1.1 Introduction

The objective of this report was to develop algorithms for optimization of least square best fit geometry for various geometric elements. The matlab code implementation was done for the various geometries and tested with the real data points obtained from co-ordinate measuring machines. The various geometries that were studied here are as follows:

Lines in a specified plane
Lines in three dimension
Planes
Circles in a specified plane
Spheres and
Cylinders

This report was written in such a way that someone who has some knowledge of Linear Algebra and curve fitting could easily follow. The concise abstract of various papers that the authors referred is given as Appendix 1 and 2. It is advised to go through the Appendix before starting to read the report. The least square best-fit reference element to Cartesian data points was only established in this report.

1.2.Least Squares Best Fit Element

The application of least square criteria can be applied to a wide range of curve fitting problems. 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 parametrization. The best plane can be specified by a point C (x_o, y_o, z_o) on the plane and the direction cosines (a, b, c) of the normal to the plane. Any point (x,y,z) on the plane satisfies

$$a(x- x_o)+b(y- y_o)+c(z- z_o) = 0 \quad (1.2.1)$$

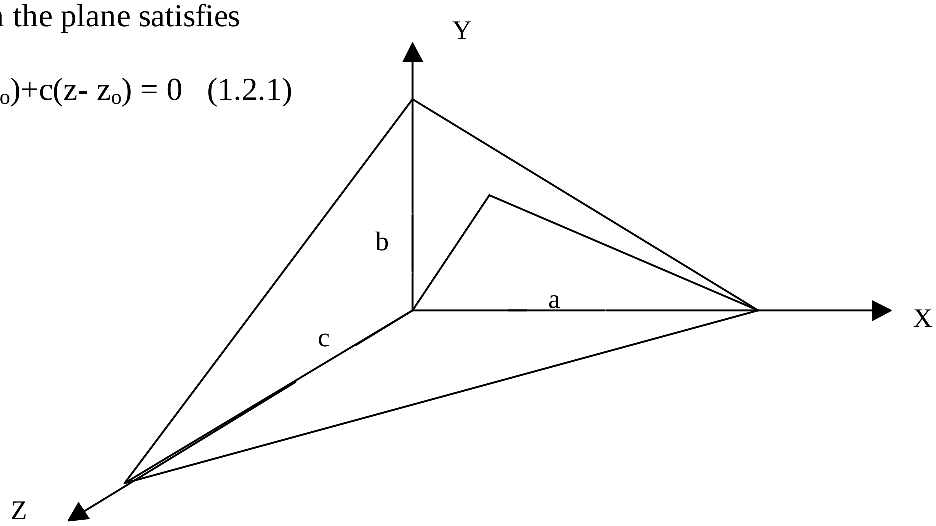

Figure 1.2 General Plane Geometry

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

$d_i = a(x_i - x_o)+b(y_i - y_o)+c(z_i - z_o)$ (1.2.2)

The sum of squares of distances of each point from the plane is

$$F = \sum_{i=1}^{n} d_i^2 \quad (1.2.3)$$

Hence our problem is to find the parameters (x_o ,y_o ,z_o) and (a,b,c) that minimizes the sum F.

1.3 Optimization

Consider a function

$$E(u) = \sum_{i=1}^{n} d_i^2(u) \quad (1.3.1)$$

which has to be minimized with respect to the parameters $u = (u_1 \dots u_n)^T$
Here in this case d_i represents the distance of the data point to the geometric element parameterized by u.

1.3.1 Linear Least Squares

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

Consider fitting a straight line

$$y = a + bx \quad (1.3.1.1)$$

through a set of data points (x_i, y_i), *i=1 to 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

$$F = \sum_{i=1}^{n} (y_i - a - bx_i)^2 \quad (1.3.1.2)$$

is the minimizing function. A necessary condition for f to be minimum is

$$\frac{\partial f}{\partial a} = 0 \text{ and } \frac{\partial f}{\partial b} = 0 \quad (1.3.1.3)$$

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

$$2(-1)\sum_{i=1}^{n}(y_i - a - bx_i) = 0$$
$$2(-x_i)\sum_{i=1}^{n}(y_i - a - bx_i) = 0 \qquad (1.3.1.4)$$

These can be simplified as

$$\sum_{i=1}^{n} y_i = na + b\sum_{i=1}^{n} x_i$$
$$\sum_{i=1}^{n} x_i y_i = a\sum_{i=1}^{n} x_i - b\sum_{i=1}^{n} x_i^2 \qquad (1.3.1.5)$$

The equations above can be solved simultaneously to give us the values for a and b.

1.3.2 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 point. So

$$\begin{aligned} y_1 &= a + bx_1 \\ y_2 &= a + bx_2 \\ y_3 &= a + bx_3 \\ &\vdots \\ y_n &= a + bx_n \end{aligned} \qquad (1.3.2.1)$$

It can be written in a matrix form

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \qquad (1.3.2.1)$$

Let $B = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$, $A = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}$, and $p = \begin{bmatrix} a \\ b \end{bmatrix}$ (1.3.2.2)

So it can be compacted as B=AP

The objective is to find a vector p that minimizes the Euclidean length of the difference

$$||B-AP||$$

if $P=P^* = \begin{bmatrix} a^* \\ b^* \end{bmatrix}$ is a minimize vector, $y = a^* + b^*x$ is a least square straight-line fit. This can be explained as

$$\|B-AP\|^2 = (y_1 - a - bx_1)^2 + (y_2 - a - bx_2)^2 + \ldots + (y_n - a - bx_n)^2 \qquad (1.3.3.3)$$

if let $d_1 = (y_1 - a - bx_1)^2$, $d_2 = (y_2 - a - bx_2)^2$, …, $d_n = (y_n - a - bx_n)^2$

d can be explained as the distance from a point of a data set to a fitting line. So

$$\|B-AP\|^2 = d_1^2 + d_2^2 + \cdots + d_n^2 \qquad (1.3.3.4)$$

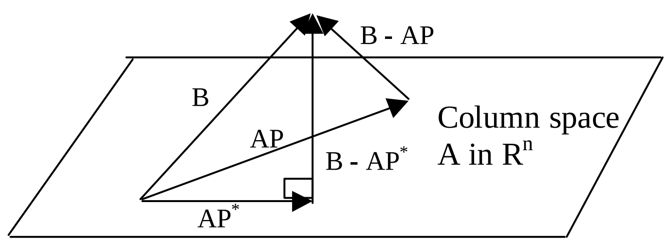

Figure 1.3.3 Finding Normal Equation

To minimize $\|B-AP\|$, AP must be equal to AP^* where AP^* is the orthogonal projection of B on the column space of A. This implies $B - AP^*$ must be orthogonal to the column space of A. So

$$(B - AP^*)AP = 0 \quad \text{for every vector P in } R^2$$

This can be written as

$$(AP)^T(B - AP^*) = 0$$

$$P^TA^T (B - AP^*) = 0$$

$$P^T (A^TB - A^TAP^*) = 0$$

$$(A^TB - A^TAP^*)P = 0$$

So $A^TB - A^TAP^*$ is orthogonal to every vector P in R^2. This implies

$$A^TB - A^TAP^* = 0$$

$$A^TAP = A^TB$$

Which implies that P^* satisfies the linear system

$$A^TAP = A^TB \qquad (1.3.3.5)$$

This equation is called as Normal equation. This will provide the solution for p as

$$P = (A^TA)^{-1}A^TB \qquad (1.3.3.6)$$

This equation can be used in the case of least square fit of a polynomial.

1.3.3 Eigen Vectors and Singular Value Decomposition

From equation (1.3.3.6), $(A^TA)^{-1}$ is very difficult to solve. So the alternative method using singular value decomposition is used to solve P

<u>Singular Value Decomposition</u>

A matrix can be decomposed in 3 matrices

$$A = USV^T \qquad (1.3.3.1)$$

Where U and V are orthogonal matrices, and S is a diagonal matrix containing the singular matrix of A

Place $A = USV^T$ into the normal equation

$$(USV^T)^T(USV^T)P = (USV^T)^TB$$

$$(VS^TU^TUSV^T)P = VS^TU^TB$$

knowing that

$$U^TU = I \qquad U^T = U^{-1} \qquad V^TV = I \qquad V^T = V^{-1}$$

So $$(VS^TSV^T)P = VS^TU^TB$$

Multiplying both sides by V^{-1}

$$(S^TSV^T)P = S^TU^TB$$

S is a diagonal matrix; therefore, $(SSV^T)P = SU^TB$

Multiplying both sides by S^{-1} 2 times

$$V^TP = S^{-1}U^TB$$

Again multiplying both sides by V^*

$$VV^TP = VS^{-1}U^TB$$

So the solution for P is

$$P=VS^{-1}U^TB \qquad (1.3.3.2)$$

1.3.4 Gauss-Newton Algorithm

Newton's method is one of the most powerful and well-known numerical methods for solving a root finding problem f(x)=0 where f(x) is a non-linear function. This technique is used in applications of linear least squares model of circle, sphere, cylinder, and cones. To motivate how such an algorithm works, first the Newton method is described.

Suppose that given a function f that its domain is [a,b] , and let $\bar{x} \in [a,b]$ such that $f'(\bar{x}) \neq 0$. The Taylor polynomial expansion for f(x) about $\bar{x}$ is

$$f(x) = f(\bar{x}) + (x - \bar{x})f'(\bar{x}) + \frac{(x - \bar{x})^2}{2} f''(x(x)) \qquad (1.3.4.1)$$

where $x(x) \in [x, \bar{x}]$.

Since f(q) = 0 where q=x, it gives

$$0 = f(\bar{x}) + (x - \bar{x})f'(\bar{x}) + \frac{(x - \bar{x})^2}{2} f''(x(x))$$

Newton's method is assumed as $|q-\bar{x}|$ is small. Therefore $(p-\bar{x})^2$ is much small. So

$$0 = f(\bar{x}) + (q - \bar{x})f'(\bar{x})$$

$$(q - \bar{x}) = -\frac{f(\bar{x})}{f'(x)}$$

Let $q = u_1$, $\bar{x} = u_0$, and $p = -\frac{f(\bar{x})}{f'(x)}$

Therefore $u_1 = u_0 + p$ and $p = -\frac{f(u_0)}{f'(u_0)}$ (1.3.4.2)

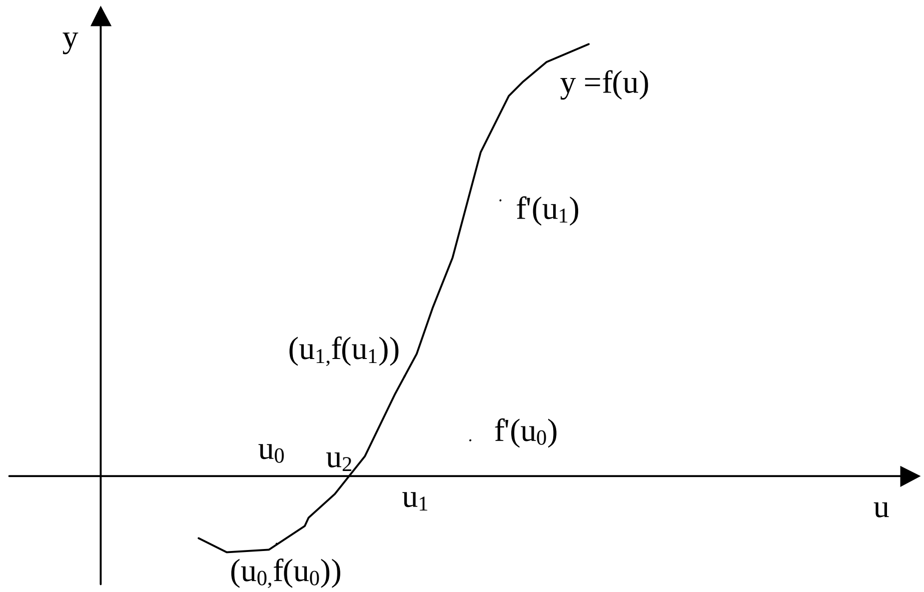

Figure 1.3.4 The iteration of Newton's method

From figure 1.3.4, the root finding problem y = f(u) can be done by following steps:

1) $f(u_0)$ and $f'(u_0)$ are evaluated
2) a tangent line to graph at the point $(u_0, f(u_0))$ is found. It cuts the u axis at u_1
3) find point u_1, and draw a tangent to graph at the point u_1 to find point u_2
4) these steps above are repeated until it is converged (close to u^*)

The main ingredients of the algorithm are

a) given an approximate solution, the problem is linearised
b) the linear version of the problem is solved
c) the solution of the linear problem is used to update estimate of the solution.

In the linear least square problem, the measurement of how well the geometric element fits a set of data point can be determined

$$E = \sum_{1}^{n} d_i^2 \qquad (1.3.4.2)$$

Where d_i is a distance from the point to the geometric element.

In the case, d_i is not linear function; the Gauss - Newton algorithm is used for finding the minimum of a sum of squares E. Assuming there is one initial estimate u^*, solve a linear least squares system of the form

$$Jp = -d \quad (1.3.4.3)$$

Where J is the m×n Jacobean matrix whose i^{th} row is the gradient of d_i with respect to the parameter u, i.e.,

$$J_{ij} = \frac{\partial d_i}{\partial u_j} \quad (1.3.4.4)$$

It's evaluated at u, and the i^{th} component of d is $d_i(u)$. The parameter is updated as

$$u := u + p \quad (1.3.4.5)$$

Converged conditions

Steps of Newton's algorithm are repeated until it reaches to a convergent point. Here 3 criteria are relevant for testing for convergence:

1) the change of E should be small
2) the size of the update, for instance $(p^T p)^{1/2}$, must be small
3) the partial derivation of E with respect to the optimization parameters, for instance $(gg^T)^{1/2}$ where $g=J^T d$, should be small.

2. Least Square Best Fit Line

Lines can be in a specified plane (2 Dimensional) or 3 dimensional. Since the 2 D line is a particular case of 3 dimensional line, the 3D line is discussed in detail. The procedure to fit a line to m data points(x_i, y_i, z_i), where $m \geq 2$, is given below.
Any point (x,y,z) on the line satisfies

$$(x,y,z) = (x_o, y_o, z_o) + t\,(a, b, c) \quad (2.1)$$

for some value of t.

It is known that the distance from a point to a line in 3 dimension as

$$d_i = \sqrt{[u_i^2 + v_i^2 + w_i^2]} \quad (2.2)$$

2.1 Parametrization

A line can be specified by

i) a point (x_o, y_o, z_o) on the line and
ii) the direction cosines (a, b, c).

2.2 Algorithm Description

The best-fit line passes through the centroid ($\bar{x}$, $\bar{y}$, $\bar{z}$) of the data and this specifies a point on L also the direction cosines have to be found out.

i)The first step is to find the average of the points x, y and z.

$\bar{x} = \Sigma x_{i,}/n$ (2.2.1)
$\bar{y} = \Sigma y_{i,}/n$
$\bar{z} = \Sigma z_{i,}/n$

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

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

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

3 Least Square Best Fit Plane

The procedure to fit a plane to m data points(x_i $y_{I,,}z_i$), where m≥3, is given below.
Any point (x,y,z) on the plane satisfies

$$a(x-x_o)+b(y-y_o)+c(z-z_o)=0. \quad (3.1)$$

It is known that the distance from a point (x_i $y_{i,}$ zi) to a plane specified by $x_{o,}$ $y_{o,,}z_o$,a, b and c is given by

$$d_i = a(x-x_o)+b(y-y_o)+c(z-z_o) \quad (3.2)$$

3.1 Parametrization

A plane can be specified by
iii) a point ($x_{o,}$ $y_{o,,}z_o$) on the plane and
iv) the direction cosines (a, b, c) of the normal to the plane

3.2 Algorithm Description

The best-fit plane passes through the centroid (xbar, ybar, zbar) of the data and this specifies a point on P also the direction cosines have to be found out.
For this, (a,b,c) is the eigen vector associated with the smallest eigen value of
$B = A^T A$

i)The first step is to find the average of the points x, y and z.

$\bar{x} = \Sigma x_{i,}/n$ (3.2.1)

$\bar{y} = \Sigma y_{i,}/n$

$\bar{z} = \Sigma z_{i,}/n$

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

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

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

4 Least Square Best Fit Circle

The procedure to fit a circle to m data points (x_i,y_i) , where $m \geq 3$, is given below.
Any point (x, y) on the circle satisfies

$$(x_i - x_o)^2 + (y_i - y_o)^2 = r^2 \quad (4.1)$$

It is known that the distance from a point $(x_i\ y_i)$ to a circle specified by x_o, y_o, and r and is given by

$$d_i = r_i - r \quad (4.2)$$

where $r_i = \sqrt{[(x_i - x_o)^2 + (y_i - y_o)^2]}$

and the elements of the Jacobean matrix J are found from the partial derivative of d_i with respect to the parameter x_o, y_o and r is given by

$$\frac{\partial di}{\partial xo} = -(xi - xo)/ri \quad (4.3)$$

$$\frac{\partial di}{\partial yo} = -(xi - xo)/ri$$

$$\frac{\partial di}{\partial r} = -1$$

4.1 Parameterization

A circle can be specified by
i) its center (x_o, y_o) and
ii) its radius r.

4.2 Algorithm Description

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

i) Minimization of F

$$F = \sum_{i=1}^{m} f_i^2 \qquad (4.2.1)$$

where

$$f_i = r_i^2 - r^2.$$

ii)This can be reduced to a linear system in x_0, y_0 and ρ as,

$$f_i = (x_i - x_0)^2 + (y_i - y_0)^2 - r^2,$$
$$= -2x_i x_0 - 2y_i y_0 + (x_0^2 + y_0^2 - r^2) + (x_i^2 + y_i^2), \qquad (4.2.2)$$

where ρ is

$$x_0^2 + y_0^2 - r^2.$$

iii)For minimizing F, the linear least squares system is solved

$$A\begin{bmatrix} x_0 \\ y_0 \\ r \end{bmatrix} = b \qquad (4.2.3)$$

where the elements of the ith row of A are the coefficients ($2x_i$,$2y_i$,-1) and the ith element of b is

$$x_i^2 + y_i^2.$$

iv) An estimate of r is obtained from the equation of ρ.

Once the initial estimates are obtained the right hand side vector d and Jacobean matrix J are formed.

Then the linear least-squares system is solved

$$J\begin{bmatrix} p_{x_0} \\ p_{y_0} \\ p_r \end{bmatrix} = -d;$$

The values of the parameters are updated according to the estimates

$$x_0 := x_0 + p_{x_0},$$
$$y_0 := y_0 + p_{y_0},$$
$$r := r + p_r.$$

The above steps are repeated until the algorithm converges.

5 Least Squares Sphere

5.1 Parametrization

A sphere is specified by its center (x_o, y_o, z_o) and radius r_o . Any point on the sphere satisfies the equation $(x - x_o)^2 + (y - y_o)^2 + (z - z_o)^2 = r^2$.

5.2 Initial estimates for center and radius

Choice of a minimizing function
A minimizing function has to be identified to obtain an initial estimate for the center and radius. Consider the function $f1 = r_i - r$ where $r_i = \sqrt{(x - x_o)^2 + (y - y_o)^2 + (z - z_o)^2}$.
Differentiating this function with respect to x_o, y_o, z_o and r_o will result in complicated equations which are difficult to solve.
Therefore, consider the function $f2 = r_i^2 - r^2$. This function can be written as $f2 = (r_i - r)(r_i + r) \approx 2r(r_i - r)$, since $r_i + r$ can be approximated as $2r$.
Differentiating this function with respect to x_o, y_o, z_o and r_o to obtain initial estimates center and radius.
Thus the **minimizing function** to obtain initial estimates for a sphere is $f = r_i^2 - r^2$.

Initial estimate math
Expanding $f = r_i^2 - r^2$, we get
$$f = (x_i - x_o)^2 + (y_i - y_o)^2 + (z_i - z_o)^2 - r^2 = -(2x_i x_o + 2y_i y_o + 2z_i z_o) + \boldsymbol{r} + (x_i^2 + y_i^2 + z_i^2)$$
where $\boldsymbol{r} = (x_o^2 + y_o^2 + z_o^2) - r^2$. The variable $\boldsymbol{r}$ is introduced to make the equation linear.

The above set of equations for *n* set of data points are now represented in matrix form.

$$\begin{pmatrix} -2x_1 & -2y_1 & -2z_1 & 1 \\ -2x_2 & -2y_2 & -2z_2 & 1 \\ \cdots & \cdots & \cdots & \cdots \\ -2x_n & -2y_n & -2z_n & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ y_o \\ z_o \\ r \end{pmatrix} - \begin{pmatrix} x_1^2 + y_1^2 + z_1^2 \\ x_2^2 + y_2^2 + z_2^2 \\ \cdots \\ x_n^2 + y_n^2 + z_n^2 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \\ f_n \end{pmatrix}$$

For a least square solution, $f_i = 0$. Introduce matrix notation,

$$A = \begin{pmatrix} -2x_1 & -2y_1 & -2z_1 & 1 \\ -2x_2 & -2y_2 & -2z_2 & 1 \\ \cdots & \cdots & \cdots & \cdots \\ -2x_n & -2y_n & -2z_n & 1 \end{pmatrix}, \; P = \begin{pmatrix} x_0 \\ y_o \\ z_o \\ r \end{pmatrix} \text{ and } B = \begin{pmatrix} x_1^2 + y_1^2 + z_1^2 \\ x_2^2 + y_2^2 + z_2^2 \\ \cdots \\ x_n^2 + y_n^2 + z_n^2 \end{pmatrix}.$$

We have *AP - B=0*. Solve this equation in least square sense to obtain P. This means that P satisfies the equation $A^T AP = A^T B$. The initial estimates for x_o, y_o, z_o and r are obtained from the above solution for P. The initial estimate for the radius *r* can be obtained from the relation $r = (x_o^2 + y_o^2 + z_o^2) - r^2$.

5.3 Gauss Newton method

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.

Minimizing function

The minimizing function is given by $d_i = r_i - r$

where $r_i = \sqrt{(x - x_o)^2 + (y - y_o)^2 + (z - z_o)^2}$

i) Building the jacobian matrix

The elements of the Jacobian matrix are given by
$$\begin{pmatrix} \frac{\partial d_1}{\partial x_o} & \frac{\partial d_1}{\partial y_o} & \frac{\partial d_1}{\partial z_o} & \frac{\partial d_1}{\partial r_o} \\ \frac{\partial d_2}{\partial x_o} & \frac{\partial d_2}{\partial y_o} & \frac{\partial d_2}{\partial z_o} & \frac{\partial d_2}{\partial r_o} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial d_n}{\partial x_o} & \frac{\partial d_n}{\partial y_o} & \frac{\partial d_n}{\partial z_o} & \frac{\partial d_n}{\partial r_o} \end{pmatrix}$$

Evaluating various components of the Jacobian and substituting in the matrix, we get

$$J = \begin{pmatrix} \frac{-(x_1 - x_0)}{r_1} & \frac{-(y_1 - y_0)}{r_1} & \frac{-(z_1 - z_0)}{r_1} & -1 \\ \frac{-(x_2 - x_0)}{r_2} & \frac{-(y_2 - y_0)}{r_{21}} & \frac{-(z_2 - z_0)}{r_2} & -1 \\ \\ \frac{-(x_n - x_0)}{r_n} & \frac{-(y_n - y_0)}{r_n} & \frac{-(z_n - z_0)}{r_n} & -1 \end{pmatrix}$$

ii) Solve the linear least square system $J\,P = -d$

where $P = \begin{pmatrix} p_{x_o} \\ p_{y_o} \\ p_{z_o} \\ p_r \end{pmatrix}$

iii) Increment parameters according to

$x_o = x_o + p_{x_o}$;

$y_o = y_o + p_{y_o}$;

$z_o = z_o + p_{z_o}$;

$r_o = r_o + p_r$;

iv) Convergence condition

Repeat steps till algorithm has converged. The convergence condition is given by $g = J^T d$ is minimum.

6 Gauss-Newton Strategy for Cylinders

Any line can be specified by giving a point on the line and direction cosine (a,b,c). So it requires 6 numbers to describe a line. Its constraint is $a^2 + b^2 + c^2 = 1$. So given two of components, the third can be determined. Consider a constraint on a line where c = 1, a vertical line, it is enough to specify two direction cosine a and b. Z can be determined from the relationship

$$ax + by + cz = 0 \tag{5.1}$$

Since c = 1, z = -ax -by (5.2)

So the advantage of this constraint is to minimize 6 parameters to 4 parameters a, b, x_0, and y_0. It also reduces the complication of solve Jacobean matrix and time to evaluate because the derivatives of distance is computed when using a Gauss-Newton method.

The Gauss-Newton algorithm is modified for the case of a cylinder. It is

1) Translate the coordinate system at beginning of each iteration, so the point on the axis is the origin of the coordinate system. It means that x = y =0.
2) Rotate the coordinate system so that the direction of the axis is along the z-axis. So a=b=0 & c=1.

6.1 Rotational Concept

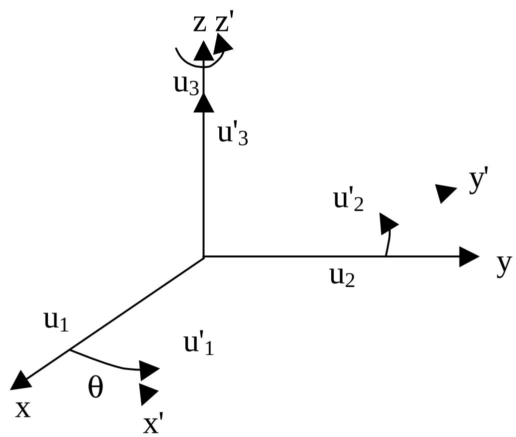

Figure 5.1 Example of how to develop rotational matrices

Consider a figure above, B={u_1, u_2, u_3} is rotated to the new basis B'={u'_1, u'_2, u'_3}, where u_1, u_2, u_3 and u'_1, u'_2, u'_3 are unit vectors. u'_1, u'_2 , and u'3 can be expressed as

$$[u'_1]_B = \begin{bmatrix} \cos(q) \\ \sin(q) \\ 0 \end{bmatrix} \qquad [u'_2]_B = \begin{bmatrix} -\sin(q) \\ \cos(q) \\ 0 \end{bmatrix} \qquad [u'_3]_B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad (5.1.1)$$

So, the transition matrix from B' to B is

$$P = \begin{bmatrix} \cos(q) & -\sin(q) & 0 \\ \sin(q) & \cos(q) & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (5.1.2)$$

and the transition matrix from B to B' is

$$P^T = \begin{bmatrix} \cos(q) & \sin(q) & 0 \\ -\sin(q) & \cos(q) & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (5.1.3)$$

So the new coordinates (x',y',z') can be computed from its old coordinates (x,y,z) by

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos(q) & \sin(q) & 0 \\ -\sin(q) & \cos(q) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (5.1.4)$$

7 Least Squares Best Fit Cylinders

Assume that a cylinder is fitted to m points (x_i, y_i, z_i), where $m \geq 5$

7.1 Parametrization

A cylinder can be specified by

1) a point (x_0, y_0, z_0) on its axis
2) a vector (a,b,c) pointing a long the axis and
3) its radius r

7.2 Initial estimate for a cylinder

Let a, b, c be the direction cosine of the axis. x_0, y_0 , and z_0 are a point on the axis, and x_i, y_i, z_i are any point on the cylinder. Then

$$A^2 + B^2 + C^2 = R^2 \quad (6.2.1)$$

Where $A = c(y_i - y_0) - b(z_i - z_0)$
$B = a(z_i - z_0) - c(x_i - x_0)$
$C = b(x_i - x_0) - a(y_i - y_0)$
R = initial estimate for radius.

Substitute into the equation above

$$[c(y_i - y_0) - b(z_i - z_0)]^2 + [a(z_i - z_0) - c(x_i - x_0)]^2 + [b(x_i - x_0) - a(y_i - y_0)]^2 = R^2$$

This equation is simplified; it yields

$$Ax^2 + By^2 + Cz^2 + Dxy + Exz + Fyz + Gx + Hy + Iz + J = 0 \quad (6.2.2)$$

Where $A = (b^2 + c^2)$
$B = (a^2 + c^2)$
$C = (a^2 + b^2)$
$D = -2ab$
$E = -2ac$
$F = -2bc$

$$G = -2(b^2 + c^2)x_0 + 2aby_0 + 2acz_0$$
$$H = 2abx_0 - 2(a^2 + c^2)x_0 + 2bcz_0$$
$$I = 2acx_0 + 2bcy_0 - 2(a^2 + b^2)z_0$$
$$J = (b^2 + c^2)x_0^2 + (a^2 + c^2)\ y_0^2 + (a^2 + b^2)\ z_0^2 - 2bcy_0z_0 - 2acz_0x_0 - 2abx_0y_0 - R^2$$

Divide both sides by A

$$x^2 + \frac{B}{A}y^2 + \frac{C}{A}z^2 + \frac{D}{A}xy + \frac{E}{A}xz + \frac{F}{A}yz + \frac{G}{A}x + \frac{H}{A}y + \frac{I}{A}z + \frac{J}{A} = 0$$

So

$$\frac{B}{A}y^2 + \frac{C}{A}z^2 + \frac{D}{A}xy + \frac{E}{A}xz + \frac{F}{A}yz + \frac{G}{A}x + \frac{H}{A}y + \frac{I}{A}z + \frac{J}{A} = -x^2 \qquad (6.2.3)$$

This can be written as a linear system

$$\begin{pmatrix} y_1^2 & z_1^2 & x_1y_1 & x_1z_1 & y_1z_1 & x_1 & y_1 & z_1 & 1 \\ y_2^2 & z_2^2 & x_2y_2 & x_2z_2 & y_2z_2 & x_2 & y_2 & z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_n^2 & z_n^2 & x_ny_n & x_nz_n & y_nz_n & x_n & y_n & z_n & 1 \end{pmatrix} \begin{pmatrix} B/A \\ C/A \\ D/A \\ E/A \\ F/A \\ G/A \\ H/A \\ I/A \\ J/A \end{pmatrix} = \begin{pmatrix} -x_1^2 \\ -x_2^2 \\ \vdots \\ -x_n^2 \end{pmatrix} \qquad (6.2.4)$$

This is the form of $A^TAP = A^TB$, and P can be solved as

$$P=(A^TA)^{-1}A^TB \qquad (6.2.5)$$

Let C'=C/A, D'=D/A, E'=E/A, F'=F/A, G'=G/A, H'=H/A, I'=I/A, and J'=J/A.

Solving for this; if |D'|, |E'|, and |F'| are close to 0, then B' close to 1 implies (a b c) = (0 0 1), and C' close to 1 implies (a b c) =(0 1 0). Otherwise

$$k = \frac{2}{(1 + B' + C')} \qquad (6.2.6)$$

A =k, B=kB', C=kC', D=kD', E=kE', F=kF', G=kG', I=kI', and J=kJ' (6.2.7)

If A and B are close to 1

$c'=(1-C)^{1/2}$, a'=E/-2C', and b' = F/-2c'

If A is close to 1, B is not close to 1

$b'=(1-B)^{1/2}$, $a'=D/-2b'$, and $c'=F/-2b'$

If A is not close to 1

$a' = (1-A)^{1/2}$, $b' = D/-2a'$, and $c'=E/-2a'$

The direction (a' b' c') is normalized to get the direction (a b c)

7.2.1 Initial estimate for the point on axis

Knowing (a b c), the definition of the coefficients G, H, I and equation $ax_0 + by_0 + cz_0 = 0$ are used to derive an estimate for (x_0,y_0,z_0). These will form into the linear system as

$$\begin{pmatrix} -2(b^2+c^2) & 2ab & 2ac \\ 2ab & -2(a^2+c^2) & 2bc \\ 2ac & 2bc & -2(a^2+b^2) \\ a & b & c \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} = \begin{pmatrix} G \\ H \\ I \\ 0 \end{pmatrix} \qquad (6.2.1.1)$$

This equation is solved by using the normal equation $A^TAP = A^TB$, where P contains the initial estimate for the point in the axis.

7.2.2 Initial Estimate for radius

This can be done by using the definition of J that is

$$J = (b^2 + c^2)x_0^2 + (a^2 + c^2)\ y_0^2 + (a^2 + b^2)\ z_0^2 - 2bcy_0z_0 - 2acz_0x_0 - 2abx_0y_0 - R^2$$

So

$$R^2 = (b^2 + c^2)x_0^2 + (a^2 + c^2)\ y_0^2 + (a^2 + b^2)\ z_0^2 - 2bcy_0z_0 - 2acz_0x_0 - 2abx_0y_0 - J \qquad (6.2.2.1)$$

7.3 Algorithm description

1. Translate of the origin
   Translate (a copy of) the data so that the point on the axis lies at centroid of the point

$$(x_i, y_i, z_i) = (x_i, y_i, z_i) - (x_c, y_c, z_c) \qquad (6.3.1)$$

2. Transform the data by a rotation matrix U which rotates (a b c) to a point the z-axis

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = U \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} \qquad (6.3.2)$$

   where U is

$$U = \begin{pmatrix} c_2 & 0 & s_2 \\ 0 & 1 & 0 \\ -c_2 & 0 & c_2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_1 & s_1 \\ 0 & -s_1 & c_1 \end{pmatrix} \qquad (6.3.3)$$

for the case (a b c) =(1 0 0)

$s_1 = 0$, $c_1 = 1$, $s_2 = -1$, $c_2 = 0$.

Otherwise

$$c_1 = c/\sqrt{b^2 + c^2} \qquad s_1 = -b/\sqrt{b^2 + c^2}$$

$$c_2 = (cc_1 - bs_1)/\sqrt{a^2 + (cc_1 - bs_1)^2}$$

$$s_2 = -a/\sqrt{a^2 + (cc_1 - bs_1)^2}$$

To rotate the axis of the cylinder, the axis of the cylinder is rotated about x-axis so that it is now in the YZ plane. Then the axis of the cylinder is rotated about the y-axis so that it is now along the z-direction.

3. Form the right side vector d and Jacobian matrix

The distance from a point (x_i, y_i, z_i) to the cylinder is found from

$$d = r_i - r \qquad (6.3.4)$$

where

$$r_i = \frac{\sqrt{u_i^2 + v_i^2 + w_i^2}}{\sqrt{a^2 + b^2 + c^2}}$$

$u_i = c(y_i - y_0) - b(z_i - z_0)$

$v_i = a(z_i - z_0) - c(x_i - x_0)$

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

Jacobian matrix is

$$J = \begin{pmatrix} -x_1/r_1 & -y_1/r_1 & -x_1 z_1/r_1 & -y_1 z_1/r_1 & -1 \\ -x_2/r_2 & -y_2/r_2 & -x_2 z_2/r_2 & -y_2 z_2/r_2 & -1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -x_n/r_n & -y_n/r_n & -x_n z_n/r_n & -y_n z_n/r_n & -1 \end{pmatrix} \qquad (6.3.5)$$

4. Solve the linear least-squares system

$$J\begin{pmatrix} p_{x0} \\ p_{y0} \\ p_a \\ p_b \\ p_r \end{pmatrix} = -d \tag{6.3.6}$$

5. update the parameter estimates according to

$$\begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} + U^T \begin{pmatrix} p_{x0} \\ p_{y0} \\ -p_{x0}p_a - p_{y0}p_b \end{pmatrix} \tag{6.3.7}$$

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = U^T \begin{pmatrix} p_a \\ p_b \\ 1 \end{pmatrix} \tag{6.3.8}$$

$$r = r + p_r \tag{6.3.9}$$

Steps above are repeated until the algorithm has converged. In step I, (a copy of) the original data set is used rather than a transformed set from a previous iteration.

6. If we want to present (x_0, y_0, z_0) on the line nearest the origin, then

$$\begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} - \frac{ax_0 + by_0 + cz_0}{a^2 + b^2 + c^2} \begin{pmatrix} a \\ b \\ c \end{pmatrix} \tag{6.3.10}$$

8 Conclusion

In this report, the least square fitting algorithms for lines, planes, circles, spheres and cylinders were discussed in detail. They the implementation of the algorithm using MATLAB and the source code is included in the appendix. The mathematical steps involved in the implementation are also discussed in detail. The summaries of the articles that were used as reference for doing this project were also included as appendix.

9 References

1. Forbes, Alistair B., Least Squares Best-Fit geometric elements, NPL report April 1989.
2. Assessment of Position, size and departure from nominal form of geometric features, British Standards Institution, BS7172: 1989
3. Computational Metrology, Manufacturing Review, Vol 6, No 4,1993
4. Press, William H. Flannery, Brian P, Teukolsy, Saul A, Vetterling, William T., Numerical Recipes in C, The aart of Scientific computing, Cambridge university Press.
5. Ramakrishna Machireddy, Sampling techniques for plane geometry for coordiante measuring machines, MS Thesis, UNCC 1992
6. Uppliappan Babu, Sampling Methods and substitute Geometry for Measuring cylinders in a Co-ordiante Measuring Machines(Chapter 2), MS Thesis ,UNCC 1993
7. Anton Rorres, Element Linear Algebra Application Version, NY 1991.