2. Curve Fitting Algorithms

Mathematical software, particularly curve and surface fitting routines, is a critical element of coordinate metrology systems. An important but difficult task is to assess the performance of such routines [6]. Data analysis software can contribute significantly to the total measurement error of a Coordinate Measuring System (CMS). Factors affecting software performance include the choice of analysis method, the quality of software implementation, and characteristics of the specific measurement task. However, there are no standard mechanisms to evaluate the effects of these factors on the quality of the measurement results.

The Physikalisch-Technische Bundesanstalt (PTB) and National Institute of Standards and Technology (NIST) have implemented projects that can assist in assessing the performance of geometry-fitting routines [7] [8]. Both the projects have evaluated the performance of Gaussian curve and surface fitting least square algorithms. Yet there are unexplored areas.

Many of the commercially available software are based on Least Square Method (LSM). In this chapter, first, LSM and its variants are discussed and then other methods are discussed.

2.1 Least Square Method

The method of least squares is a standard approach to the approximate solution of over determined systems, i.e., sets of equations in which there are more equations than unknowns. "Least squares" means that the overall solution minimizes the sum of the squares of the errors made in the results of every single equation [9]. The following is the example to fit a plane from data set.

A simple data set consists of n points (data pairs) \( (x_i, y_i), i = 1, \ldots, n \) where \( x_i \) is an independent variable and \( y_i \) is a dependent variable whose value is found by observation. The model function has the form \( f(x, \beta) \), where the \( m \) adjustable parameters are held in the vector \( \beta \). The goal is to find the parameter values for the model which “best” fits the data. The least squares method finds its optimum when the sum \( F \), of squared residuals

\[
F = \sum_{i=1}^{n} d_i^2 \tag{2.1}
\]
is a minimum. A residual defined as the difference between the actual value of the
dependent variable and the value predicted by the model.

\[ d_i = y_i - f(x_i, \beta) \]  

(2.2)

A data point may consist of more than one independent variable. For
example, when fitting a plane to a set of height measurements, the plane is a
function of two independent variables, \(x\) and \(z\), say. In the most general case there
may be one or more independent variables and one or more dependent variables at
each data point.

The application of least square method can be applied to a wide range of
curve fitting problems in coordinate metrology. Least square best-fit element to
data is explained by taking the problem of fitting the data to a plane.

This is a problem of parameterization. The best plane can be specified by a
point \(C(x_0, y_0, z_0)\) on the plane and the direction cosines \((a, b, c)\) of the normal to the
plane. Any point \(P(x, y, z)\) on the plane satisfies

\[ a(x - x_0) + b(y - y_0) + c(z - z_0) = 0 \]  

(2.3)

The distance from any point \(Q(x_i, y_i, z_i)\) to a plane specified above is given by

\[ d_i = a(x_i - x_0) + b(y_i - y_0) + c(z_i + z_0) \]  

(2.4)

The sum of squares of distances of each point from the plane is calculated by (2.1).
Hence, the problem is to find the parameters \((x_0, y_0, z_0)\) and \((a, b, c)\) that minimizes
the sum \(F\).

The most important application is in data fitting. The best fit in the least-
squares sense minimizes the sum of squared residuals, a residual being the
difference between an observed value and the fitted value provided by a model.
When the problem has substantial uncertainties in the independent variable (the 'x'
variable), then simple regression and least squares methods have problems; in such
cases, the methodology required for fitting errors-in-variables models may be
considered instead of that for least squares.

Least squares problems fall into two categories: linear or ordinary least
squares and non-linear least squares, depending on whether or not the residuals are
linear in all unknowns. The linear least-squares problem occurs in statistical regression analysis; it has a closed-form solution. A closed-form solution (or closed-form expression) is any formula that can be evaluated in a finite number of standard operations. The non-linear problem has no closed-form solution and is usually solved by iterative refinement; at each iteration the system is approximated by a linear one, thus the core calculation is similar in both cases.

The least-squares method is usually credited to Carl Friedrich Gauss (1795), but it was first published by Adrien-Marie Legendre [10].

2.2 Linear Least Squared Method

Linear least squares is an approach for fitting a mathematical model to data in cases where the idealized value provided by the model for any data point is expressed linearly in terms of the unknown parameters of the model. The resulting fitted model can be used to summarize the data, to predict unobserved values from the same system, and to understand the mechanisms that may underlie the system.

Mathematically, linear least squares is the problem of approximately solving an over-determined system of linear equations, where the best approximation is defined as that which minimizes the sum of squared differences between the data values and their corresponding modeled values. The approach is called "linear" least squares since the assumed function is linear in the parameters to be estimated. Linear least squares problems are convex and have a closed-form solution that is unique, provided that the number of data points used for fitting equals or exceeds the number of unknown parameters, except in special degenerate situations.

2.2.1 Least Square Best Fit Line

The conventional approach for least square fitting of a straight line is described below for understanding. The matrix formulation of the problem is also explained in detail, as it is very useful when solving large scale problems.

Consider fitting a straight line

\[ g = a + bx \]  

(2.5)
through a set of data points \((x_i, y_i), i = 1, ..., n\). The minimizing function minimizes the sum of the squares of the distances of the points from the straight line measured in the vertical direction. Thus

\[
F = \sum_{i=1}^{n} (y_i - a - bx_i)^2
\]  

(2.6)
is the minimizing function. A necessary condition for \(F\) to be minimum is

\[
\frac{\partial F}{\partial a} = 0, \quad \frac{\partial F}{\partial b} = 0.
\]

Thus partial differentiation of the above function with respect to \(a\) and \(b\) gives

\[
2(-1) \sum_{i=1}^{n} (y_i - a - bx_i) = 0
\]

\[
2(-x) \sum_{i=1}^{n} (y_i - a - bx_i) = 0
\]

These can be simplified as

\[
\sum_{i=1}^{n} y_i = na + b \sum_{i=1}^{n} x_i
\]

\[
\sum_{i=1}^{n} x_i y_i = a \sum_{i=1}^{n} x_i + b \sum_{i=1}^{n} x_i^2
\]

The equations above can be solved simultaneously to find the values of \(a\) and \(b\).

**Normal Equation**

Consider to fit a straight line, \(y = a + bx\), to the set of data \((x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\). If the data points were collinear, the line would pass through \(n\) points. So

\[
y_1 = a + bx_1
\]

\[
y_2 = a + bx_2
\]

\[
y_3 = a + bx_3
\]

\[
\vdots
\]

\[
y_n = a + bx_n
\]

It can be written in a matrix form

\[
B = AC
\]

(2.7)
where, 

\[
B = \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \end{bmatrix}, \quad \text{and} \quad C = \begin{bmatrix} a \\ b \end{bmatrix}
\]

The objective here is to find a vector \( C \) that minimizes the Euclidean length of the difference \( \|B - AC\| \). If \( C = C^* \) is minimum vector, \( y = a' + b'x \) is a least square straight-line fit. This can be explained as

\[
\|B - AC\|^2 = (y_1 - a - bx_1)^2 + (y_2 - a - bx_2)^2 + \cdots + (y_n - a - bx_n)^2 \quad (2.8)
\]

Let \( d_1 = (y_1 - a - bx_1)^2, \ d_2 = (y_2 - a - bx_2)^2, \ldots, \ d_n = (y_n - a - bx_n)^2 \), then \( d \) can be explained as the distance from a point of a data set to a fitting line. So

\[
\|B - AC\| = \sum_{i=1}^{n} d_i \quad (2.9)
\]

To minimize \( \|B - AC\| \), \( AC \) must be equal to \( AC^* \), where \( AC^* \) is the orthogonal projection of \( B \) on the column space of \( A \) (refer Figure 2.1). This implies \( B - AC^* \) must be orthogonal to the column space of \( A \). So, \( (B - AC^*)AC = 0 \) for every \( C \) in \( \mathbb{R}^2 \).

![Figure 2.1 Finding Normal Equation](image_url)

This can be simplified as \( (A^T C - A^T AC^*)C = 0 \). So, \( A^T C - A^T AC^* \) is orthogonal to every vector \( C \) in \( \mathbb{R}^2 \). This implies \( A^T B - A^T AC^* = 0 \), which implies that \( C^* \) satisfies the linear system \( A^T AC^* = A^T B \). This equation is called as Normal equation. This will provide the solution for \( C \) in

\[
(A^T A)C = A^T B \quad (2.10)
\]
The equation (2.10) is difficult to solve because of the large size of $A^T A$. But, the same equation can be solved using singular value decomposition [11] [12]. A matrix can be decomposed in three matrices

$$A = U S V^T$$

(2.11)

where $U$ and $V$ are orthogonal matrices, and $S$ is a diagonal matrix containing the singular matrix of $A$. Putting (2.11) in (2.10), we have

$$(VS^T S V^T) C = V S^T U^T B$$

(2.12)

as $U^T U = I$, $U^T = U^{-1}$, $V^T V = I$ and $V^T = V^{-1}$. Multiplying (2.12) first by $V^T$, then by $S^T$ two times and finally by $V^T$, we get

$$V V^T C = V S^{-1} U^T B$$

Hence, the solution for $C$ is

$$C = V S^{-1} U^T B$$

Algorithm

Lines can be in a specified plane (2D) or in a space (3D). Since the 2D line is a special case of 3D line, the algorithm for 3D line is explained in detail.

Let, $P(x_i, y_i, z_i), i = 2, ..., m; m \geq 2$ is, $m$ point in data point set to fit a line. Any point $(x, y, z)$ on the line satisfies with $t$ as parameter

$$(x, y, z) = (x_0, y_0, z_0) + t(a, b, c)$$

(2.13)

It is known that the distance from a point to a line in three dimension is

$$d_i = \sqrt{u_i^2 + v_i^2 + w_i^2}$$

(2.14)

The best-fit line passes through the centroid $(\bar{x}, \bar{y}, \bar{z})$ of the data and this specifies a point on line. Also, the direction cosines have to be found out.
a. The first step is to find the average coordinates of the points \((x_i, y_i, z_i)\).

\[
\bar{x} = \frac{\sum x_i}{m} \\
\bar{y} = \frac{\sum y_i}{m} \\
\bar{z} = \frac{\sum z_i}{m}
\]  

(2.15)

b. The matrix \(A\) is formulated such that its first column is \(x_i - \bar{x}\), second column is \(y_i - \bar{y}\) and third column is \(z_i - \bar{z}\).

c. This matrix \(A\) is solved by singular value decomposition. The smallest singular value of \(A\) is selected from the matrix and corresponding singular vector is chosen which is the direction cosines \((a, b, c)\).  
d. The best-fit line is specified by \(\bar{x}, \bar{y}, \bar{z}, a, b\) and \(c\).

2.2.2 Least Square Best Fit Plane

The procedure to fit a plane to \(m\) data points \(P(x_i, y_i, z_i); i = 1, \ldots, m; m \geq 3\) is given below. Any point \((x, y, z)\) on the plane satisfies

\[
a(x-x_o) + b(x-x_o) + c(x-x_o) = 0
\]

(2.16)

where, \((a, b, c)\) is direction cosine of normal to the plane and \((x_o, y_o, z_o)\) is a point on plane. It is known that the distance from a point \(P(x_i, y_i, z_i); i = 3, \ldots, m\) to a plane specified by direction cosine of the normal to plane \((a, b, c)\) and a point on plane \((x_o, y_o, z_o)\) is given by

\[
d_i = a(x-x_o) + b(x-x_o) + c(x-x_o)
\]

(2.17)

Since the data has been translated to the origin, and the centroid of the data must be a point on the least square plane, the algorithm for the least square plane is similar to least square line.

Algorithm

The best fit plane passes through the centroid \((\bar{x}, \bar{y}, \bar{z})\) of the data and this specifies a point on least square plane. Also, direction cosines normal to least square plane have to be found out.

The first step is to find the average of the points
a. The first step to find the average coordinates of the points \((x_i, y_i, z_i)\).

\[
\bar{x} = \frac{\sum x_i}{m}, \quad \bar{y} = \frac{\sum y_i}{m}, \quad \bar{z} = \frac{\sum z_i}{m}
\]

b. The matrix \(A\) is formulated such that its first column is \(x_i - \bar{x}\), second column is \(y_i - \bar{y}\) and third column is \(z_i - \bar{z}\).

c. This matrix \(A\) is solved by singular value decomposition. The smallest singular value of \(A\) is selected from the matrix and corresponding singular vector is chosen which is the direction cosines \((a, b, c)\).

d. The best-fit plane is specified by \(\bar{x}, \bar{y}, \bar{z}, a, b\) and \(c\).

### 2.3 Non-linear Least Square Method

Non-linear least squares is the form of least squares analysis which is used to fit a set of \(m\) observations with a model that is non-linear in \(n\) unknown parameters \((m > n)\). It is used in some forms of non-linear regression. The basis of the method is to approximate the non-linear model by a linear one and to refine the parameters by successive iterations. There are many similarities to linear least squares, but there are some significant differences also.

Consider a set of \(m\) data points, \((x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\) and a curve (model function) \(y = f(x, \beta)\), that in addition to the variable \(x\) also depends on \(n\) parameters, \(\beta = (\beta_1, \beta_2, \ldots, \beta_n)\), with \(m > n\). It is desired to find vector \(\beta\) of parameters such that the curve best fits the given data in the least squares sense, that is, the sum of squares

\[
S = \sum_{i=1}^{m} r_i^2
\]

(2.18)

is minimized, where the residuals (errors) \(r_i\) are given by

\[
r_i = y_i - f(x_i, \beta)
\]

(2.19)

for \(i = 1, 2, \ldots, m\).

The minimum value of \(S\) occurs when the gradient is zero. Since the model contains \(n\) parameters, there are \(n\) gradient equations:
\[
\frac{\partial S}{\partial \beta_j} = 2 \sum_{i=1}^{m} r_i \frac{\partial r_i}{\partial \beta_j} = 0
\]  

(2.20)

for \( j = 1, 2, \ldots, n \). In non-linear system, the derivatives \( \frac{\partial r_i}{\partial \beta_j} \) are functions of both the independent variable and the parameters, so these gradient equations do not have a closed solution. Instead, initial values must be chosen for the parameters. Then, the parameters are refined iteratively, that is, the values are obtained by successive approximation,

\[
\beta_j \approx \beta_j^{k+1} = \beta_j^{k} + \Delta \beta_j
\]

Here, \( k \) is iteration number and the vector of increments, \( \Delta \beta \) is known as the shift vector. At each iteration the model is linearized by approximation to a first-order Taylor series expansion about \( \beta^k \).

\[
f(x_i, \beta) \approx f(x_i, \beta^k) + \sum_j \frac{\partial f(x_i, \beta^k)}{\partial \beta_j} (\beta_j - \beta_j^k)
\]

\[
\approx f(x_i, \beta^k) + \sum_j J_{ij} \Delta \beta_j
\]

The Jacobian \( J \) is a function of constants, the independent variable and the parameters, so it changes from one iteration to the next. Thus, in terms of the linearized model, \( \frac{\partial r_i}{\partial \beta_j} = -J_{ij} \) and the residuals are given by

\[
r_i = \Delta y_i - \sum_{j=1}^{n} J_{ij} \Delta \beta_j = -f(x_i, \beta^k).
\]  

(2.21)

Substituting (2.21) into the gradient equation (2.20),

\[
-2 \sum_{i=1}^{m} J_{ij} (\Delta y_i - \sum_{j=1}^{n} J_{ij} \Delta \beta_j) = 0
\]

Which, on rearrangement, become \( n \) simultaneous linear equations, the normal equation

\[
\sum_{i=1}^{m} \sum_{j=1}^{n} J_{ij} J_{ij} \Delta \beta_j = \sum_{i=1}^{m} J_{ij} \Delta y_i
\]

for \( j = 1, 2, \ldots, n \).
The normal equations are written in matrix notation as

\[(J^T J) \Delta \beta = J^T \Delta y.\]

When the observations are not equally reliable, a weighted sum of squares may be minimized,

\[S = \sum_{i=1}^{m} W_{ii} r_i^2\]

Each element of the diagonal weight matrix \( W \) should, ideally, be equal to the reciprocal of the error variance of the measurement [13]. The normal equations are then

\[(J^T W J) \Delta \beta = J^T W \Delta y.\]

These equations form the basis for the Gauss-Newton algorithm for a non-linear least squares problem.

In linear least squares the objective function, \( S \), is a quadratic function of the parameters.

\[S = \sum_i W_i (y_i - \sum_j X_{ij} \beta_j)^2\]

When there is only one parameter, the graph of \( S \) with respect to that parameter will be a parabola. With two or more parameters the contours of \( S \) with respect to any pair of parameters will be concentric ellipses (assuming that the normal equations matrix \( X^T W X \) is positive definite). The minimum parameter values are to be found at the center of the ellipses. The geometry of the general objective function can be described as paraboloid elliptical. In non-linear least square, the objective function is quadratic with respect to the parameters only in a region close to its minimum value, where the truncated Taylor series is a good approximation to the model.

\[S \approx \sum_i W_i (y_i - \sum_j J_{ij} \beta_j)^2\]

The more the parameter values differ from their optimal values, the greater the contours deviate from elliptical shape. A consequence of this is that initial parameter estimates should be as close as practicable to their (unknown!) optimal values. It also explains how divergence can come about as the Gauss–Newton method.
algorithm is convergent only when the objective function is approximately quadratic in the parameters.

**Initial Parameter Estimates**

Problems of ill-conditioning and divergence can be improved by finding initial parameter estimates that are near to the optimal values. A good way to do this is by computer simulation. Both the observed and calculated data are displayed on a screen. The parameters of the model are adjusted by hand until the agreement between observed and calculated data is reasonably good. Although this will be a subjective judgment, it is sufficient to find a good starting point for the non-linear refinement [13].

**Gauss-Newton Method (GNM)**

The normal equations

\[(J^TWJ)\Delta\beta = (J^TW)\Delta y\]

may be solved for \( \Delta\beta \) by Cholesky decomposition, as described in linear least squares. The parameters are updated iteratively

\[\beta^{k+1} = \beta^k + \Delta\beta\]

where \( k \) is an iteration number. While this method may be adequate for simple models, it will fail if divergence occurs. Therefore protection against divergence is essential. If divergence occurs, a simple expedient is to reduce the length of the shift vector, \( \Delta\beta \), by a fraction, \( f \)

\[\beta^{k+1} = \beta^k + f\Delta\beta\]

For example the length of the shift vector may be successively halved until the new value of the objective function is less than its value at the last iteration. The fraction \( f \) could be optimized by a line search [13]. As each trial value of \( f \) requires the objective function to be re-calculated, it is not worth optimizing its value too stringently.

When using shift-cutting, the direction of the shift vector remains unchanged. This limits the applicability of the method to situations where the direction of the shift vector is not very different from what it would be if the objective function were approximately quadratic in the parameters, \( \beta^0 \).
2.3.1 Least Square Best Fit Circle

Let \((x_i, y_i), i = 1, 2, \ldots, m, m \geq 3\) be the data points representing a circle. Any point \((x, y)\) on the circle satisfies

\[
(x - x_0)^2 + (y - y_0)^2 = r^2. \tag{2.22}
\]

It is known that the distance from a point \((x_i, y_i)\) to a circle specified by its circle \((x_0, y_0)\) and radius \(r\) is given by

\[
d_i = r - r = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - r \tag{2.23}
\]

and the elements of the Jacobian matrix \(J\) are found by the partial derivative of \(d_i\) with respect to the parameters \((x_0, y_0)\) and \(r\) by

\[
\frac{\partial d_i}{\partial x_0} = \frac{-(x_i - x_0)}{r_i}, \quad \frac{\partial d_i}{\partial y_0} = \frac{-(y_i - y_0)}{r_i}, \quad \frac{\partial d_i}{\partial r} = -1 \tag{2.24}
\]

Algorithm

The algorithm that is used to find best-fit circle is Gauss-Newton algorithm, which is already explained. The initial estimates of the center and radius of the circle are made available by solving the problem as linear least square model. The steps followed are as:

a. The objective function which is to be minimized is

\[
F = \sum_{i=1}^{m} f_i^2 = \sum_{i=1}^{m} (r_i^2 - r^2)
\]

b. This can be reduced to a linear system in \(x_0, y_0\) and \(\rho\) as,

\[
f_i = (x_i - x_0)^2 + (y_i - y_0)^2 - r^2 = -2x_ix_0 - 2y_iy_0 + \rho + (x_i^2 + y_i^2),
\]

where \(\rho = (x_0^2 + y_0^2 - r^2)\)

c. For minimizing \(F\), following linear least square system is to be solved.

\[
AP = B \tag{2.25}
\]
where, the elements of the $i^{th}$ row of the $A$ are the coefficients $(2x_i, 2y_i, -1)$, the $i^{th}$ element of column vector $B$ is $(x_i^2 + y_i^2)$ and column vector $P$ is 

\[
\begin{bmatrix}
x_0 \\
y_0 \\
\rho
\end{bmatrix}
\]

d. An estimate of $r$ is obtained from the equation of $\rho$. These steps give initial estimate of center and radius of the circle.

e. Once the initial estimates are obtained the right hand side vector $d$ and Jacobean matrix $J$ are formed. Then, the following linear least-squares system is solved.

\[
J \begin{bmatrix} P_{x_0} \\ P_{y_0} \\ P_{\rho}
\end{bmatrix} = -d
\]

f. The values of the parameters are updated according to following equations.

\[
\begin{align*}
x' &= x_0 + P_{x_0} \\
y' &= y_0 + P_{y_0} \\
r' &= r + P_{\rho}
\end{align*}
\]

g. Steps ‘e’ and ‘f’ are repeated until the algorithm converges.

2.3.2 Least Square Best Fit Sphere

A sphere is specified by a center $(x, y, z)$ and a radius $r$. Any point on the sphere satisfies

\[
(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2.
\]  

A minimizing function has to be identified to obtain an initial estimate for the center and radius. Consider the function $f_i = r_i - r$, where $r_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}$. Differentiating this function with respect to $x_0$, $y_0$, $z_0$ and $r_0$ will result in complicated equations which are difficult to solve.

Therefore, consider the function $f_2 = r_i^2 - r^2$. This function can be written as $f_2 = (r_i - r)(r_i + r) \approx 2r(r_i - r)$, since $(r_i + r)$ can be approximated as $2r$. Differentiate this function with respect to $x_0$, $y_0$, $z_0$ and $r_0$ to obtain initial estimates of the center.
and radius of sphere. Thus the minimizing function to obtain initial estimates for a sphere is

\[ f = r_i^2 - r^2. \]  

(2.27)

**Algorithm**

The initial estimates of the center and radius of the sphere are made available by solving the problem same as least square circle. The steps are followed are as:

a. Expanding the equation (2.27), we get

\[ f = -(2x_i x_o + 2y_i y_o + 2z_i z_o) + \rho + (x_i^2 + y_i^2 + z_i^2) \]

Where, \( \rho = (x_o^2 + y_o^2 + z_o^2) - r^2 \). The variable \( \rho \) is introduced to make the equation linear.

b. The above set of equations for set of data points are now represented in matrix form which is same as equation for least square circle i.e. equation (2.25). But, here elements of \( i^{th} \) row of matrix A are the coefficients \((-2x, -2y, -z, 1)\), the \( i^{th} \) element of B is \((x_i^2 + y_i^2 + z_i^2)\) and column vector P is

\[
[x_0 \ y_0 \ z_0 \ \rho]^T.
\]

c. An initial estimate of \( r \) is obtained from the equation of \( \rho \).

After obtaining the initial estimates for the center and radius \( r \), the Gauss-Newton method is used to arrive at the final values for center and radius. The elements of the Jacobian matrix for the given minimizing function is

\[
J = \begin{bmatrix}
\frac{\partial d_i}{\partial x_0} & \frac{\partial d_i}{\partial y_0} & \frac{\partial d_i}{\partial z_0} & \frac{\partial d_i}{\partial r_0} \\
\frac{\partial d_i}{\partial x_o} & \frac{\partial d_i}{\partial y_o} & \frac{\partial d_i}{\partial z_o} & \frac{\partial d_i}{\partial r_o} \\
\vdots & \vdots & \vdots & \vdots \\
\frac{\partial d_i}{\partial x_o} & \frac{\partial d_i}{\partial y_o} & \frac{\partial d_i}{\partial z_o} & \frac{\partial d_i}{\partial r_o}
\end{bmatrix}.
\]

Evaluating various components of the Jacobian and substituting in the matrix, we get
Solution of equation $JP = -d$, where $P = [P_z, P_y, P_x, P_t]^T$ gives correction for the parameters $x_o, y_o, z_o$ and $r_o$. Repeat this last step until the algorithm is converged. The convergence condition is given by $g = J^Td$.

### 2.3.3 Least Square Best Fit Cylinder

Assume that a cylinder is fitted to $m$ points $(x_i, y_i, z_i)$, $i = 1, ..., m$ and $m \geq 5$.

The cylinder is specified by

$$v = v + a (x_i - x_o) + b (y_i - y_o) + c (z_i - z_o)$$

where, $(x_o, y_o, z_o)$ is a point on axis of cylinder, $(a, b, c)$ is vector pointing along the axis and $r$ is radius of its base. The equation (2.28) can be rewritten as

$$C x_i^2 + D y_i^2 + E z_i^2 + F x_i y_i + G x_i z_i + H y_i z_i + I x_i + K y_i + L z_i + M = 0$$

where, $C = (b^2 + c^2)$, $D = (a^2 + c^2)$, $E = (a^2 + b^2)$, $F = -2ab$, $G = 2ac$, $H = -2bc$, $I = -2(b^2 + c^2)x_o + 2aby_o + 2acz_o$, $K = 2abx_o - 2(a^2 + c^2)x_o + 2bcz_o$, $L = 2acx_o + 2bcy_o - 2(a^2 + c^2)z_o$, and $M = (b^2 + c^2)x_o^2 + (a^2 + c^2)y_o^2 + (b^2 + a^2)z_o^2 - 2bcy_o z_o - 2acx_o z_o - 2abx_o y_o - r^2$. Dividing both sides of equation (2.29) and rearranging, it reduces to a linear system as

$$AP = B$$

where, $A = \begin{bmatrix} y_i^2 & z_i^2 & x_i y_i & x_i z_i & y_i z_i & x_i & y_i & z_i & 1 \\ y_2^2 & z_2^2 & x_2 y_2 & x_2 z_2 & y_2 z_2 & x_2 & y_2 & z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_m^2 & z_m^2 & x_m y_m & x_m z_m & y_m z_m & x_m & y_m & z_m & 1 \end{bmatrix}$, $P = \begin{bmatrix} D & E & F & G & H & I & K & L & M \end{bmatrix}^T$ and
\[ B = \begin{bmatrix} -x_1^2 & -x_2^2 & \cdots & -x_n^2 \end{bmatrix}^T. \]

Let, \( D' = D/C, \ E' = E/C, \ G' = G/C, \ G' = G/C, \ H' = H/C, \ I' = I/C, \ K' = K/C, \ L' = L/C \) and \( M' = M/C \). Solving for this; if \( |F'|, |G'|, |H'|, |I'| \) are close to zero, then if \( D' \) is close to one implies \( (a, b, c) = (0, 0, 1) \) or else if \( E' \) is close to one implies \( (a, b, c) = (0, 1, 0) \) otherwise

\[ k = \frac{2}{a + D' + E'}. \tag{2.31} \]

If \( C \) and \( D \) are close one, then

\[ E' = \sqrt{(1 - E)}, C' = G / 2E', D' = -F / 2E'. \]

If \( C \) is close to one, \( B \) is not close to one, then

\[ D' = \sqrt{1 - C}, C' = -F / 2D', F' = -H / 2D'. \]

If \( C \) is not close to one, then

\[ C' = \sqrt{1 - C}, D' = -F / 2C', E = -G / 2C'. \]

The direction \( \{a', b', c'\} \) is normalized to get the direction \( \{a, b, c\} \).

The point on axis is initially estimated by knowing \( \{a, b, c\} \), the definition of the coefficients \( G, H, I \) and equation \( ax_0 + by_0 + cz_0 = 0 \). This will form into the linear system as

\[
\begin{bmatrix}
-2(b^2 + c^2) & 2ab & 2ac \\
2ab & -2(a^2 + c^2) & 2bc \\
2ac & 2bc & -2(a^2 + b^2)
\end{bmatrix}
\begin{bmatrix}
x_0 \\
y_0 \\
z_0
\end{bmatrix} = \begin{bmatrix}
G \\
H \\
I
\end{bmatrix}. \tag{2.32}
\]

The initial estimate for radius can be done using definition of \( M \) given in equation (2.29).

Algorithm

a. Translate data point such that the point on the axis lies at the centroid of the point.

\[
(x, y, z) = (x_i, y_i, z_i) - (x_0, y_0, z_0)
\]
b. Transform the data by rotation matrix $U$ which rotates $\begin{pmatrix} a & b & c \end{pmatrix}$ to a point on the $z$-axis

$$
\begin{pmatrix}
x_i \\
y_i \\
z_i
\end{pmatrix}^T = U \begin{pmatrix}
x_i \\
y_i \\
z_i
\end{pmatrix}^T
$$

where

$$
U = \begin{bmatrix}
C_2 & 0 & S_2 \\
0 & 1 & 0 \\
-S_2 & 0 & C_2
\end{bmatrix}
$$

If $\begin{pmatrix} a & b & c \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$, then $s_1 = 0, c_1 = 1, s_2 = -1, c_2 = 0$. Otherwise, $c_1 = c / \sqrt{b^2 + c^2}$, $c_2 = (cc_1 - bs_1) / \sqrt{a^2 + (cc_1 - bs_1)^2}$, $s_2 = -a / \sqrt{a^2 + (cc_1 - bs_1)^2}$. This particular transformation transforms the axis of cylinder such that the axis is in $YZ$-plane and then rotate the axis of cylinder about the $y$-axis to make it along the $z$-axis.

c. Calculate $d$ vector from data points $(x_i, y_i, z_i) \ i = 1, ..., m$ and Jacobian matrix using following equations.

$$
d = r - r
$$

where

$$
r_i = \sqrt{u_i^2 + v_i^2 + w_i^2}, \quad u_i = c(y_i - y_0) - b(z_i - z_0), \quad v_i = a(z_i - z_0) - c(x_i - x_0)
$$

and

$$
w_i = b(x_i - x_0) - a(y_i - y_0)
$$

and

$$
J = \begin{bmatrix}
-x_1 / r_1 & -y_1 / r_1 & -z_1 / r_1 & -x_1 z_1 / r_1 & -y_1 z_1 / r_1 & -1 \\
-x_2 / r_2 & -y_2 / r_2 & -z_2 / r_2 & -x_2 z_2 / r_2 & -y_2 z_2 / r_2 & -1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
-x_m / r_m & -y_m / r_m & -z_m / r_m & -x_m z_m / r_m & -y_m z_m / r_m & -1
\end{bmatrix}
$$

d. Solve following linear-square system.

$$
\begin{bmatrix}
p_{x_0} \\
p_{y_0} \\
p_b \\
p_c
\end{bmatrix} = -d
$$

e. Update the parameter estimates according to following.
Above steps are repeated until the algorithm converges.

2.4 More on Least Square Method

The Least Square Method for standard curves (features) using Gauss-Newton optimizing algorithm are discussed in section 2.1 to 2.3. Gander, Colub and Trebel used Gauss-Newton Algorithm (GNA) to minimize the square of error distances for circle [14]. If GNA starts in the close vicinity of the solution, it converges quickly. In other cases, it requires more iteration to converge and sometimes it may not converge at all. Hence, a good initial guess is required for the solution to converge [13] [15]. The GNM is like Newton’s method with line search strategy. The Levenberg-Marquardt Method can be derived by replacing the line search strategy with a trust-region strategy. The use of a trust region avoids the weakness of GNM i.e. initial guess is required near to the solution.

The Levenberg-Marquardt algorithm has been implemented in the Algorithm Testing System (ATS) at National Institute of Standards and Technology (NIST), USA by members of the ASME B89.4.10 Working Group [16]. The algorithms for standard features have successfully solved a number of difficult fitting problems that could not be solved by many commercial software packages used on Coordinate Measuring Systems (CMSs). (Some of the most difficult problems are cylinders or cones sampled over a small patch at NIST.) The Levenberg-Marquardt algorithms have an extremely broad range of convergence. Failure to converge has only been observed for pathological fitting problems (e.g., fitting a circle to collinear points). Special checks can detect most of these situations. The Levenberg-Marquardt Algorithm (LMA) is generally robust. For most fits, a good starting guess is not required to reach the global minimum. This is due, in part, to the careful choice of fitting parameters, the use of certain constraints and for cylinders and cones, the technique of restarting a search after an initial
solution is found. However, even for well-behaved functions and reasonable starting parameters, the LMA tends to be a bit slower than the GNA. LMA can also be viewed as improved GNA with trust region approach [13] [17]. Also, convergence of the solution is highly dependent on choice of Levenberg-Marquardt parameter and its selection is challenging.

Apart from least square best fit standard features like circle, the algorithms for least square best fit of other features are also discussed by different researchers. The algorithms for least square best fit cone and torus is presented in [16] and [18]. Ahn S. J. et al have presented algorithms for least square fit hyperbola and parabola [19]. The work is extended in three dimension by Dai M. et al by developing algorithm for least square fit paraboloids [20].

2.5 Other Circle Fitting Algorithms

Chernov and Ososkov proposed two new set of algorithms for full circle-fitting and circular arc namely Iterational Linear Regression Method (ILRM) and Modified Linear Regression Method (MLRM) [21]. Although ILRM is well-suited for fitting any size of circular arc including full circle, it is slower. The second suggested method (MLRM) is faster, but works only for small arcs. These algorithms are quite typically used in high energy physics especially automatic recognition of elementary particle trajectories on images of multiple track events produced by some physical device that detects particle collisions.

Drezner, Steiner and Wesolowsky suggested use of heuristic algorithms for finding a circle whose circumference is close to given set of points [22]. Three objectives are considered: minimizing the sum of squares of distances (Least Squares), minimizing the maximum distance (Minimax), and minimizing the sum of distances (Minisum). If the distance between \(i\)th point and the center of circle is calculated by the equation (2.22), the objective functions are:

Least Squares:

\[
\min_{x,y,r} \left\{ \sum_{i=1}^{n} (d_i - r)^2 \right\} \quad (2.34)
\]

Minmax:

\[
\min_{x,y,r} \left\{ \max_{i} \{ |d_i - r| \} \right\} \quad (2.35)
\]
It is proved that these problems are equivalent to minimizing the variance, minimizing the range, and minimizing the mean absolute deviation, respectively. These problems are formulated and heuristically solved as mathematical programs.

2.6 Trust-region Strategy: Dogleg Method

It is started by stating the Taylor’s theorem, which is the main tool used to solve the unconstrained nonlinear least square problems.

**Taylor's theorem:** Suppose that $f: \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and $p \in \mathbb{R}^n$. Then we have

$$f(x + p) = f(x) + \nabla f(x) \cdot p + \frac{1}{2} p^T \nabla^2 f(x) p$$

For, some $t \in (0, 1)$. Moreover, if $f$ is twice continuously differentiable, we have necessary and sufficient conditions for the optimality of $f$ are describe by placing the conditions on the gradient $\nabla f(x)$ and the Hessian $\nabla^2 f(x)$ [13].

To solve the unconstrained optimization of smooth function many algorithms are developed. All the algorithms require to supply a starting point, say, $x_0$ and generates a sequence of iterates $\{x_k\}_{k=0}^\infty$ that terminates when either no more progress can be made or when it seems that a solution point has been approximated with sufficient accuracy. There are two fundamental strategies for moving from the current point $x_k$ to a new point $x_{k+1}$: The line-search strategy and the trust-region strategy.

The Guass-Newton Methods described in section 2.3 are typical examples of line-search strategy. The Levenberg-Marquardt Algorithms described by Shakarji are typical cases of trust-region strategy [16]. However, the convergence of the solution is highly dependent on choice of Levenberg-Marquardt parameter. The selection of this parameter is challenging. Another variant of trust-region strategy is developed to evaluate the circularity and sphericity during the research work.
In the trust-region strategy, the information gathered about \( f \) is used to construct a model function \( m_k \) whose behavior near the current point \( x_k \) is similar to that of the actual objective \( f \). Because the model \( m_k \) may not be a good approximation of \( f \) when \( x \) is far from \( x_k \), the search is restricted for the minimize of \( m_k \) to some region around \( x_k \). In other words, the candidate step \( p \) is found by approximately solving the sub-problem:

\[
\min_p m(x_k + p) \tag{2.37}
\]

where, \( x_k + p \) lies inside the trust-region.

If the candidate solution does not produce a sufficient decrease in \( f \), we conclude that the trust-region is too large and it should be shrunk and resolved. Usually a trust-region is a ball defined by \( \|p\| < \Delta \), where the scalar \( \Delta > 0 \) is called the trust-region radius [13]. The model \( m_k \) in (2.37) is usually defined to be a quadratic function of the form

\[
m_k(x_k + p) = f_k + p^T \nabla f_k + \frac{1}{2} p^T B_k p
\]

where \( f_k, \nabla f_k \) and \( B_k \) are a scalar, vector and matrix, respectively. \( f_k \) and \( \nabla f_k \) are chosen to be the function and gradient values at the point \( x_k \), so that \( m_k \) and \( f \) are in agreement to the first order at the current iterate \( x_k \). The matrix \( B_k \) is either the Hessian \( \nabla^2 f_k \) or some approximation to it.

The formulation of the problem under the strategy is given as:

Let \( (x_i, y_i) \quad i = 1, 2, \ldots, m, m \geq 3 \) be the data points representing a circle. Any point \((x, y)\) on the circle satisfies

\[
(x - x_0)^2 + (y - y_0)^2 = r^2.
\]

It is known that the distance from a point \((x_i, y_i)\) to a circle specified by \( X = (x_0, y_0, r) \) where \( x_0, y_0 \) are coordinates of the center of circle and \( r \) is its radius, given by
\[ d_i(X) = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - r \]

and

\[ r(X) = (d_1, d_2, \ldots, d_n)^T \]

and hence, the objective function will be,

\[ f(X) = \frac{1}{2} \| r \|^2, \tag{2.38} \]

which is almost smooth.

Consider a linear model for the function \( r(X) \) as,

\[ r(X + p) = r(X) + J(X) \tag{2.39} \]

where, \( J(X) = \left[ \frac{\partial d_i}{\partial x_0} \frac{\partial d_i}{\partial y_0} \frac{\partial d_i}{\partial r} \right] \), \( 1 \leq i \leq m \) is Jacobian of \( r(X) \). \( \frac{\partial d_i}{\partial x_0} = \frac{(x_i - x_0)}{d_i + r} \),

\[ \frac{\partial d_i}{\partial y_0} = \frac{(y_i - y_0)}{d_i + r} \quad \text{and} \quad \frac{\partial d_i}{\partial r} = -1, 1 \leq i \leq m \]

represents the column in \( J(X) \). Equation (2.39) generates a sub-problem for the objective function \( f \) given in (2.38) as,

\[ \min_r \frac{1}{2} \| r + Jp \|^2 \tag{2.40} \]

To solve the sub-problem (2.40), the set up for a spherical trust-region strategy is,

\[ \min_r \frac{1}{2} \| r + J_kp \|^2 \]

subject to \( \| p \| < \Delta_k \)

\[ \| p \| < \Delta_k \]

where, \( \Delta_k > 0 \) is the trust-region radius. Choosing

\[ m_k(p) = \frac{1}{2} \| r + p^T J_k r \| + \frac{1}{2} p^T J_k J_k p \]

as the model function; where \( J^T r = \nabla f \) and the matrix \( B \) is taken as the first order derivative approximation \( J^T J \), to the Hessian \( \nabla^2 f \), neglecting the second order derivative terms, as \( J^T J \) is a symmetric and positive semidefinite.
The Dogleg method, finds an approximate solution step $p$ along the trajectory $\tilde{p}(\tau)$ for $\tau \in [0, 2]$, for $\|p^U\| > \Delta$, when $J^T J$ is positive definite [13]. This can be given by

$$\tilde{p}(\tau) = \begin{cases} \tau p^U; & 0 \leq \tau \leq 1, \text{if } \|p^U\| > \Delta \\ p^U + (\tau - 1)(p^b - p^U); & 1 \leq \tau \leq 2 \end{cases}$$

Here, $\tau p^U; \ 0 \leq \tau \leq 1$ is a line-segment runs from the origin to the unconstrained minimize along the steepest decent direction given by,

$$p_U = -\frac{(J^T r)T(J^T r)}{(J^T r)T(J^T r)(J^T r)} J^T r$$

while, $p^U + (\tau - 1)(p^b - p^U); \ 1 \leq \tau \leq 2$ runs from $p^U$ to the full step solution $p^b$, given by, $p^b = -(J^T J)^{-1}(J^T r)$. In case, $\tilde{p}(\tau) = \tau p^U; \ \tau$ is taken as $\tau = \frac{\Delta}{\|p^U\|}$ and the step $p$ is taken as $p = \frac{\Delta}{\|p^U\|} p^U$, when $\|p^U\| \leq \Delta$, the Dogleg path $\tilde{p}(\tau)$ intersects with the trust-region boundary at exactly one point and that point (step) $p$ is obtained by solving

$$\|p^U + (\tau - 1)(p^b - p^U)\|^2 = \Delta^2$$

for, $\tau$ and step $p$ is taken as $p = p^U + (\tau - 1)(p^b - p^U)$ and for, $\|p^U\| \leq \Delta$, the step $p$ is chosen as, $p = p^b$. 

38