P(u_j, v_j)] \cdot \overline{n(u_j, v_j)}$$

- Then we find the smooth distance function using Fourier continuation method for the discrete values obtained from equation 3.1.
- So, now we have smooth distance function and a smooth projection surface/curve to determine the surface parameterization for the selected point cloud, given by:

$$(X(t), Y(t)) = P(t) + d(t) \cdot \overline{n(t)} \dots \dots \dots (3.2)$$

NOTE: Similarly, for 2-Dimensional functions, equation 3.2 is

$$(X(u, v), Y(u, v), Z(u, v)) = P(u, v) + d(u, v) \cdot \overline{n(u, v)}$$

This whole process can be understood with the following example:

STEP 1:

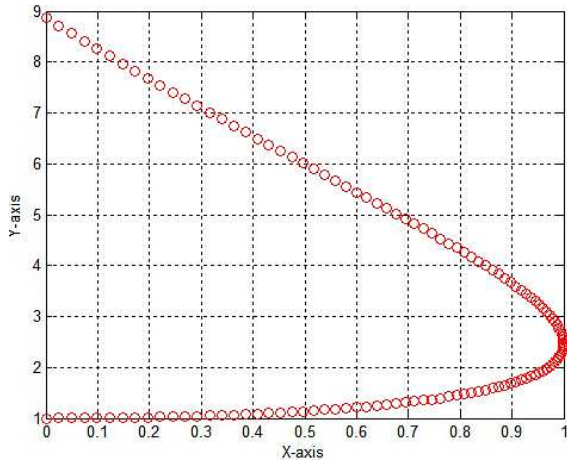


Figure 3.1: Randomly generated point cloud in interval $[0, 1]$ for the curve given by the parametric equations $x(t) = \sin(t)$, $y(t) = \cos(t) + t^2$

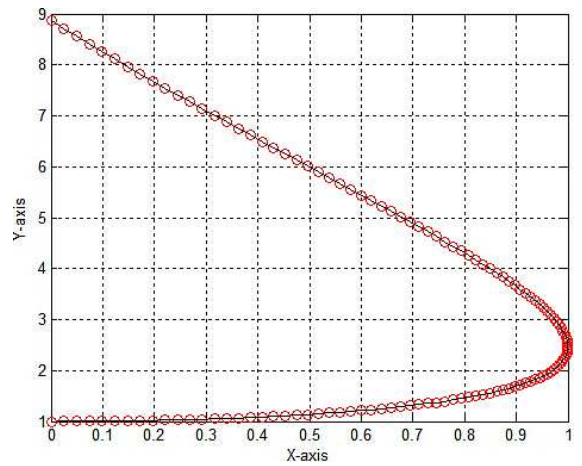


Figure 3.2: Desired curve (black line) which represents the randomly generated point equations cloud (red) best.

STEP 2:

We now choose a projection curve for this point cloud (for the sake of demonstration only, though, we can automatically generate a good projection curve for a point cloud using the algorithm in Section 4) so that the resulting distance function is continuous. (Figure 3.3 shows the point cloud, projection curve and the projections)

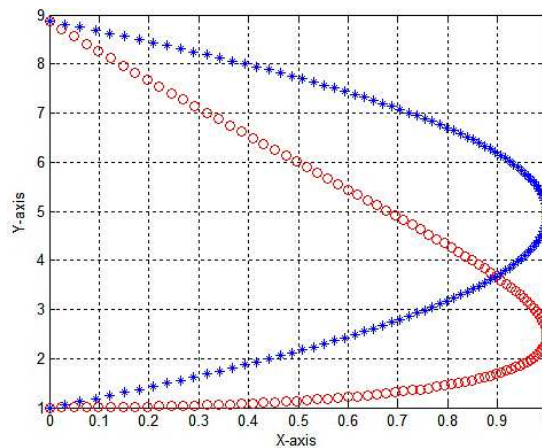


Figure 3.3: Original point cloud (red) and the selected projection curve given by the parametric equations $X(t) = \frac{4}{\pi^2} \cdot t \cdot (\pi - t)$, $Y(t) = \left(\frac{\pi^2 - 2}{\pi}\right) \cdot t + 1$

STEP 3:

Now we project the given set of data points on this point cloud orthogonally. In order to project the points, we solve the equation $[x_j - X(t_j)] \cdot X'(t_j) + [y_j - Y(t_j)] \cdot Y'(t_j) = 0$ which is the dot product of tangent and normal to the projection curve at point $X(t_j)$ and $Y(t_j)$ and since this equation may have trigonometric, exponential, logarithmic etc. terms depending on the projection curve selected (when we generate the projection curves/surfaces automatically, we can actually convert the equation, we solve to find projection points, to a quadratic, cubic, quartic or octic equations for which we know the solutions explicitly [2]), we use **Newton-Raphson method** to find the roots. Since, Newton-Raphson method is an iterative process, it takes a long to find project even a single point, so this approach is very inefficient. Figure 3.4(a) shows the skewed view of projection and Figure 3.4(b) shows the actual view.

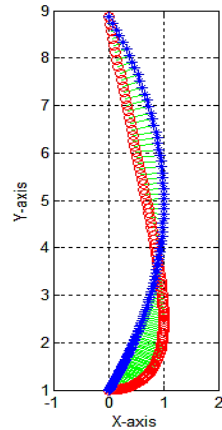
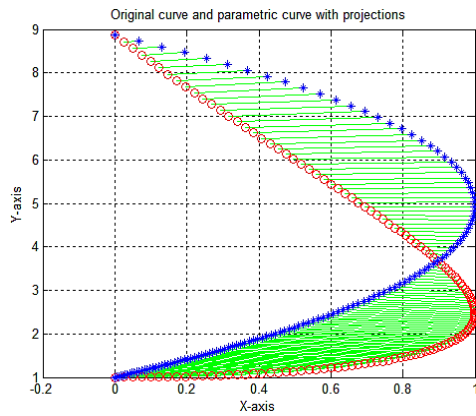


Figure 3.4(a) and 3.4(b): Skewed view and the actual view resp. of the projections (green) of the point cloud (red) on the projection curve (blue) selected.

STEP 4:

We now find the directed distance of each point in the point cloud from the point of orthogonal projection (See Figure 3.5).

STEP 5:

We then find the smooth distance function using Fourier Continuation Method (Section 2). Figure 3.6 shows the smooth distance function in the domain of the data. The curve is smooth in the domain but it can be bad outside it which we needn't worry as we are concerned with the data in the domain only.

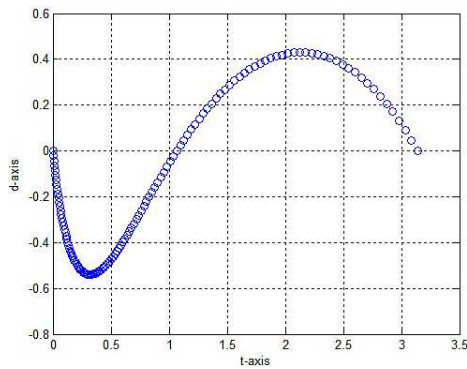


Figure 3.5: Directed distance plot of the points in the point cloud from their orthogonal point of projection.

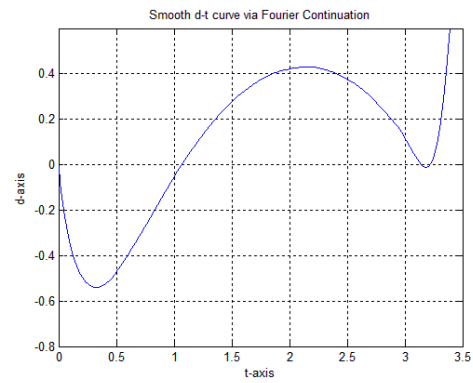


Figure 3.6: Smooth distance function (in the domain) obtained using the Fourier Continuation method.

STEP 6:

We are ready to find the parameterization of the smooth curve for the original point cloud using equation 3.2. (See Figure 3.7)

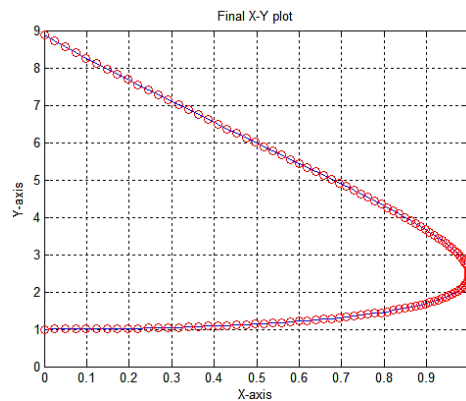


Figure 3.7: Smooth surface (blue) obtained for the given point cloud (red) using equation 3.2

4 Algorithm for finding the projection curve/surface automatically

The projection surface should be such that the resulting directed distance plot is continuous to ensure the smooth surface/curve parameterization for the point cloud. Such a projection curve/surface need not be unique but our algorithm gives a good projection curve *automatically*.

- **1-Dimensional functions**

In order to find the projection curve for the point cloud of 1-Dimensional functions, we interpolate the given point cloud by quadratic curves given by:

$$Ax^2 + 2Bxy + Cy^2 + Dx + Ey + F = 0 \dots\dots\dots (4.1)$$

Here A, B, C, D, E and F are unknown coefficients and we determine them using the method of least squares (Section 2.1). In order to solve the resulting homogeneous system of equations say $Gt = 0$, where $t \in N(G)$, We use the *Single Value Decomposition (SVD)* method [3 Chapter 2, Section 2.6, Page-69]. A brief description of SVD method is given in Section 4.2. After finding the coefficients, we parameterize the equation 4.1 using the algorithm explained in Section 4.3.

4.1 Why interpolate by Quadratic curves?

Interpolating by quadratic curves has advantages:

1. We get a smooth projection curve for the given point cloud so that the final curve parameterization is also smooth.
2. To find the projection points of the points in point cloud, we actually have to solve a cubic equation or quartic equation (Section 4.4) and we explicitly know the solutions for them [2].

4.2 Single Value Decomposition method

Given any $m \times n$ matrix A , we can write it as:

$$A = UDV^T \text{ where,}$$

- U is $m \times n$ orthogonal matrix with column vectors as Eigen vectors of AA^T and can be written as $(u_1 \ u_2 \ \dots \ \dots \ \dots \ \dots \ u_n)$.
- D is $n \times n$ diagonal matrix with non-zero singular values σ_i such that $\sigma_1 \geq \sigma_2 \geq \dots \dots \dots \geq \sigma_r$ where r is the rank of matrix A .
- V is $n \times n$ orthogonal matrix with column vectors as Eigen vectors of $A^T A$ and can be written as $(v_1 \ v_2 \ \dots \ \dots \ \dots \ \dots \ v_n)$.

Using the SVD method, we find the least square solution to the system of homogeneous equations $Ax=0$ as follows:

Clearly, $x \in N(A)$ and we have an oversampled system of equations to solve, so $m \geq n$.

Suppose $rank(A) = n-1$, where, $m \geq n - 1$, $\sigma_n = 0$, then the solution is $x=v_n$ as $\dim N(A) \neq 0$.

Suppose $rank(A) = n$, then there are only trivial solution to the system of equations as now $\dim N(A) = 0$, but we choose the solution $x=v_n$.

4.3 Algorithm for parameterizing the quadratic equations

In order to parameterize the quadratic equation :

$$Ax^2 + 2Bxy + Cy^2 + Dx + Ey + F = 0 \dots\dots\dots (4.1)$$

- Write the equation 3.1.1 in the matrix form as:

$$\begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} A & B \\ B & C \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} D & E \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + F = 0$$

- Now, diagonalise the matrix $M = \begin{pmatrix} A & B \\ B & C \end{pmatrix}$, say $M = S\beta S^{-1}$ where $\beta = \begin{pmatrix} \gamma_1 & 0 \\ 0 & \gamma_2 \end{pmatrix}$, and transform into new coordinate system X-Y where $S^{-1} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} X \\ Y \end{pmatrix}$. Let the equation in new coordinate system be:

$$aX^2 + bY^2 + dX + eY + f = 0 \dots\dots\dots (4.2)$$

- Now, by applying completing the squares approach on equation 4.2 if $a \neq 0$ & $b \neq 0$ or directly parameterizing it if either of a or b is 0, the above equation will represent a parabola, ellipse or a hyperbola. We then parameterize the resulting equation depending on the quadratic curve it represents. This gives a smooth parametric form of projection curve.

4.4 How to project the point cloud on the projection curve orthogonally

To project the points in the point cloud, we follow the following approach:

- We find the point cloud representation in the new coordinate system $X - Y$ using the value of X and Y in terms of x and y $\left(S^{-1} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} X \\ Y \end{pmatrix} \right)$.
- Let (X_i, Y_i) be a point in the point cloud in the new coordinate system. If (X, Y) is the orthogonal projection of the point on the projection curve and $\nabla F(X, Y)$ is the gradient of curve at point (X, Y) , then the following equation holds:

$$\left((X_i - X), (Y_i - Y) \right) = \alpha \nabla F(X, Y) \dots\dots\dots (4.3)$$

Now there are three unknowns $(X, Y$ and $\alpha)$ and we also know the parametric representation of X and Y , so essentially, we have two unknowns (parameter (say t) and α). As we said in Section 4.1, we eliminate the variable α by dividing the two equations in 4.3 and convert the resulting equation into a cubic or quartic equation in $\sin(\theta)$ or $\sin(2\theta)$ (in case of hyperbola and ellipse respectively) or quadratic equation in case of parabola. We know the explicit solutions to quadratic, cubic and quartic equations [2]. This gives us the possible projections of a point on the projection curve. We then select the point which is closest to the point in the point cloud.

Implementation results of above algorithm (for 1-Dimensional functions):

STEP 1:

We generated the point cloud (N=260) for the curve represented by the parametric equations (See Figure 4.1):

$$\begin{aligned} x(t) &= \cos t + 0.65 \cos 2t - 0.65 \\ y(t) &= 1.5 \sin t \end{aligned}$$

Our goal is to find the parameterization of the curve that will represent this point cloud best. Now our only input to the algorithm will be the original point cloud.

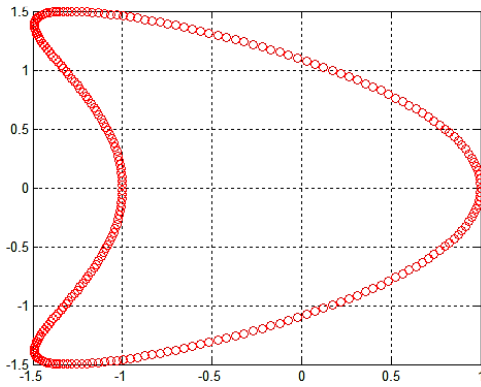


Figure 4.1: Randomly generated point using the Parametric equations above

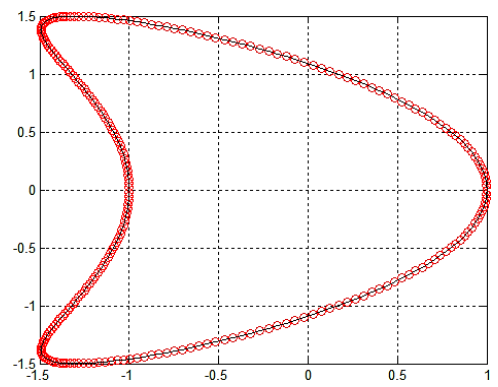


Figure 4.2: Desired curve (black) for the point cloud generated (red)

STEP 2:

We need a projection surface to project the points. Our algorithm implementation generated an ellipse as a good projection surface (See Figure 4.3). In order to project the points on the curve, we have to solve a fourth degree equation (Section 4.4) for which we know the solutions explicitly [2]. This approach is very much fast and efficient.

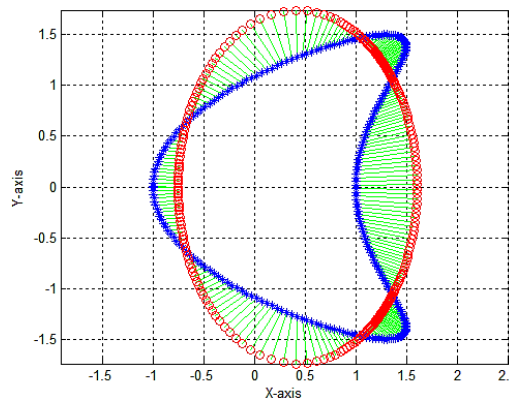


Figure 4.3: Randomly generated point cloud (blue) and the projection surface (red) with projection (green) in the *new Coordinate system*

STEP 3:

We now find the directed distance and find a smooth distance function in the domain of data using Fourier Continuation Method (See Figure 4.4 and 4.5)

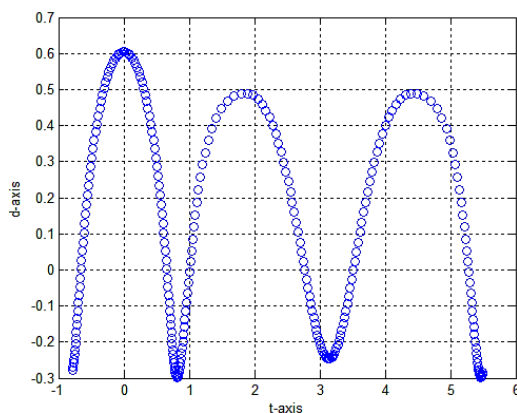


Figure 4.4: Directed distance of the points in The point cloud from their point of projection

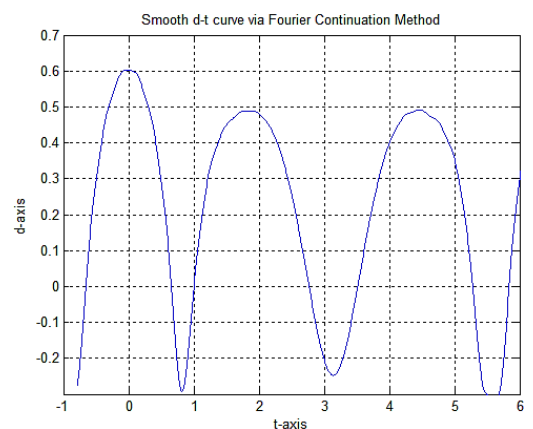


Figure 4.5: Smooth distance function in the domain obtained via Fourier Continuation method.

STEP 4:

Now, find the final smooth curve parameterization for the point cloud using equation 3.2. (See Figure 4.6)

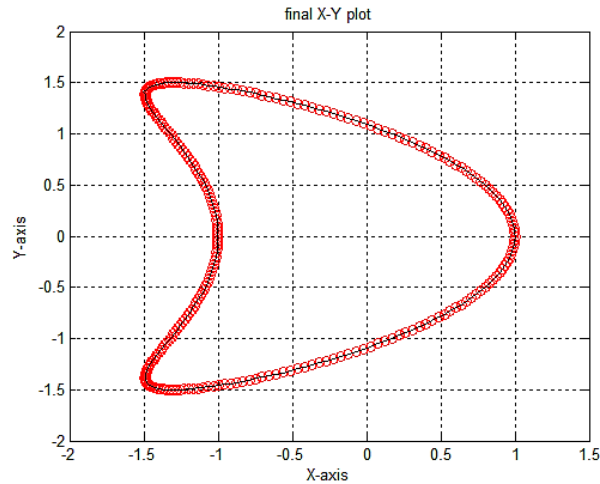


FIGURE 4.6: Final smooth curve (black) parameterization obtained For the point cloud (red)

The only input to our algorithm is the randomly generated point cloud. The overall process is very efficient. Manually selecting the projection curves [1] consumes a lot of time but our approach took less than 5 sec. (in this case) to give the final surface parameterization for the point cloud.

- **2-Dimensional Functions**

To find the projection curve for 2-Dimensional functions, we interpolate by the quadratic curves given by the following equation:

$$Ax^2 + By^2 + Cz^2 + 2Dxy + 2Eyz + 2Fxz + Gx + Hy + Iz + J = 0$$

Now, to parameterize this equation, the matrix M in the algorithm given in Section 4.3 changes to:

$$M = \begin{pmatrix} A & D & E \\ D & B & F \\ E & F & C \end{pmatrix}$$

After getting the parametric form of projection surface, we project the points on this projection surface by solving the following equation(s):

$$((X_i - X), (Y_i - Y), (Z_i - Z)) = \alpha \nabla F(X, Y, Z) \dots\dots\dots(4.4)$$

Suppose u and v are the parameters in the parametric form of ellipsoid, hyperboloid and paraboloid. In case of ellipsoid, by eliminating the variable α , we get two equations (in $\sin(u)$ and $\sin(v)$) to solve. These equations have the following form:

$$(1 - p^2)(1 - q^2)(a_1p^2 + b_1p + c_1) = d_1p^2 \dots\dots\dots(4.5)$$

$$q^2(1 - p^2)(a_2p^2 + b_2p + c_2) = d_2p^2 \dots\dots\dots(4.6)$$

where $p = \sin(u)$ & $q = \sin(v)$ and $a_1, a_2, b_1, b_2, c_1, c_2, d_1$ & d_2 are the constants. We can now eliminate q using the equations 4.5 and 4.6 to get an octic equation in variable p . To explicitly find roots of the octic equation, we propose to write it as a product of two quartic equations and then we know how to solve quartic equations [2] (we haven't yet found a way to factorize the octic equation and used MATLAB's inbuilt solve function for our implementations). After finding the values of p , we find the corresponding values of q and then the general roots of $\sin(u) = p$ and $\sin(v) = q$. Then, select the root which gives the closest point to the point in point cloud on the projection curve. Similarly, in case of hyperboloid, for one sheet hyperboloid, we get an octic equation in terms of $\tan(u)$ and $\tan(v)$ and for a two-sheet hyperboloid; we get an octic equation in terms of $\sec(u)$ and $\sec(v)$ [NOTE: For the parameterization of hyperboloid projection curve, we are using \tan and \sec instead of $\cosh()$ and $\sinh()$ as we then don't need to worry about which sheet of

hyperboloid should we select for the projections]. In case of paraboloid projection curve, the equations 4.5 and 4.6 are the following:

$$a_1u^3 + b_1u^2 + (c_1v^2 + d_1v + e_1)u + f_1v^2 + g_1v + h_1 = 0 \dots\dots\dots(4.7)$$

$$a_2v^3 + b_2v^2 + (c_2u^2 + d_2u + e_2)v + f_2u^2 + g_2u + h_2 = 0 \dots\dots\dots(4.8)$$

In order to solve these equations to find the roots, we propose to solve eq. 4.7 (using the technique described in [2]) for u treating v as a constant and substitute the resulting values of u (in terms of v), to find the values of v and u later.

Implementation results of our approach for 2-Dimensional functions:

We generated a point cloud randomly for the surface (Figure 4.7) represented by the equation:

$$\frac{x^2}{a^2(1-\alpha_3 \cos(\pi z))} + \frac{(\alpha_1 \cos(\pi z)+y)^2}{b^2(1-\alpha_2 \cos(\pi z))} + \frac{z^2}{c^2} = 1 \dots\dots\dots(4.9)$$

where a=0.8 =b, c=1, $\alpha_1 = 0.3$, $\alpha_2 = 0.4$, $\alpha_3 = 0.1$

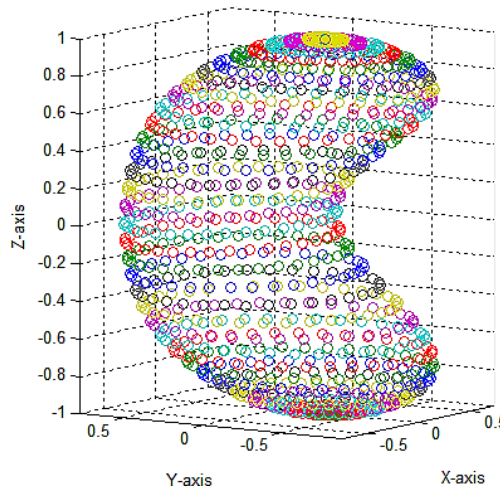


Figure 4.7: Randomly generated point cloud on the surface

We aim at finding the parameterization of the surface for the above point cloud. The following figures show the steps followed and the results we obtained using our approach.

As we said in the beginning, we need to extract the proper point cloud, which will represent a smooth surface, out of the original point cloud in order to satisfy the high-order application. A given point cloud may represent a surface which has many singularities. For such point clouds, some pre-processing is necessary.

For the above given set of data, we extracted the following point clouds and got the following results [The following figures depict the following: (a) shows the point cloud extracted (b) shows the point of orthogonal projection (red) for each point in the point cloud (c) shows the corresponding directed distance smooth surface obtained via Fourier Continuation Analysis and (d) shows the parameterizations of the surface (smooth) obtained for the extracted point cloud]

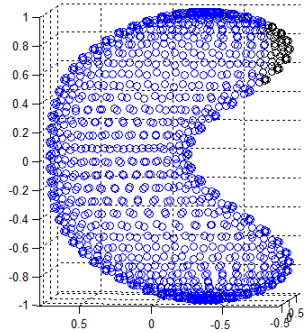


Figure 4.8(a)

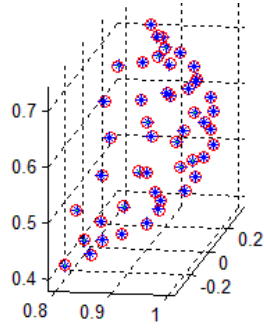


Figure 4.8(b)

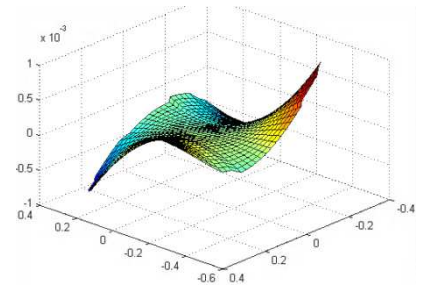


Figure 4.8(c)

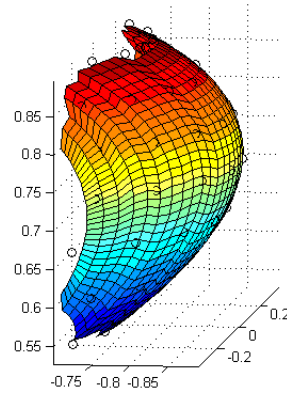


Figure 4.8(d)

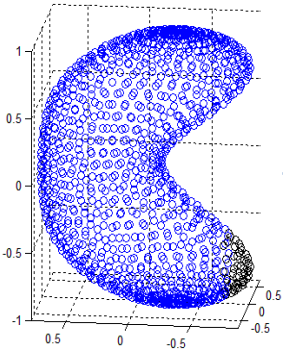


Figure 4.9(a)

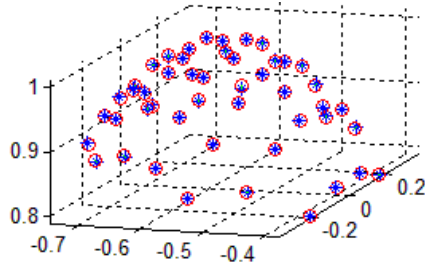


Figure 4.9(b)

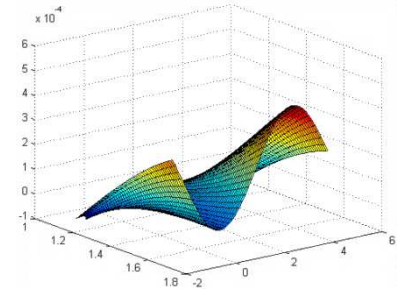


Figure 4.9(c)

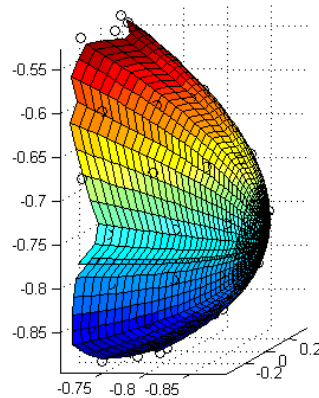


Figure 4.9(d)

The above point clouds were extracted because of the singular points on the surface.

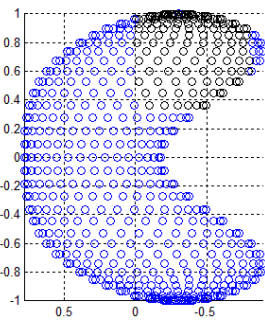


Figure 4.10(a)

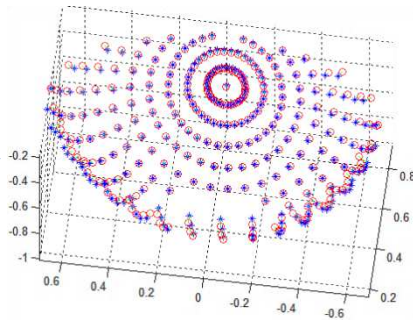


Figure 4.10(b)

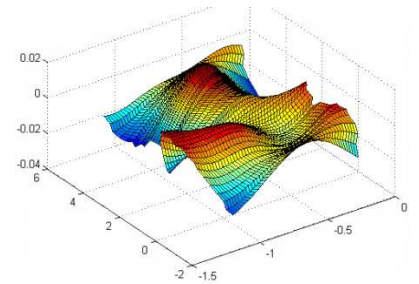


Figure 4.10(c)

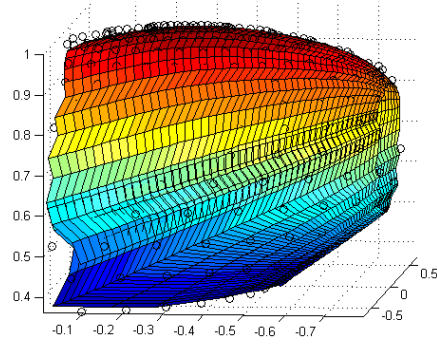


Figure 4.10(d)

Another singular point on the surface which doesn't will not let the distance surface to be continuous if not extracted out. A similar singular point occurs on lower side also.

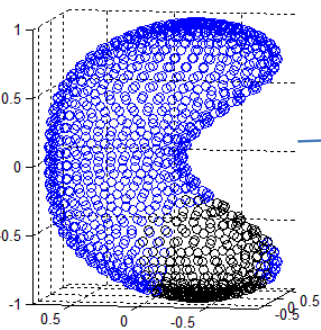


Figure 4.11(a)

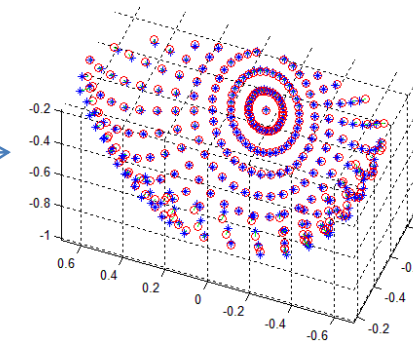


Figure 4.11(b)

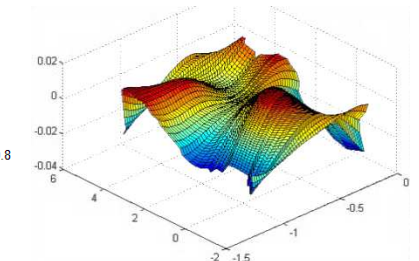


Figure 4.11(c)

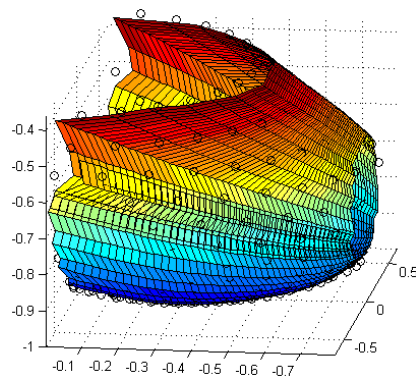


Figure 4.11(d)

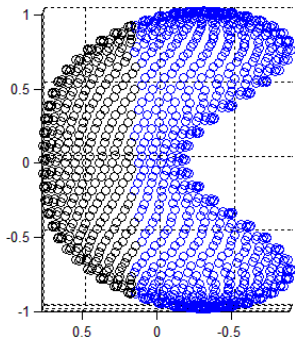


Figure 4.12(a)

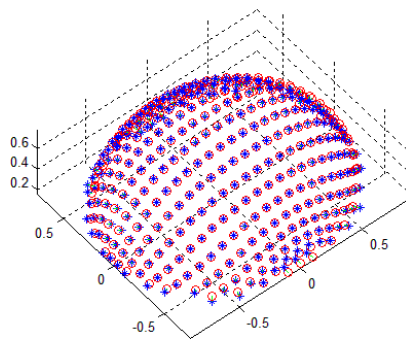


Figure 4.12(b)

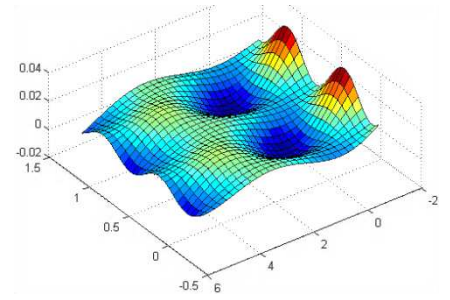


Figure 4.12(c)

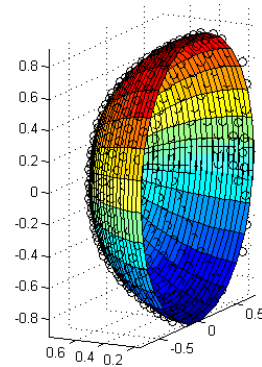


Figure 4.12(d)

[NOTE: The point clouds extracted must have some data in common so that when we finally plot the surfaces corresponding to each point cloud together, we get a continuous surface.]

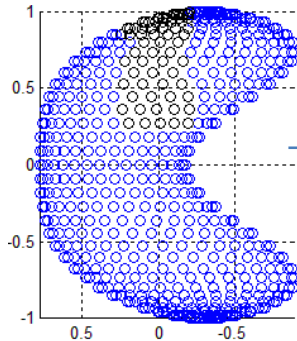


Figure 4.13(a)

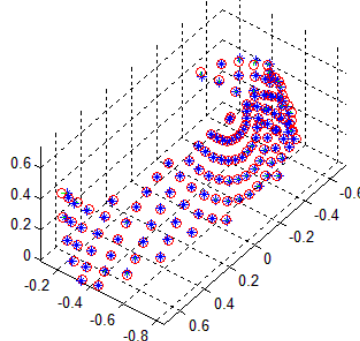


Figure 4.13(b)

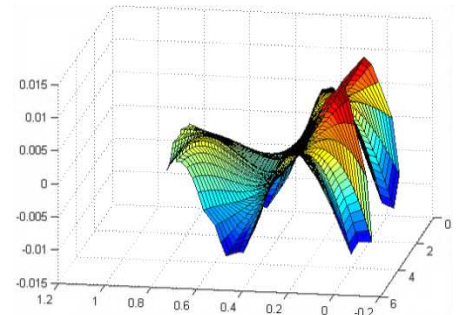


Figure 4.13(c)

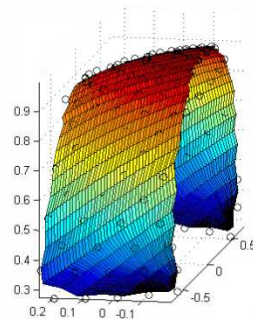


Figure 4.13(d)

This point cloud is extracted so that we can make smooth connection between the singular point surface and the surface in Figure 4.12(d).

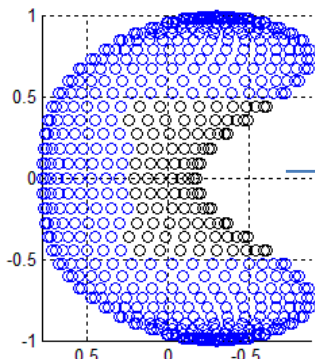


Figure 4.14(a)

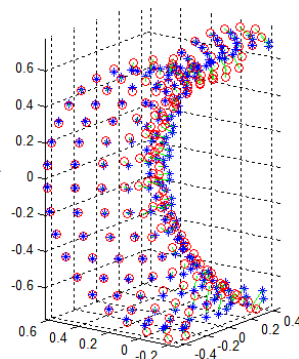


Figure 4.14(b)

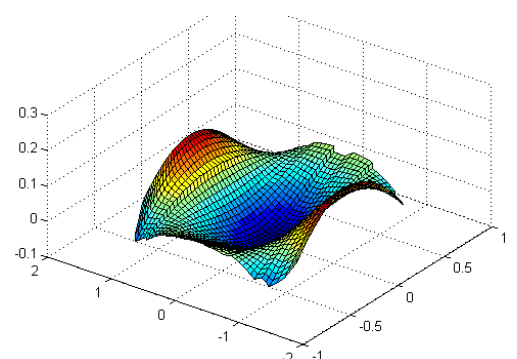


Figure 4.14(c)

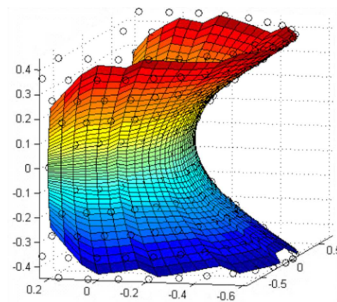


Figure 4.14(d)

This front section was extracted out because of the singular point at the top and bottom of the Surface and this act as the connector between two and the back surface. We always keep some point cloud extra at the edges so that we can have a continuous surface in the end. After plotting the surfaces corresponding to each extracted point cloud together, we obtained a final surface representation for the complete original point cloud as shown in Figure 4.15

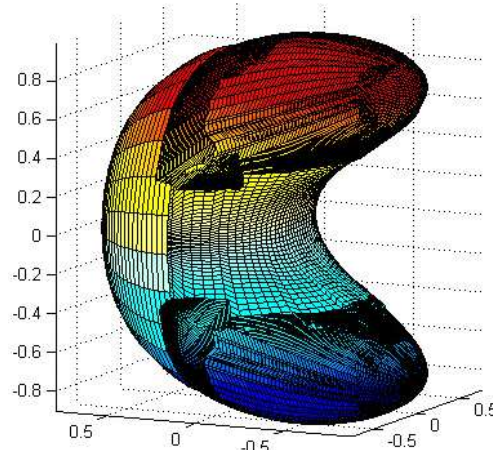


Figure 4.15

When we plot all the surfaces together, the overlap at the boundaries may not be complete [1]. There is some error for which blending of surfaces is done and has not been focused in this report.

5 Implementation results of our approach on Falcon Aircraft

We aim at creating the full surface high-order surface parameterizations for this point cloud automatically and efficiently. In this report, we show the surface parameterizations we obtained for the engines and the fuselage of the plane.

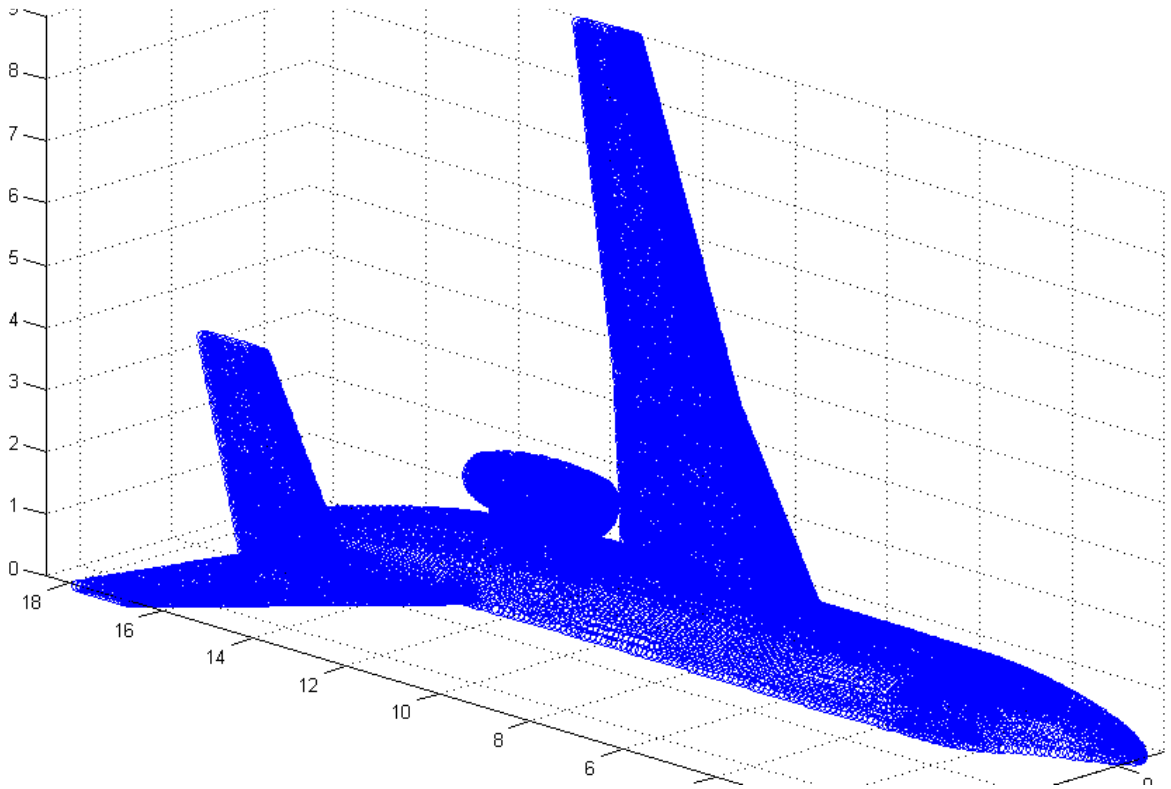


Figure 5.1: Original point cloud of Falcon Aircraft

Similar to the previous demonstration, we pre-processed the point cloud of Falcon aircraft and following are the results we obtained:

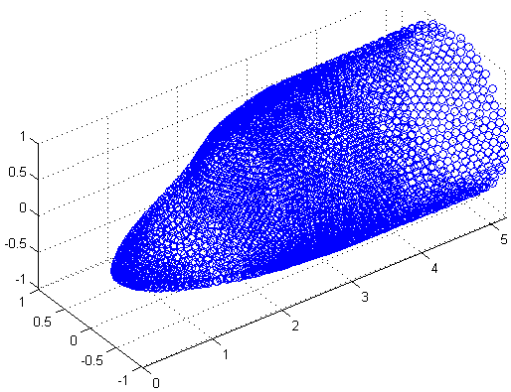


Figure 5.2(a)

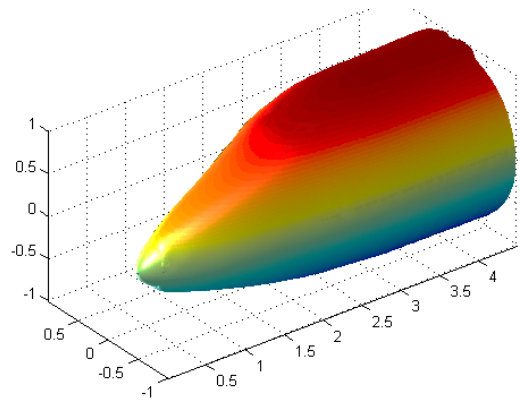


Figure 5.2(b)

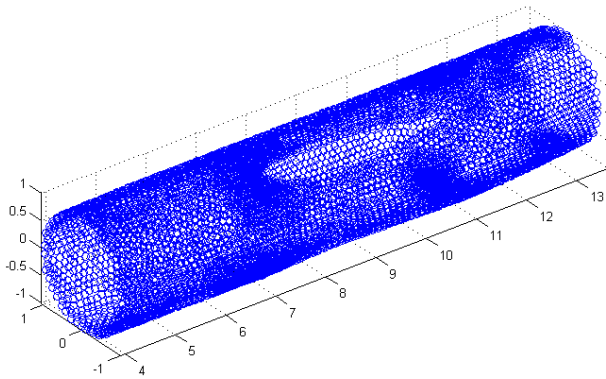


Figure 5.3(a)

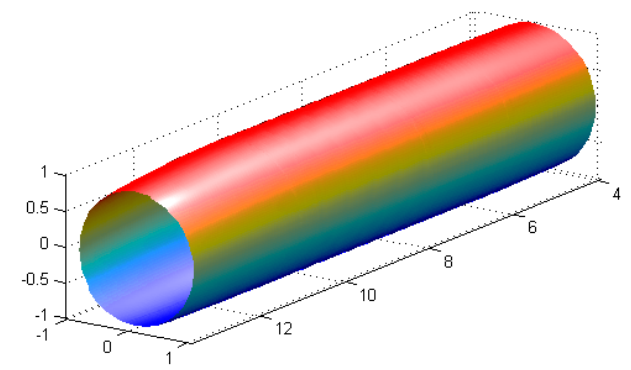


Figure 5.3(b)

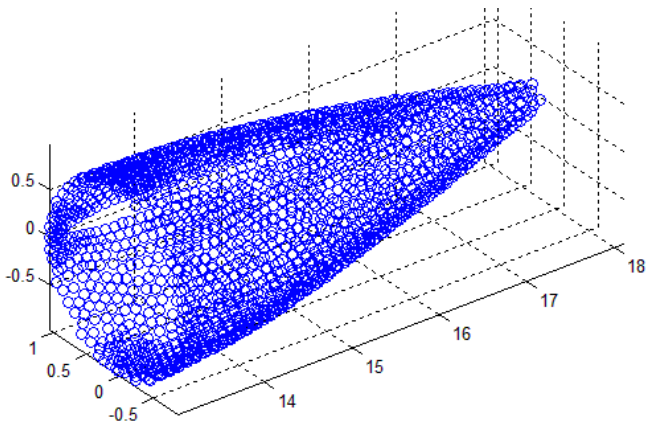


Figure 5.4(a)

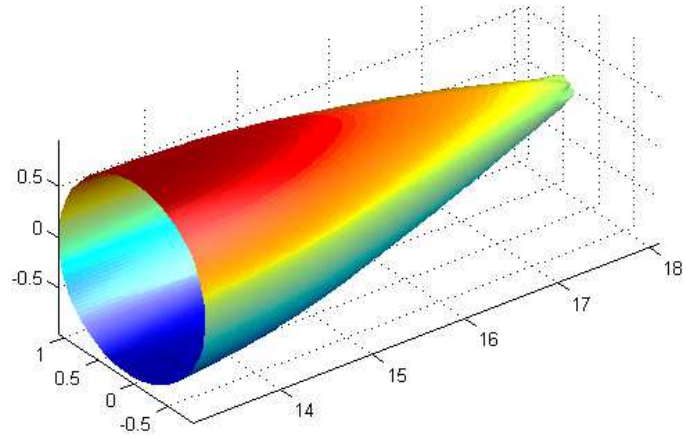


Figure 5.4(b)

Figure 5.2(b), 5.3(b) and 5.4(b) when overlapped; give the complete fuselage of Falcon aircraft. The fuselage has singularities at the very start and the end of surface. To obtain smooth surface parameterizations for those parts, we can extract their point cloud and find smooth surface parameterizations for them separately. Above figures are actually the overlaps of resulting patches.

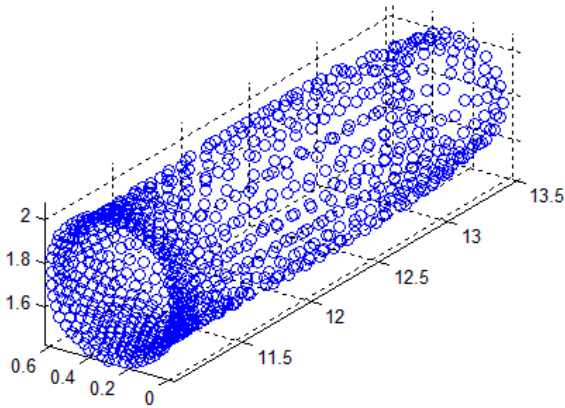


Figure 5.5(a)

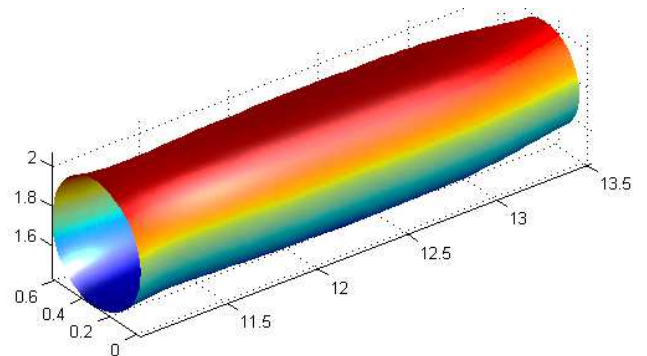


Figure 5.5(b)

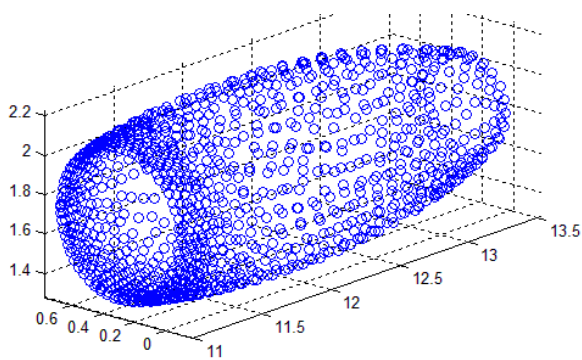


Figure 5.6(a)

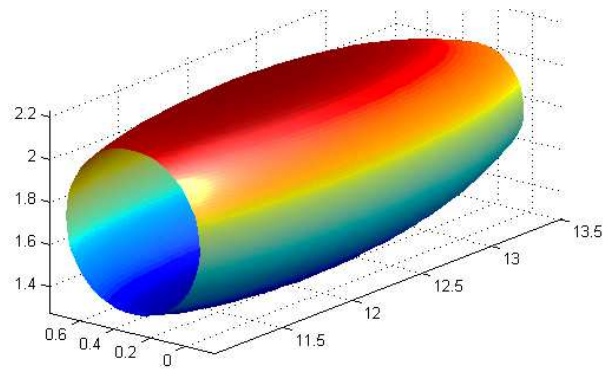


Figure 5.6(b)

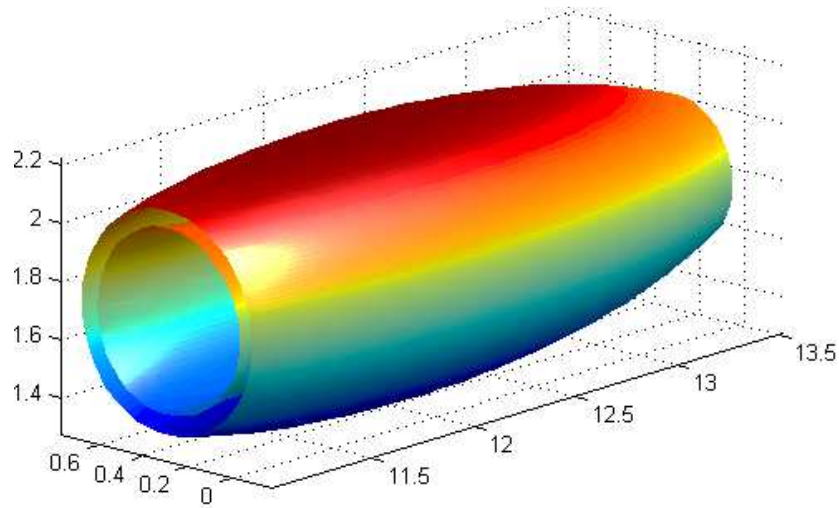


Figure 5.7

Figure 5.7 is the result of overlap of figure 5.5(b) and figure 5.6(b) which are the interior and exterior parts respectively, of the engines. For the front portion of engine, none of the projection surfaces (ellipsoid, hyperboloid or paraboloid) is a good projection surface. So we implemented a different class of curves, Surface of revolution (SOR). In this case, the SOR is a parabola rotated about an axis passing through the center of engine (the process automating the selection of a good surface of revolution & generalizing it, is in progress).

Following figures show our implementation of surface of revolution for the engine front portion. Once we know the parametric form of surface of revolution, we follow our approach of projecting the points on the surface and finding the surface parameterization for the given/extracted point cloud.

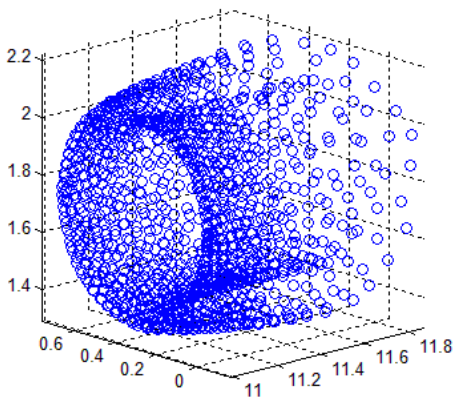


Figure 5.8(a)

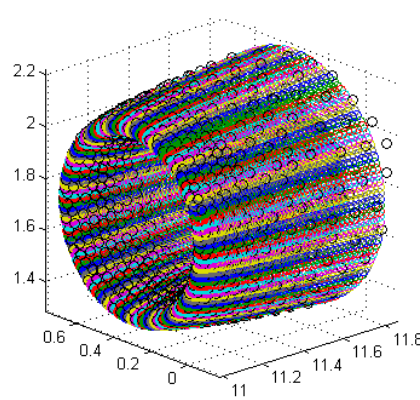


Figure 5.8(b)

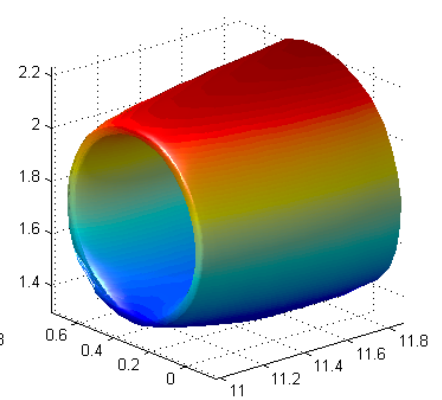


Figure 5.8(c)

Figure 5.8(a) shows the given point cloud for the front part of the engine. Figure 5.8(b) shows the selected surface of revolution along with the original point cloud (black points). Figure 5.8(c) shows the final surface obtained after applying projection and continuation algorithm Section 3. Figure 5.9 shows the overlap of figure 5.5(b), 5.6(b) and 5.8(c) to obtain the complete engine.

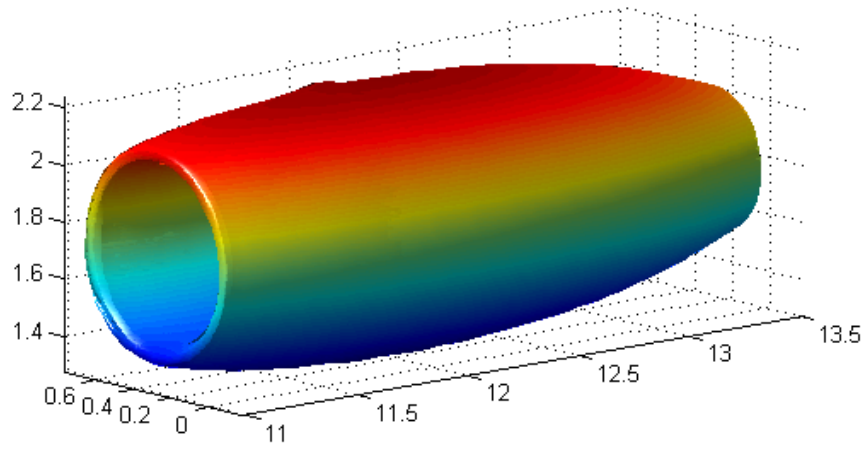


Figure 5.9

After overlapping the surfaces in figure 5.2(b), 5.3(b), 5.4(b), 5.5(b), 5.6(b) and 5.8(c), we obtain the final surface parameterization for the fuselage and engines as shown in Figure 5.10.

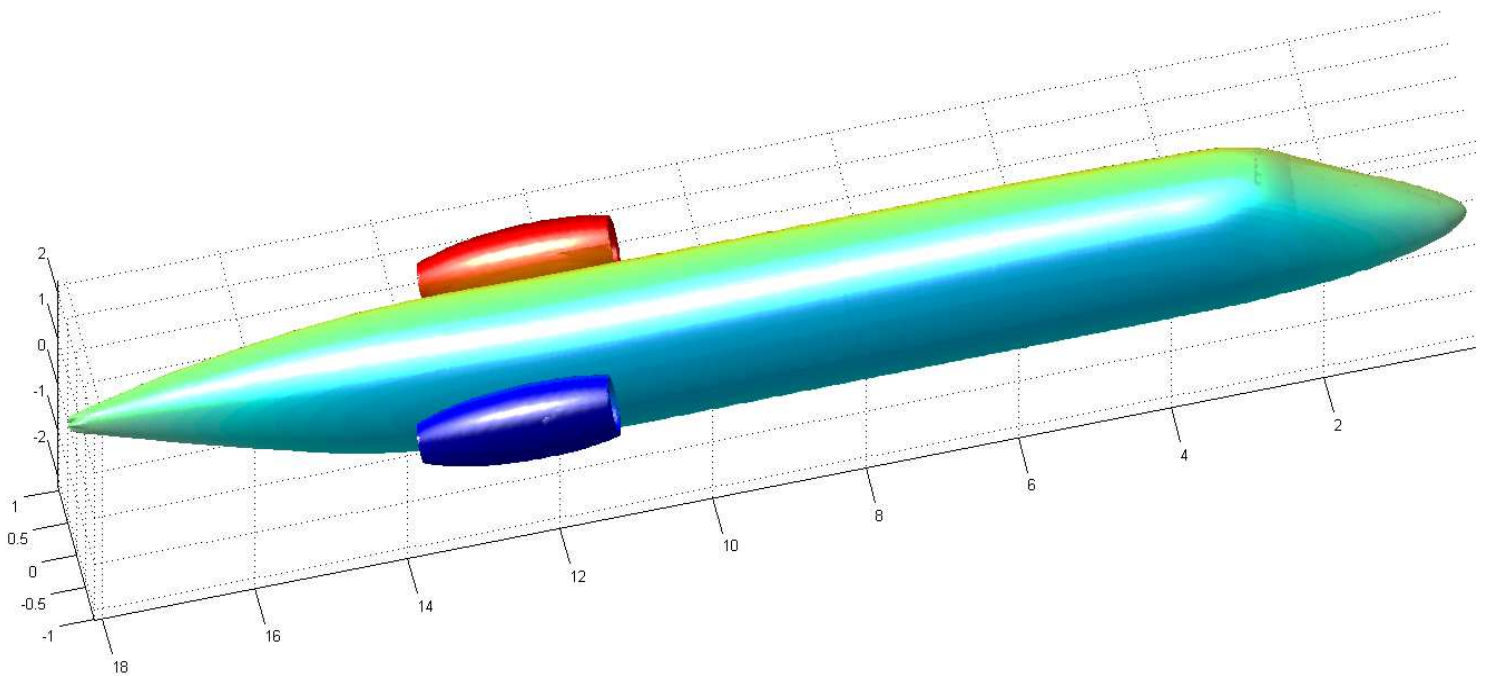


Figure 5.10

6 Conclusions and Future Work

6.1 Summary and Concluding Remarks

In this report, we presented an algorithm for making the intermediate process of finding good projection surface automatic. We demonstrated results of implementation of our approach for a randomly generated point cloud on a known surface and for the given point cloud of engines and fuselage of Falcon aircraft.

We interpolate the given point cloud by quadratic curves using the method of Least Squares and use Single value decomposition method to find the solution to the resulting homogeneous system of equations. Once we know the quadratic surfaces, we find their parametric form using the algorithm presented in section 4.3. After that, we project the given point cloud orthogonally on the projection surface and find the directed distance of each point from the point of projection. An algorithm for doing this efficiently has been presented in section 4.4. We then use Fourier Continuation method to find the smooth function for discrete directed distance point cloud and then we find the surface parameterization for the given point cloud. For this process to work, we need to pre-process the given point cloud to satisfy the requirements as we extracted the point cloud of singularity in our demonstration of 2-Dimensional function in section 4. For the front of engine of falcon aircraft, we implemented a different class of projection surface – surface of revolutions. (The process of generalization of SOR for any point cloud is in progress as mentioned in future directions).

6.2 Future Directions

- Automating the pre-processing of point cloud i.e. given a point cloud, extracting the point in cloud which will represent a smooth surface of piecewise smooth surface. (This takes into account dealing with geometric singularities).
- Implementation of all possible surfaces of revolution and finding an algorithm for selecting a suitable SOR for a given point cloud automatically.
- When the different patches of surface are overlapped, the overlapping may not be exact [1]. For this, we need to develop an algorithm to properly blend the surfaces of singularity.

REFERENCES

1. **Oscar P. Bruno, Youngae Han, Matthew M. Pohlman**, “Accurate, high-order representation of complex three-dimensional surfaces via Fourier continuation analysis “, JOURNAL OF COMPUTATIONAL PHYSICS, VOLUME 227, ISSUE 2, 10 DECEMBER 2007, PAGES 1094-1125.
2. **Karl Hahn**, “Solving Cubic and Quartic Polynomials”, JULY 23, 2005
<http://www.karlscalculus.org/pdf/cubicQuartic.pdf>
3. **Gene H. Golub, Charles F. Van Loan**, “Matrix Computations “, The John Hopkins University press; 3rd Edition (October 15, 1996).