THE UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

METHODS FOR DETERMINATION OF THE CONICITY TOLERANCE

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for

degree of

MASTER OF SCIENCE

By

SUNG HUN CHUNG

Norman, Oklahoma

1997

METHODS FOR DETERMINATION
OF
THE CONICITY TOLERANCE

A THESIS APPROVED FOR
THE SCHOOL OF INDUSTRIAL ENGINEERING

BY [redacted]

Dr. Shivakumar Raman, chair

[redacted]

Dr. F. Hank Grant

[redacted]

Dr. Craig M. Harvey

ACKNOWLEDGEMENT

There is no word to express my respect to my father of knowledge Dr. Shivakumar Raman. At the fourth semester in University of Oklahoma, I was so depressed that I could not keep studying in the United States. He kept encouraging me to study and showed me the new world of technology. He saved my entire life, made my dream come true, and I should show my appreciation to his wife and his children. I will remember his help forever. I wish to express my gratitude to Dr. Hank Grant and Dr. Craig M. Harvey for their guidance in the completion of this thesis.

In addition, I should express my appreciation to Dr. Simin Pulat and Dr. A. Ravindran for their interest in my thesis.

I am greatly indebted to my senior Dr. Jae Soo Lee and his wife for their help during the hardest time of my life. I would like to thank my friends Sung Hwan Joo, Sung Hyun Hwang, Nu Yi Kong, Yee Sun Kwon, my senior Jae Jin Seok and Dr. Sang Tae Kim who is a professor in Yeung Nam University, Korea.

I also wish to thank my father Moo Jo Chung and my mother Sun Hee Kang for their endless love, understanding and sacrifice. I love you, my father and my mother.

TABLE OF CONTENTS

ABSTRACT

This thesis attempts, for the first time, to develop comprehensive equations for determination of the conicity tolerance. No such attempts are evident in the current literature in conicity analysis. This thesis develops linear and nonlinear equations for the representation of a cone. For the minimum zone estimation of the conicity tolerance, linear and non-linear optimization methods are developed. Hence, the fundamental contribution of this work is in the mathematical development for cones. These methods are compared against the popular Least Squares Method (respectively linear and nonlinear formulation). Initial experiments are conducted and a preliminary analysis of results is presented. It was observed in these pilot studies that the Least Squares Method does not yield the minimum zone in the linear as well as the nonlinear cases. The form tolerance using the Minimum Zone Method is approximately 13 % less than the form tolerance using the Least Squares Method. Further, the linear formulation may provide a good approximation and be used as an alternative to the general nonlinear formulation.

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

X_0: The initial movement of a center axis in X direction

Y_0: The initial movement of a center axis in Y direction

Z_0: The initial movement of a center axis in Z direction

R_0: The initial radius of a circle, a cylinder and a cone.

l_0: The slope (changing rate) of X coordinate according to the increase of Z ($l_0 = \frac{\Delta X_i}{\Delta Z_i}$)

m_0: The slope (changing rate) of Y coordinate according to the increase of Z ($m_0 = \frac{\Delta Y_i}{\Delta Z_i}$)

S: The slope of the hypotenuse of a cone on the X-Z plane (the increase of R coordinate according to the increase of Z coordinate, $S = \frac{\Delta R_i}{\Delta Z_i}$)

A: An unknown constant in the nonlinear equation for a cone ($A = \left(\frac{c}{b}\right)^2$)

R_i, θ_i, Z_i: Coordinates of points on the surface of a substitute cone in the Cylindrical Coordinate System

r_i, θ_i, z_i: Coordinates of the measured points in the Cylindrical Coordinate System

X, Y, Z: The Coordinates of a point on the surface of a substitute cone in the Cartesian Coordinate System

X_i Y_i Z_i: The Coordinates of measured points in the Cartesian Coordinate System

e_i: The deviation between the measured points and calculated points on the surface of the substitute cone at the same θ_i and z_i

a, b, c: The constants of an asymptotic cone

(a_1, b_1): Center of Chetwynd's circle in the Cartesian Coordinate System

(E, ϕ): Center of Chetwynd's circle in the Cylindrical Coordinate System

R_s: The radius from the center axis of a substitute cone to points on the surface of a substitute cone ($R_s = R_0 + S\ Z_i$)

Td: The radial deviation between the innermost cone and the outermost cone

hTd: the radial deviation between the outermost point and the substitute cone (Td/2)

h_x: The half of the allowance of coaxility in the drawing

ε: Eccentricity ratio

α: The vertex angle of the substitute cone ($\alpha = 2\theta'$)

α_{max}: The maximum angle of the vertex

α_{min}: The minimum angle of the vertex

θ: The angle between the center axis and the X axis

θ': The half of the vertex angle

θ'': $\theta - 90^{\circ}$

$\vec{V}_x$: A vector of the radius

$\vec{V}_c$: A vector of the center axis

$\vec{V}_n$: A vector of the hypotenuse of the cone. where, the hypotenuse represents a line connected from the vertex of the cone to the a point on the circumference of the bottom of the cone

(X_c, Y_c): Coordinates of any point on the center axis of the tilted cone in the Cartesian Coordinate System

CHAPTER 1

INTRODUCTION

Coordinate Measuring Machines (CMMs) are employed for checking conformance of form, orientation, profile and location specifications of part features. They typically employ canned procedures for estimating the least squares fit zone of the measured points. It has been proved over the last decade and half that the Least Squares Method does not always result in a minimum zone in Shunmugam (1986), (1987), Lin et al (1995), Murthy et al (1979), Suen (1996), Kanada (1993) and Huang (1993). Accordingly, linear and nonlinear optimization methods have been used by various researchers to develop minimum zones. However, the majority of works have dealt only with flatness, straightness, circularity and cylindricity tolerances. This is partly due to the reason that most standards define only these tolerances with reference to inspection. To date, no attempts are evident in the current literature in the development of minimum zones for inspection of cones and other complex (three dimensional) geometrical shapes. The appropriate equations for the cone system are more complex when compared with those for circles and cylinders. However, the inspection of many industrial parts such as tapered cylinders, frustrum holes and bevel gears require an accurate assessment of the conicity tolerance.

This thesis has studied the conicity tolerance for right cone, moved cone and tilted cone and develops the relevant mathematical equation. The last-mentioned cone is particularly important as it is most commonly employed within many CMMs. Importantly, our work concentrates on the development of mathematical relationships for the linear as well as non-linear Least Squares Method for evaluating conicity. Further, linear and non-linear optimization procedures are developed, using standard software, to evaluate the minimum tolerance zone for cones. The Least Squares Method is then compared against the minimum zones obtained through the Simplex Method and the non-linear optimization methods. Each of these models are validated and compared using real data obtained using a Browne and Sharpe Coordinate Measuring Machine (Model: PFx 454 System). Overall, a comprehensive equations for conicity analysis is developed using several methods. Preliminary observations are made and known trends verified using pilot data.

Chapter 2 provides a short review of the related literature and Chapter 3 gives equations relevant to cones. General problems of tolerances in cones are highlighted in Chapter 4. The Least Squares Method and Minimum Zone Method for cones using linear equations are discussed in Chapter 5 while nonlinear optimization for cones is elaborated in Chapter 6. Concluding remarks are made in Chapter 7.

CHAPTER 2

LITERATURE REVIEW

It is to be noted that many of the papers written on CMM inspection do not verify their algorithms with data obtained using an actual CMM in Shunmugam (1986), (1987), Lin et al (1995), Murthy and Abdin (1979), Suen (1996), Huang (1993), Roy et al (1992) and (1995). All the same, Significant work has been done in the development of algorithms for inspection using CMMs. The research may be categorized into two areas: sampling point selection and minimum zone estimation. The sampling point problem deals with the selection of points for inspection such that representative data to verify flatness, straightness, or roundness is obtained. Among the common techniques employed are uniform sampling, pseudo-random sampling, the Hammersly Method, and the Halton-Zaremba Method. These issues are described in Woo et al. (1995) and Lee et al. (1997).

The Least Squares Method is most common in CMM inspection for data fitting. Minimum Zone Estimation has arisen due to the inaccuracies caused by the Least Squares Method. Of the techniques employed in the literature, the most common are linear and non-linear modeling.

This research deals with the second area, namely the minimum zone problem. Further, most minimum zone methods have been developed for representing straightness (the Median Technique, the MinMax Method, the MinAvg Method), flatness (the Median Technique, the Simplex Search Method, the MinMax Method), roundness or circularity (the Simplex Method using limacon approximation, the Median Technique, the Simplex Search Method) and cylindricity (the Simplex Search Method, the MinMax Method, the MinAvg Method). No attempts are evident in the development of methods for estimating the conicity allowance, although Kanada and Suzuki (1993) illustrated minimum zone evaluation in spheres. In fact, the mathematical extension of the studied geometries to cones is non-trivial. An extensive presentation of all articles is avoided, and for brevity only the most relevant works for the understanding the Minimum Zone Method are presented here.

Chetwynd (1985*)* introduced a limacon-approximated equation for circles. He analyzed Minimum Radius Circumscribing Limacons (circles), Minimum Zone Limacons (circles), and Minimum Zone Straight Lines and Planes. For CMMs, he performed minimum zone analysis for circles using limacon approximation. He approximated the nonlinear equation of circles.

$r = E\,\cos(\theta - \phi) + [R^2 - E^2 \sin^2(\theta - \phi)]^2$ (True circle)--------------------------[1]

into a linear equation such as:

$\cong a\cos\theta + b\sin\theta + R$ (limacon circle)---[2]

where $E, \varnothing$ are coordinates in the Polar Coordinate System, and a and b are coordinates in the Cartesian Coordinate System.

He compared the accuracy of limacon circles against true circles in this article. He said that because roundness measuring instruments usually introduce a slight geometrical distortion of the data, fitting a limacon, equation [2], nearly always gives a more accurate measurement than would fitting a true circle. His opinion is used in this thesis. Although he introduced the Minimum Zone Method using a constrained linear optimization technique for a circle, he did not test his algorithm using real data to prove his theory. However, several other articles in this area are based on the limacon equation for representing three dimensional circular objects such as cylinder and sphere.

Shunmugam (1986) used equations for lines, planes and circles, similar to Chetwynd's equations for lines, planes and circles. Further, he extended Chetwynd's linear equation for circles into a linear equation for a cylinder. He also presented an equation for a sphere based on the Sphere Coordinate System. Based on these equations for lines, planes, circles, cylinders and spheres, he used a Median Technique to evaluate the minimum tolerance zone for each shape. He mentions that the Median Technique considers only the extreme points, namely the crest and valley points. He also illustrated the use of matrices to find unknown constants using the Least Squares Method for various objects.

In an extension, Shunmugam (1987) compared linear deviation expressed as linear equations against normal deviation expressed as nonlinear equations. When he analyzed the normal deviation, he used the Simplex Search Technique for nonlinear optimization. The Simplex Search Technique (Reklaitis,1983) is different from the Simplex Method in linear optimization. The Simplex Search Technique is a nonlinear optimization technique which only requires function evaluations.

Lin et al (1995) used a Minimum Max-Deviation Method (MinMax), a Minimum Average Deviation Method (MinAvg) and the Convex Hull Method to find minimum form tolerances for lines, circles, planes and cylinders, compared to the Least Squares Method. They used nonlinear equations to express normal deviations from substitute elements using the MinMax algorithm, and linear equations to express norminal deviations using the MinAverage algorithm. They stated that the tolerance using the MinAverage algorithm is less than the tolerance using the MinMax algorithm due to limacon approximation used in the equations for MinAverage Algorithm. However, the comparison is not properly controlled. To solve the MinMax and MinAverage problems, the IDESIGN software was used to solve the linear and nonlinear optimization problems.

Murthy and Abdin (1979) used normal deviations to express form tolerances for lines, planes, circles and spheres. They also used nonlinear equations and nonlinear optimization techniques such as the Monte Carlo Technique, Simplex Search Technique, and Spiral Search Technique. These techniques are based on function evaluations rather than gradient based techniques. He emphasized the fact that the normal Least Squares Method does not always lead to minimum zone deviation.

Suen (1996) used an interval bias adaptive linear neural network structure together with a least mean squares learning algorithm to obtain an accurate algorithm for a minimum zone circle. However, they still used a linear equation developed by Chetwynd (1985) for a circle.

Orady et al (1996) developed an algorithm of the Minimum Zone Method for lines using nonlinear optimization and called it the Nonlinear Optimization Method (NOM). He used a data filtering algorithm to remove the outermost points before using nonlinear optimization. The results were compared to NOM with the results from the CMM, the Convex Hull Method, and Nonlinear Least Squares Method. Since the outermost measured points were removed through the data filtering algorithm, it is expected that the results are less than the outputs of the Least Squares Method.

Kanada and Suzuki (1993) presented a nonlinear equation for planes. The equation represented the errors between the substitute plane and the measured points in normal direction from the plane. They formulated a MinMax problem to minimize the difference of the maximum and minimum errors normal to the plane. To optimize the nonlinear equation, they used two direction search methods: the Downhill Simplex

Method and Repetitive Bracketing Method. These methods only require function evaluations to find optimal objective function values. They found that the Downhill Simplex Method is better than the Repetitive Bracketing Method from the viewpoint of the number of iterative calculations.

Since the Simplex Method is only a basic algorithm for nonlinear optimization, the results need not be the best results for precision measurement. Himmelblau said that the Davidon-Fletcher-Powell Method (Quasi Newton Method), Broyden-Fletcher-Shanno Method, and Powell Direct-Search Method are the superior methods in the sense of robustness (Reklaitis, 1983).

Kanada and Suzuki (1993), in an explanation of their work, used the Nelder-Mead Simplex Search Methods (Reklaitis, 1983), the Bracketing Method with quadratic interpolation, the Bracketing Method with golden section, and a new method called TKM. While presenting their new method, they used an assumption that the extreme points from the optimization using the Least Squares Method are the same extreme points as the optimization using the Minimum Zone Method. However, this assumption seems different based on the results of this thesis.

Kanada (1994) has also applied a Downhill Simplex Search Technique to a sphere expressed as a nonlinear equation. To evaluate a form tolerance, sphericity, he used Shunmugam's linear equation to formulate the Least Squares Method and a nonlinear equation to formulate the Minimum Zone Method. The nonlinear equation for spheres expresses the sphere moved in the X, Y, and Z directions. To find the three unknown constants in this nonlinear equation, he used the Downhill Simplex Technique, using system origin as the initial approximate values.

Huang (1993) developed the Control Line Rotation Schema for straightness analysis. His algorithm was a geometrical approach, the so-called 2-1 Modal. His algorithm found two control points and one control point to make two control lines. Using two control lines, he calculated the straightness. His algorithm was applied to evaluation of lines. However, it has a limitation of extension into the three dimensions.

Roy and Xhang (1992) used a computational-geometry-based method, the Voronoi Diagram, for measuring circularity. Their algorithm is a geometrical approach rather than an optimization technique. They compared the circularity using the Least Squares Method with the circularity using the Voronoi diagram.

In a later paper, Roy and Xhang (1995) provide definitions of form tolerances such as straightness, flatness, roundness and cylindricity. To estimate cylindricity, they used sections of a cylinder normal to the CMM's local Z axis rather than a continuous equation that expresses the cylinder. This could lead to a problem in that the center of each circle need not lie on a unique center line. To solve this controversy, they used a line as a center axis which consists of center points of the circle, using the Least

Squares Method. However, the center axis of the cylinder estimated through the Least Squares Method need not be the same as the center axis of the cylinder analyzed through the Minimum Zone Method. Also, no results from experiments or simulation were represented.

Wang (1992) analyzed straightness, flatness, roundness and cylindricity using nonlinear equations with sample points. To measure the minimum zone for those tolerances, he used a constrained optimization technique called the Fast Feasible Direction Method. More specifically, he used the NCONG of IMSL developed by the Fast Feasible Direction Method for this analysis. One of problems in his article is the equation of a circle. His equation for a circle can not be tilted. This means that he violates the ISO standard, and his equation for a circle can not generate the tilted vector of the center axis to calculate the intersecting angle between the center axis of the other objects.

These articles clearly point to the necessity of developing sturdy methods for conicity inspection. And yet, several industrial shapes are cones, internal or external. The present research tends to borrow important ideas, from the general area of minimum zone estimation as well as least square approximation and applies them for conicity analysis. In this aspect, the articles surveyed in this chapter have undoubtedly influenced the research contained in this thesis.

CHAPTER 3

EQUATIONS RELEVANT TO CONES

Two equations for a cone, one linear and the other nonlinear, are introduced in this chapter. The first equation is a linearized equation of the cone, and the second equation is the general nonlinear equation of the cone. Even though the nonlinear equation provides a conventional expression for the cone, it is almost impossible to find globally optimized values of unknown constants with current nonlinear optimization software, unless some close guess values for each constant are known. Therefore, most of the roundness measurement instruments, including CMMs, use linear equations to find the unknown variables of the equation for each of the different types of three dimensional substitute elements such as a cylinder, a cone or a torus. One of the well noted linearization techniques, limacon approximation, is used here for linearizing the conventional nonlinear equation for a cone. A comparison between the linearized equation using limacon approximation and the nonlinear form of the tilted cone is presented in the following sections.

3.1. Linear Formulation for a Cone

D.G. Chetwynd (1985) introduced the method of limacon approximation for linearizing the non-linear equation of a circle. The limacon approximation is applied to the transformation of the nonlinear equation to a linear equation of a circle by

Chetwynd (1985). In his article, it said that if the radial coordinate of a center of a circle in the Cylindrical Coordinate System E is quite small (see figure 1), compared with a radius of a circle R_0, the measurement by a limacon circle is more accurate than the measurement by the true circle which is expressed by a nonlinear equation. This means that utilization of the limacon circle gives a smaller error than the true circle, on the condition that the eccentricity ratio, $\varepsilon = \dfrac{E}{R_0}$, is less than 0.01.

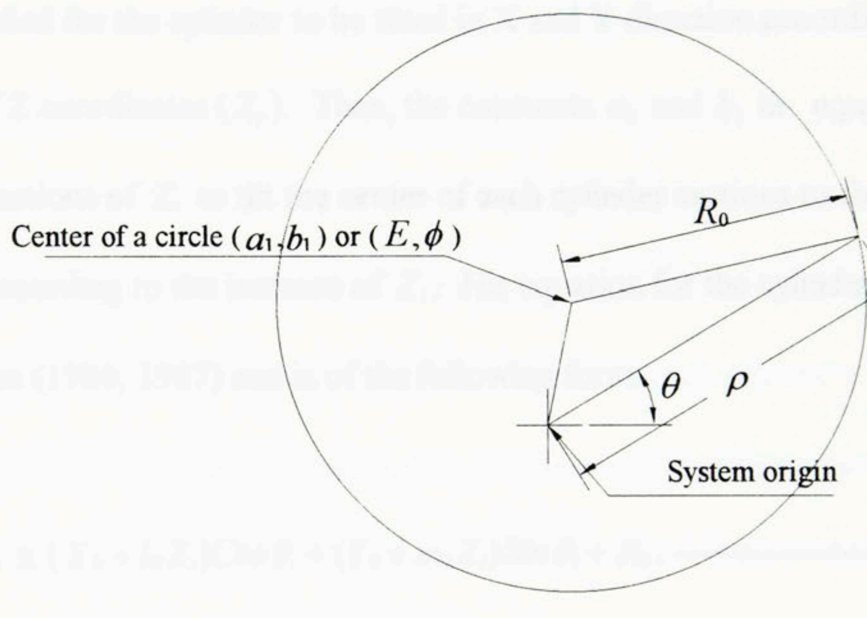


Fig 1. The linearized equation for a circle with Limacon approximation (Chetwynd,1985). where (E,ϕ) represents a center of a circle in the Polar Coordinate System, and (a_1, b_1) in the Cartesian Coordinate System. ρ is a radius from the system origin to the point of a circle, which is a substitute element.

The limacon equation to approximate the true circle as given in Chetwynd (1979, 1985) is:

$$\begin{aligned}\rho &= E\,Cos(\theta-\phi)+\sqrt{R_0^2-E^2\,Sin^2(\theta-\phi)}\\ &\cong E\,Cos(\theta-\phi)+R_0 \quad Where\ E \ll R_0 \text{ ------------------------[3]}\\ or &\cong a_1 Cos\theta + b_1 Sin\theta + R_0\end{aligned}$$

Based on equation [3], Shunmugam (1986) extended his theories relevant to the two dimensional limacon circle to a three dimensional object, a cylinder. Equation [3] is modified for the cylinder to be tilted in X and Y direction according to the increase of Z coordinates (Z_i). Then, the constants a_1 and b_1 in equation [3] become functions of Z_i to tilt the center of each cylinder sections to the X and Y direction according to the increase of Z_i. His equation for the cylinder is given in Shunmugam (1986, 1987) and is of the following form:

$$R_i \cong (X_0 + l_0 Z_i)Cos\,\theta_i + (Y_0 + m_0 Z_i)Sin\,\theta_i + R_0 . \text{ ---------------------[4]}$$

where R_i is a radial coordinate of a substitute cylinder of which the center axis is tilted. R_0 is a radius of a cylinder. θ_i and Z_i are θ and Z coordinates of a substitute cylinder in the Cylindrical Coordinate System. X_0 and Y_0 are the initial X and Y coordinates of the center of the circular section at the bottom of the cylinder.

$l_0 = \dfrac{\Delta X_i}{\Delta Z_i}$ (The amount of X_i increase according to the increase of Z_i) and

$m_0 = \dfrac{\Delta Y_i}{\Delta Z_i}$ (The amount of Y_i increase according to the increase of Z_i).

Modifying Shunmugam's notation and drawing for a cylinder, a cone can be expressed as in Figure 2. The drawing on the left, in Figure 2, expresses a cone which consists of a vertex of α degrees, a tilted center axis and cone sections which are represented as α, the middle line among three lines and circles. Points P_i, where $i = 1,2,3,\cdots,n$., are points measured by a standard CMM. The points can be expressed with three coordinates: r_i, θ_i, *and* z_i. Lines hy in Figure 2 represent lines on the surface of the substitute cone connected from the vertex of the cone to the bottom of the cone. These lines will be referred to as the hypotenuse of the cone throughout this thesis. The drawing on the right, expresses each section of the cone, and R_i is the radial coordinate on the circumference of each circular section which is calculated by the Least Squares Method or variations of the Minimum Zone Method using the measured points P_i. The radial coordinate of a substitute cone, R_i, is used to calculate half of the radial deviation which is denoted as hTd in the Minimum Zone Method. R_s represents the radius from the center axis of a substitute element to points on the surface of the substitute cone at Z_i and θ_i ($R_s = R_0 + S Z_i$). The r_i are radial coordinates of measured points obtained by a standard CMM. This means that the conicity can be calculated from radial coordinates on the substitute element, R_i, subtracted from r_i.

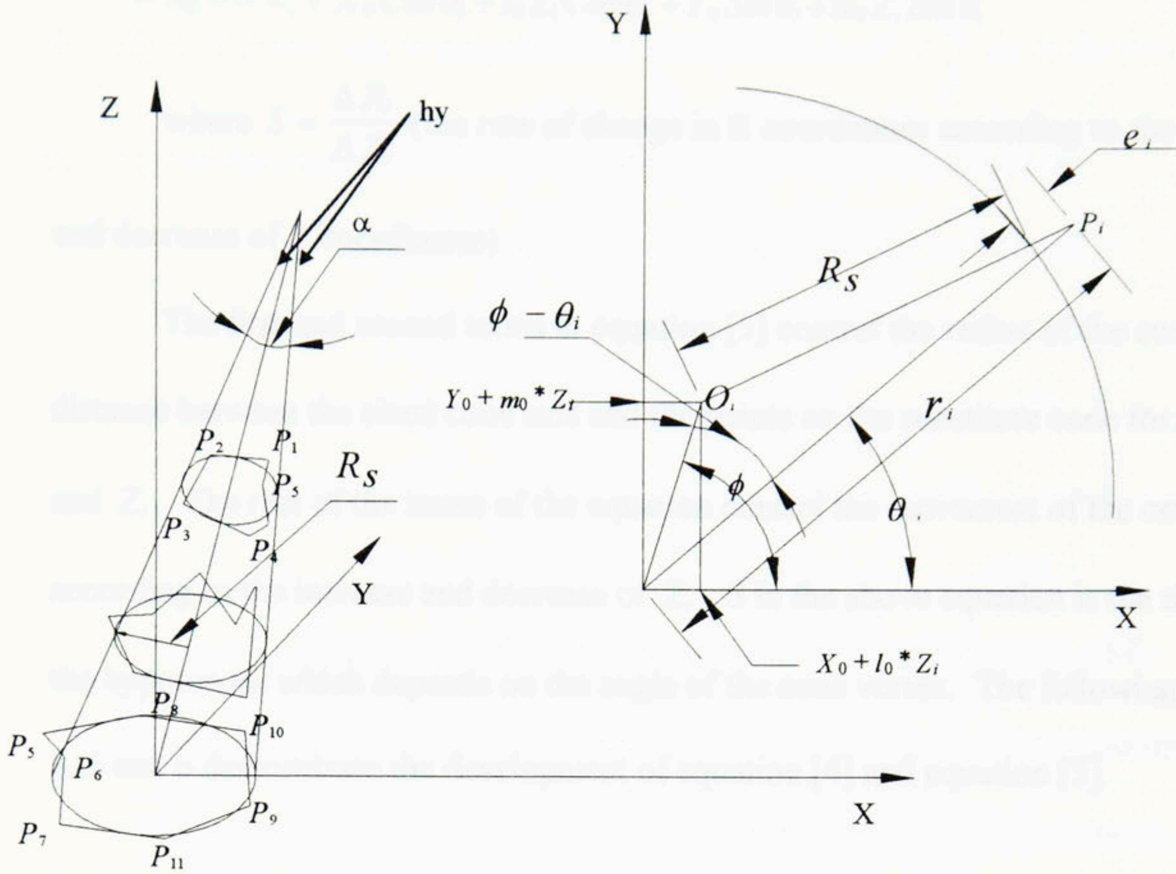


Figure 2. The graphic representation of the radial deviation e_i on each cone section

The equations derived in this work are based on equation [4] appropriately modified using analytic geometry. It is realized that if the vertex angle of the cone is increased to 180 degrees, the cone becomes a cylinder. If the constant R_0 in Shumugam's equation is allowed to increase according to the increase of Z_i, the Z coordinate of each section, equation [4] can be expressed to represent the shape of the cone in terms of Z_i and θ_i. Therefore, the radial coordinate of a substitute cone can be expressed in terms of θ_i, Z_i, and six constants.

$$R_i = (R_0 + S\,Z_i) + (X_0 + l_0 Z_i)Cos\,\theta_i + (Y_0 + m_0 Z_i)Sin\,\theta_i$$
$$= R_0 + S\,Z_i + X_0 Cos\,\theta_i + l_0 Z_i Cos\,\theta_i + Y_0 Sin\,\theta_i + m_0 Z_i Sin\,\theta_i \quad \text{-------[5]}$$

where $S = \dfrac{\Delta R_i}{\Delta Z_i}$ (the rate of change in R coordinates according to the increase and decrease of Z coordinates).

The first and second terms in equation [5] control the radius of the cone, the distance between the tilted cone axis and the points on the substitute cone for each θ_i and Z_i. The rest of the terms of the equation control the movement of the center axis according to the increase and decrease of Z_i. S in the above equation is the slope of the hypotenuse which depends on the angle of the cone vertex. The following Figures 3, 5 and 6 demonstrate the development of equation [4] and equation [5].

Step 1.

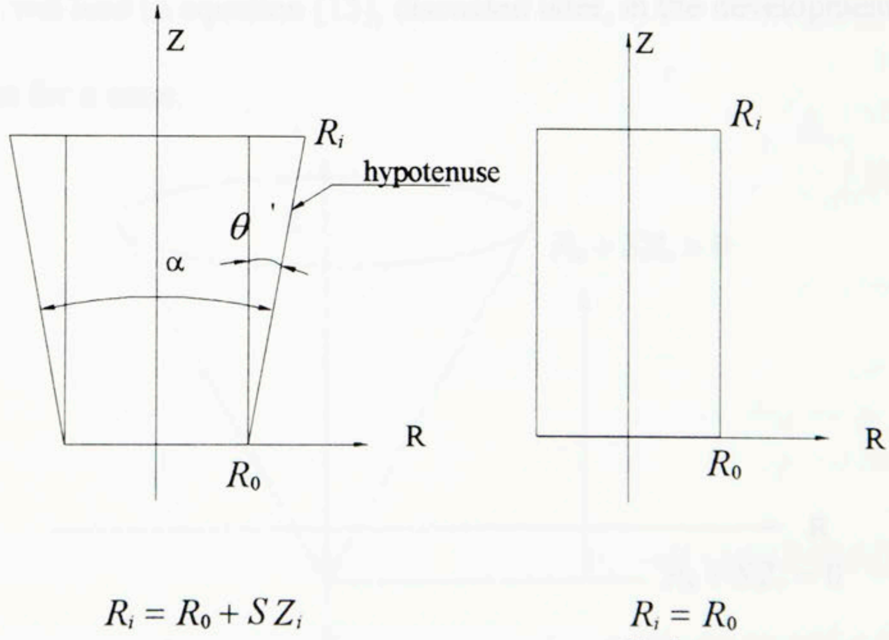


Figure 3. Representation of the first step of development for a cone in this thesis and a cylinder of Shunmugam (1986).

When the center axis of the cone coincides with the Z axis in the Cylindrical Coordinate System, and is perpendicular to the R axis, $\theta' = \frac{\alpha}{2}$ (α is the angle of the vertex), and the cone is referred to as a right cone throughout this thesis.

The transition from a cone and a cylinder is obtained in

$$S = \tan \theta'$$

$$\text{when } \theta' = 0^o \text{ then } \alpha = 180^o .$$

$$\tan 0^o = 0$$

$\therefore$ The term SZ_i should be dropped at $\theta' = 0^o$ or $\alpha = 180^o$.

Therefore, the hypotenuse of cone becomes the hypotenuse of the cylinder.

Since the tapered hole (with the shape of a cone) in Figure 6 is used to calculate the minimum zone in this thesis, the radius terms $R_0 + SZ_i$ should remain positive. This fact will lead to equation [13], discussed later, in the development of a nonlinear constraint for a cone.

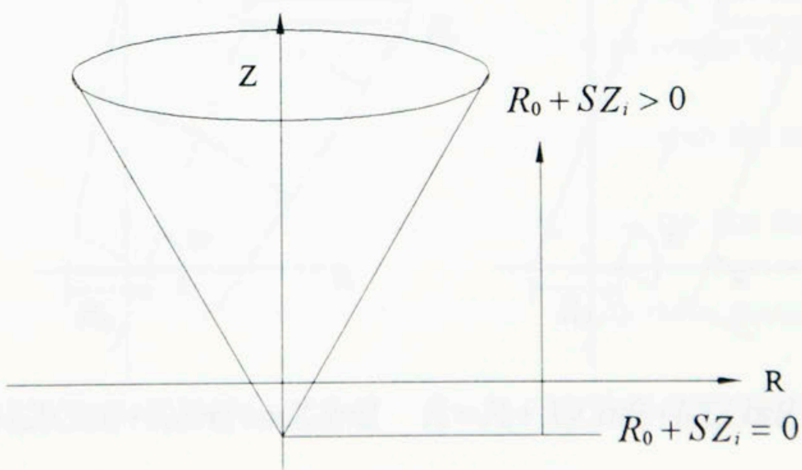
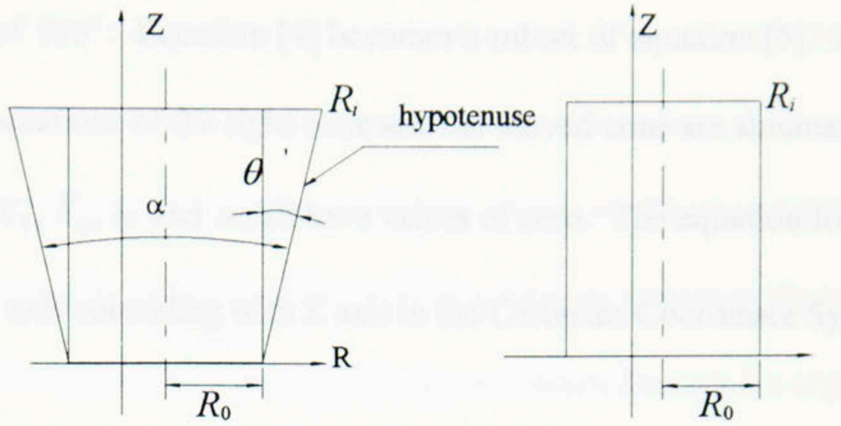


Figure 4. The scope of calculation to generate a cone hole.

Step 2.

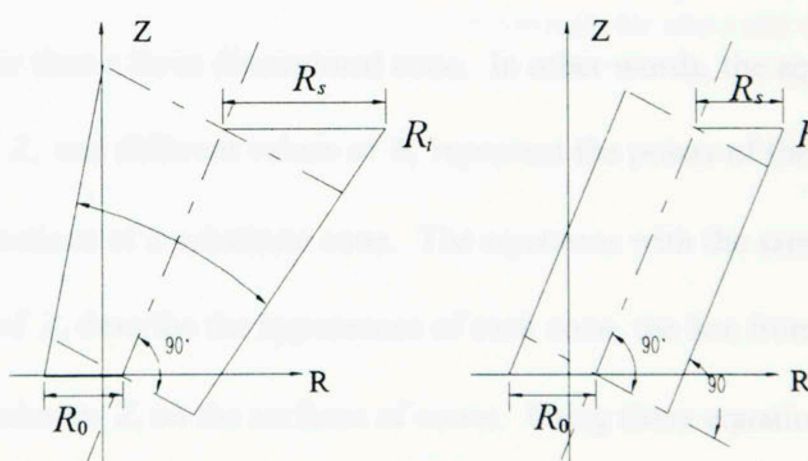


$R_i = R_0 + SZ_i + X_0 Cos\theta_i + Y_0 Sin\theta_i$ $\qquad$ $R_i = R_0 + X_0 Cos\theta_i + Y_0 Sin\theta_i$

cone $\qquad$ cylinder

Figure 5. Representation of the second step of development for a cone in this thesis and a cylinder of Shunmugam (1986). The center axis is allowed to be moved horizontally in the direction of the X and Y axes. This type of a cone is referred to as a moved cone throughout this thesis.

Step 3.



$R_i = R_0 + SZ_i + X_0 Cos\theta_i + l_0 Z_i Cos\theta_i + Y_0 Sin\theta_i + m_0 Z_i Sin\theta_i$ $\qquad$ $R_i = R_0 + X_0 Cos\theta_i + l_0 Z_i Cos\theta_i + Y_0 Sin\theta_i + m_0 Z_i Sin\theta_i$

Figure 6. Representation of the last step for a cone in this thesis and a cylinder of Shunmugam (1986). The center axis is allowed to be moved horizontally and tilted in the X and Y directions. This type of a cone is called a tilted cone through this thesis.

The above proofs and procedures show that equation [4] and equation [5] are identical for an α of 180^o. Equation [4] becomes a subset of equation [5]. Based on equation [5], the equations of the right cone and the moved cone are summarized here: For a right cone, X_0, Y_0, l_0 and m_0 all have values of zero. The equation for a right cone with a center axis coinciding with Z axis in the Cartesian Coordinate System is given as:

$$R_i = R_0 + S\,Z_0 \text{ ---[6]}$$

If the cone is assumed to be a right cone whose center axis is moved in X and Y directions, l_0 and m_0 assume values of zero in equation [5]. The equation of the moved cone is

$$R_i = R_0 + S\,Z_i + X_0 Cos\,\theta_i + Y_0 Sin\,\theta_i \text{ ----------------------------[7]}$$

Equations [5], [6] and [7] also represent radial coordinates of points on sections of a cone rather than a three dimensional cone. In other words, the equations with the same value of Z_i and different values of θ_i represent the points of the circumference on the sections of a substitute cone. The equations with the same value of θ_i and the increase of Z_i describe the hypotenuse of each cone, the line from the minimum Z_i to the maximum Z_i on the surfaces of cones. Using these equations, the radial deviations between points measured by the CMM and points on each type of the substitute cone can be calculated by equations [5], [6] and [7].

Since a hypotenuse generates a cone as it rotates around the center axis of the cone, the linear equation may explain the cone at the same θ_i more precisely than the nonlinear equation of the cone. Therefore, the radial deviations with the linear equation might be less than those with the nonlinear equation. Chetwynd (1979) and Lin (1995) arrive at similar opinions while discussing limacon approximation for circular objects. Chetwynd (1979) proved the above opinion through his experiments with 100 circles. Lin (1995) provides a confirmation of this opinion with a three dimensional substitute element, that of a cylinder. In Chapters 5 and 6, it will be determined whether the above opinion is correct in the case of cones, or not.

3.2. Nonlinear Formulation for a Cone

In this section, the procedure of developing a nonlinear equation for a tilted cone is discussed in detail. The procedure starts from an asymptotic cone and ends up as a cone with tilted center axis.

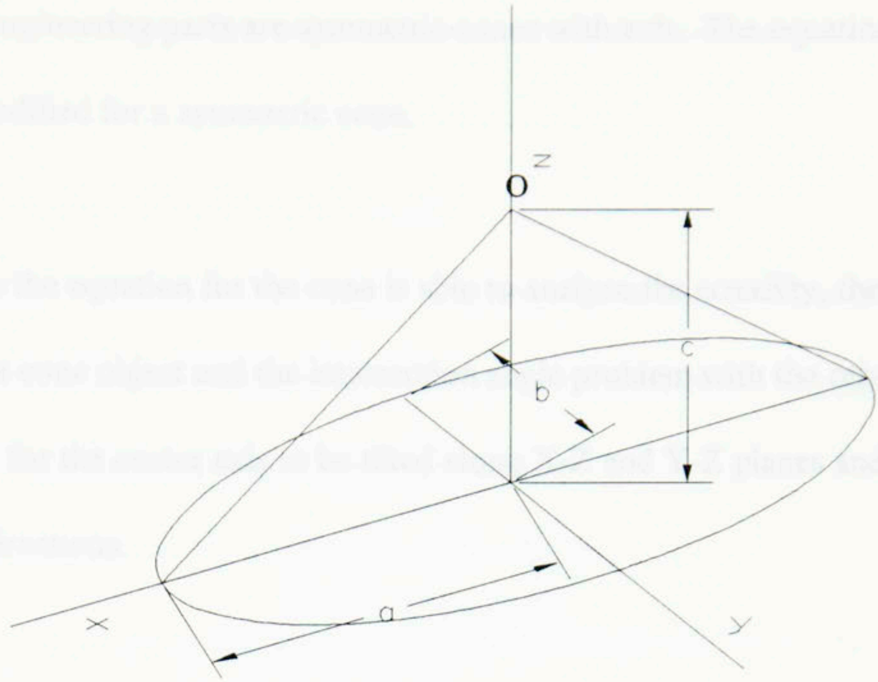


Figure 7. Asymptotic Cone

If a right cone of elliptical cross-section is assumed, with a center axis coinciding with the Z axis, the surface of a cone can be expressed by the following equation:

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{c^2}=0 \text{ --- [8]}$$

The real quadric cone is defined in John (1947) as a surface symmetrical with respect to each coordinate plane, axis and the origin. He mentioned that if $P_1(x_1, y_1, z_1)$ is any point on the surface, different from the origin O, then any point on the line O P_1 is also on the surface, since the coordinates of such a point have the form $k\,x_1$, $k\,y_1$ and $k\,z_1$, that satisfy the equation [8] of a cone. Since 'a' is not equal to 'b', equation [8],above, represents an asymmetric cone (Figure 7). However, the majority of engineering parts are symmetric cones with a=b. The equation for them should be modified for a symmetric cone.

Since the equation for the cone is able to analyze the coaxility, the form tolerance in a cone object and the intersection angle problem with the other object, it should allow for the center axis to be tilted along X-Z and Y-Z planes and be moved in X,Y and Z directions.

Step 1. If the cone is assumed to be a circular cone, a is equal to b in equation [8] resulting in:

$$X^2 + Y^2 = \left(\frac{c^2}{b^2}\right) Z^2 \text{ --[9]}$$

Step 2. Since the cone to be estimated should be allowed to move along X, Y and Z directions, the X, Y and Z may be replaced as:

$$X = X - X_0$$
$$Y = Y - Y_0$$
$$Z = Z - Z_0$$

thus yielding,

$$(X - X_0)^2 + (Y - Y_0)^2 = \left(\frac{c^2}{b^2}\right)(Z - Z_0)^2 \text{ ----------------------------[10]}$$

where X_0, Y_0 and Z_0 are initial movements of the cone in the X, Y and Z directions.

Step 3. To express a tilted cone, the center axis should be allowed to tilt along the X-Z and Y-Z planes. Five new constants are hence introduced in equation [9] and the new X, Y and Z in equation [9] are expressed as:

$$X = X - (X_0 + l_0 Z)$$
$$Y = Y - (Y_0 + m_0 Z)$$
$$Z = Z - Z_0$$

where $l_0 = \dfrac{\Delta X}{\Delta Z}$ (The increase of X coordinate according to the increase of Z coordinate), $m_0 = \dfrac{\Delta Y}{\Delta Z}$ (The increase of Y coordinate according to the increase of Z coordinate)

The variables in equation [9], X, Y and Z, can be replaced with the above linear equations and the constant term $\left(\dfrac{c}{b}\right)^2$ can be replaced with A for simplicity. Equation [9] will hence be expressed in the following revised form:

$$\left(X-(X_0+l_0Z)\right)^2+\left(Y-(Y_0+m_0Z)\right)^2=\left(\frac{c}{b}\right)^2(Z-Z_0)^2$$

or

$$\left(X-(X_0+l_0Z)\right)^2+\left(Y-(Y_0+m_0Z)\right)^2=A(Z-Z_0)^2 \quad \text{-------------[11]}$$

where

$$A=\left(\frac{c}{b}\right)^2 \; and \quad A\geq 0$$

The above term A always has a positive value.

Step 4. Since the deviations that we are looking for are the deviations in the radial direction, the coordinate system must be changed from the Cartesian Coordinates to the Cylindrical Coordinates to express the radial deviations easily. The variables, X, Y and Z, in equation [11] are hence replaced as:

$$X=r\cos\theta$$
$$Y=r\sin\theta$$
$$Z=Z$$

Thus, equation [11] can be transformed into a new form:

$$\begin{aligned}&r^2\{\cos^2\theta+\sin^2\theta\}-2r\{X_0\cos\theta+l_0Z\ \cos\theta+Y_0\sin\theta+m_0Z\ \sin\theta\}\\&+\{X_0^2+2X_0l_0Z+l_0^2Z^2+Y_0^2+m_0^2Z^2+2Y_0m_0Z-AZ^2-AZ_0^2+2AZ_0Z\}=0\end{aligned} \quad \text{--[12]}$$

The roots of the above formulation could be found as:

$$r=\left\{X_0\cos\theta+l_0Z\cos\theta+Y_0\sin\theta+m_0\ Z\ \sin\theta\right\}$$
$$\pm\sqrt{\begin{aligned}&\left\{X_0\cos\theta+l_0Z\cos\theta+Y_0\sin\theta+m_0\ Z\ \sin\theta\right\}^2\\&-\{X_0^2+2X_0l_0Z+l_0^2Z^2+Y_0^2+m_0^2Z^2+2Y_0m_0Z-AZ^2-AZ_0^2+2AZ_0Z\}\end{aligned}}$$

When comparing the above equation to equation [5], the square root is equivalent to $R_0 + S\,Z_i$ in equation [5] which controls radius of the tilted cone. Since the radius in the Cylindrical Coordinate System must be positive and added to the center axis to make the cone shape, the second term, the value of the square root, should be restricted to prevent getting a negative radius from the tilted axis. Due to this restriction, the equation for the radial coordinates in the Cylindrical Coordinate System can be expressed as :

$$
\begin{aligned}
& R_i = \{X_0\cos\theta + l_0 Z \ \cos\theta + Y_0\sin\theta + m_0 Z \ \sin\theta\} \\
& + \sqrt{D_i} \\
& where,\ D_i = \{X_0\cos\theta + l_0 Z \ \cos\theta + Y_0\sin\theta + m_0 Z \ \sin\theta\}^2 \\
& - \{X_0^2 + 2\,X_0 l_0 Z \ + l_0^2 Z^2 + Y_0^2 + m_0^2 Z^2 + 2\,Y_0 m_0 Z \ - A Z^2 - A Z_0^2 + 2A\,Z_0 Z\,\} \\
& and\ D_i \geq 0
\end{aligned}
\qquad \text{------[13]}
$$

Equation [13] consists of two terms: a center axis term and a radius term. The first term with the parenthesis controls the movement of the center axis according to the increase of the Z and θ coordinates. The second term with the square root decides the radius of the cone according to the increase of Z and θ coordinates. Adding the two terms allows one to express the radial coordinate of a cone in the Cylindrical Coordinate System in terms of an increase in θ and Z.

The above equation has two initial constraints, that must be satisfied.

- Radial coordinate (R_i) can not be negative in the Cylindrical Coordinate System.

- The values of D_i can not be negative. Otherwise, the radius will have imaginary parts.

These two constraints must be satisfied to obtain a cone.

<u>Step 5.</u> Since the variables, Z and θ, are the same variables as Z_i and θ_i in equation [5], these will be replaced for consistency of notation

.

$$
\begin{aligned}
&R_i = \{X_0\cos\theta_i + l_0 Z_i\cos\theta_i + Y_0\sin\theta_i + m_0 Z_i\sin\theta_i\} \\
&+ \sqrt{D_i} \\
&\textit{where}, D_i = \{X_0\cos\theta_i + l_0 Z_i\cos\theta_i + Y_0\sin\theta_i + m_0 Z_i\sin\theta_i\}^2 \qquad \text{---[14]} \\
&- \{X_0^2 + 2X_0 l_0 Z_i + l_0^2 Z_i^2 + Y_0^2 + m_0^2 Z_i^2 + 2Y_0 m_0 Z_i - AZ_i^2 - AZ_0^2 + 2AZ_0 Z_i\} \\
&\textit{and } D_i \geq 0
\end{aligned}
$$

Using these equations, two different types of equations for a tilted cone are presented in the above sections. Linear and nonlinear optimization techniques for each equation shown is presented in Chapters 5 and 6.

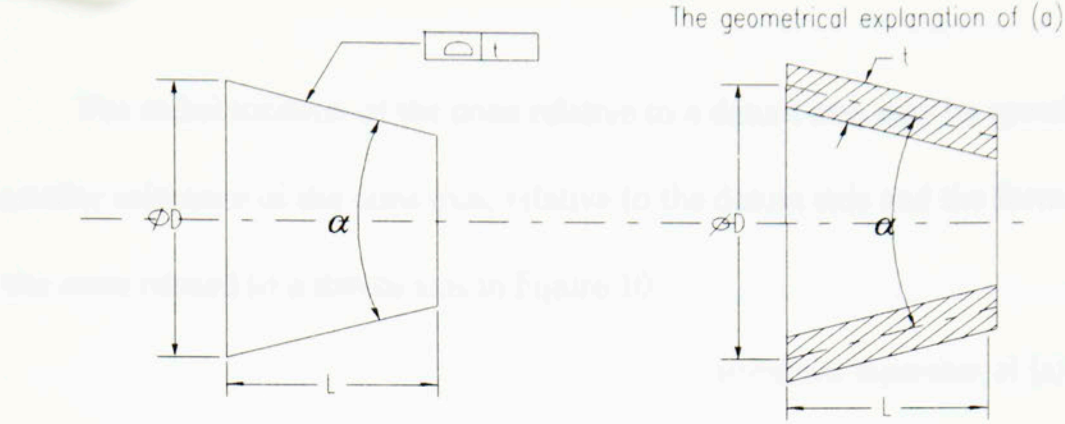
CHAPTER 4

TOLERANCE PROBLEMS IN CONES

General tolerance problems existing in cone inspection are defined by Henzold (1996). Typical problems dealt with are classified into four categories:

1. The form of the cone (so-called conicity);
2. Orientation and radial location of the cone relative to a datum;
3. Axial location of the cone relative to a datum (so-called the coaxiality problem); and
4. Distances of the endfaces of the truncated cone relative to a datum.

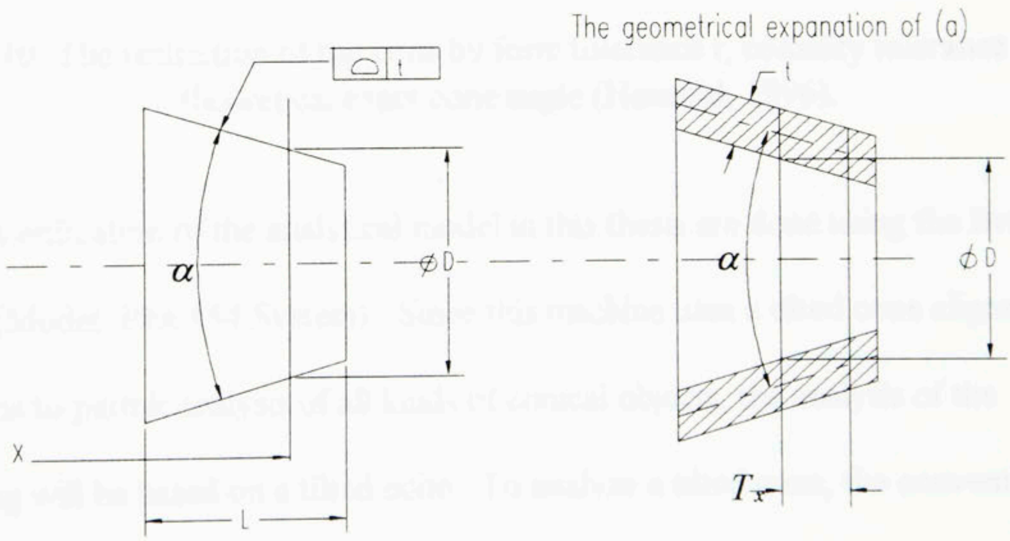
To understand the above problems, three drawings will be introduced in this section which are adapted from Henzold (1996). Figure 8 represents that the vertex angle of a cone can be specified by the theoretical exact angular dimension, and the form tolerance can be specified by the conicity t. Where Figure 8a is the conventionally dimensioned drawing with mechanical engineering notations, and Figure 8b presents a graphical interpretation of Figure 8a.



(a) (b)

Figure 8. Tolerancing specification in the convention of drawing by theoretical exact cone diameter and theoretical exact cone angle (Henzold,1996).

The form tolerance conicity can be controlled by 't' shown in Figure 9, and the axial location of the cone can be specified by the theoretical exact cone diameter and theoretical exact distance of the cone diameter from a datum in Figure 9.

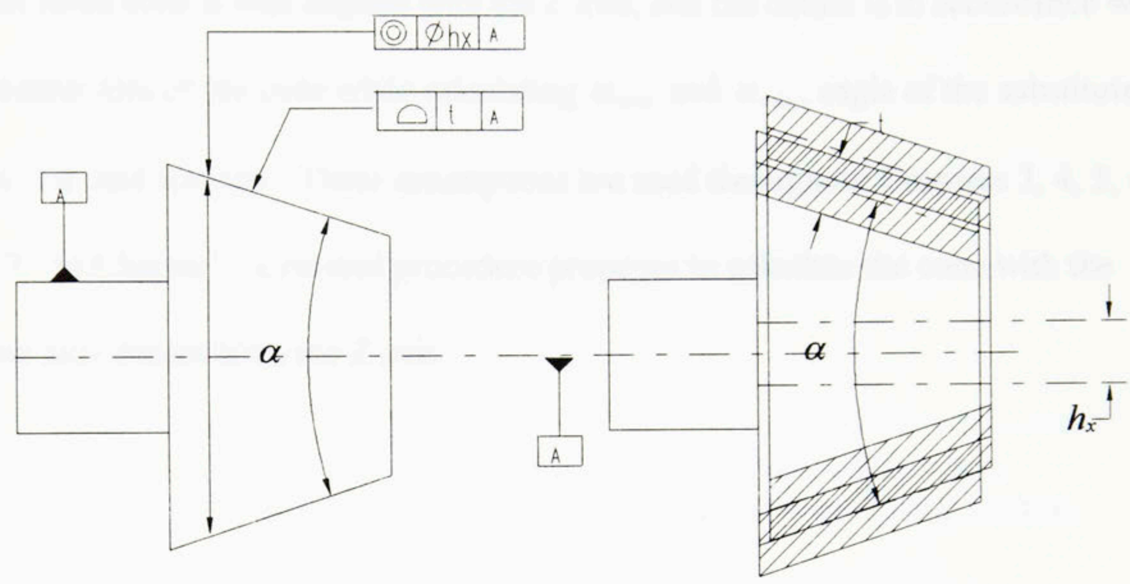


(a) (b)

Figure 9. The restriction of the radial location of the cone (Henzolod, 1996).. The axial deviation is limited by the tolerance of axial cone location T_x.

The radial location of the cone relative to a datum axis may be specified by the coaxiality tolerance of the cone axis, relative to the datum axis and the form tolerance of the cone related to a datum axis in Figure 10.

The geometrical explanation of (a)



(a) (b)

Figure 10. The restriction of the cone by form tolerance t, coaxility tolerance h_x and theoretical exact cone angle (Henzold, 1996).

Verification of the analytical model in this thesis are done using the Browne & Sharpe (Model: PFx 454 System). Since this machine uses a tilted cone aligned with the Z axis to permit analysis of all kinds of conical objects, the analysis of the following will be based on a tilted cone. To analyze a tilted cone, the conventional equations for T_d, α_{max}, α_{min}, conicity, and coaxiality are modified from the equations provided by Henzold (1996) for a right cone. α_{max} is the maximum vertex angle of cones. α_{min} is the minimum vertex angle of cones. T_d is the summation of a minimum deviation and a maximum deviation. In other words, it represents the distance in R direction or X direction from the two extreme points from the substitute cone. Furthermore, vector analysis is used here instead of arithmetic calculations as used in the Henzold (1996). The calculations used in this chapter assume that the center axis of the tilted cone is well-aligned with the Z axis, and the datum is in accordance with the center axis of the cone while calculating α_{max} and α_{min}, angle of the substitute cone, Td, and conicity. These assumptions are used throughout Chapters 3, 4, 5, 6, and 7. In Chapter 7, a revised procedure proposes to calculate the cone with the center axis distant from the Z axis.

4.1. Form Tolerance in Cones

The form tolerance is specified by the conicity t, in the above drawings. If the cone is a right cone, it is easy to find the conicity for the cone. However, if the cone is assumed to be a tilted cone, analysis becomes more complex, since the angle in the equation required to find the conicity is not related to a single angle, but the summation of two angles; half of the cone vertex angle and the angle of tilt. To find the summation of the two angles, consider Figure 11:

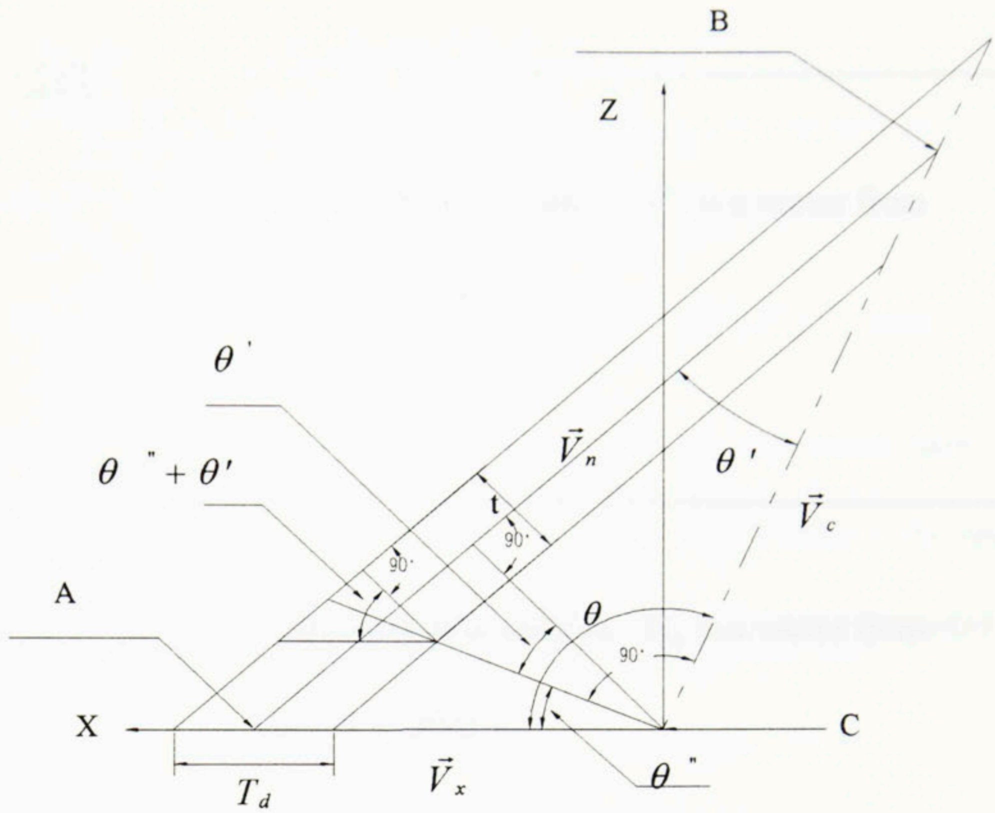


Figure 11.The analysis of conicity using vector calculations
This triangle represents half of the cone section on the X and Z plane.

$t = Td * \cos(\theta' + \theta'')$ --[15]

where t is the conicity, the normal distance of Td

θ' is the half of the vertex angle of cones,

the vertex angle of cones (α) is $2 * \theta'$, and

$$\theta'' = \theta - 90^0$$

θ is the angle between the center axis and the X axis

$\vec{V}_x, \vec{V}_c$ *and* $\vec{V}_n$ are vectors of radius, center axis and hypotenuse.

$$\cos\theta = \frac{\vec{V}x \bullet \vec{V}c}{|\vec{V}x||\vec{V}c|} \text{ --- [16]}$$

Where $\vec{V}_x$ is a vector from point C to point A, $\vec{V}_c$ is a vector from point C to point B

$$\cos\theta' = \frac{\vec{V}n \bullet \vec{V}d}{|\vec{V}n||\vec{V}d|} \text{ --- [17]}$$

where $\vec{V}_n$ is a vector from point B to point A. $\vec{V}_d$ is a vector from point B to point C.

Even though the estimated cone is not a cone with a right triangle section in X-Z plane, the above equations can be used to calculate the angle of vertex in the case of the tilted cone.

4.2. The Equation to Measure the Coaxiality of Cones.

To analyze the coaxiality problem in the cone, it is assumed that the datum is the Z axis. The equation to measure the coaxiality of the cone is a simple equation that uses the equation of a circle. If any points on the center axis of the cone are within the specified boundary, the cone is within the specified coaxiality h_x, based on the above assumption.

The equation to decide whether the center is out of the coaxiality zone of the cone or not, can be expressed as a circle with a radius and a center (system origin) giving:

$$X_c^2 + Y_c^2 = \left(\frac{h_x}{2}\right)^2 \text{ --[18]}$$

where X_c is the X coordinate of any point on the center axis of the tilted cone in the Cartesian Coordinate System, Y_c is the Y coordinate of any point on the center axis of the tilted cone in the Cartesian Coordinate System, and h_x is the coaxiality tolerance of the cone.

4.3. The Equations for Measuring $\alpha_{\max}$ and $\alpha_{\min}$

The $\alpha_{\max}$ and $\alpha_{\min}$ are useful information while determining whether the vertex of the measured cone is within the specified angle tolerance. The equation for these angles is introduced in the next section to find the maximum and minimum angles of the tilted cone. Since each equation to measure the $\alpha_{\max}$ or $\alpha_{\min}$ consists of the vectors $\vec{V}_{\max}$ and $\vec{V}_d$ or $\vec{V}_{\min}$ and $\vec{V}_d$, $\vec{V}_{\max}$ and $\vec{V}_{\min}$ should be found. $\vec{V}_{\max}$ is a vector from point B to point $C_{\max}$, and $\vec{V}_{\min}$ is a vector from point B to point $C_{\min}$. To find the X coordinate of $C_{\max}$, the maximum deviation should be added to point A, in Figure 12. Subtracting the absolute value of the minimum deviation from point A in Figure 12 leads to the X coordinate of $C_{\min}$.

The other point B in $\vec{V}_{\max}$ and $\vec{V}_{\min}$ should be found to obtain these vectors. The vertex of point B is found using the property that when the radius of the cone is zero, the point is the position of the vertex. When the linear equation is used to establish a substitute cone, the Z value of the vertex can be found by the following equation:

$$R_0 + S\,Z_i = 0 \quad \text{--[19]}$$

If a nonlinear equation is used to find the coordinates of the vertex (X_v, Y_v, Z_v), the following equations can be derived from equation [11]. The equations are derived with $Z_i = Z_0$ to find Radius=0. These equations are expressed in terms of five unknown constants providing:

$$\begin{aligned} X_v &= X_0 + l_0\,Z_0 \\ Y_v &= Y_0 + m_0\,Z_0 \quad \text{--[20]} \\ Z_v &= Z_0 \end{aligned}$$

where, X_v, Y_v and Z_v are X, Y and Z coordinates of the vertex.

After finding the coordinates of the vertex of the cone, the $\alpha_{\max}$ and $\alpha_{\min}$ can be found by equation [21]:

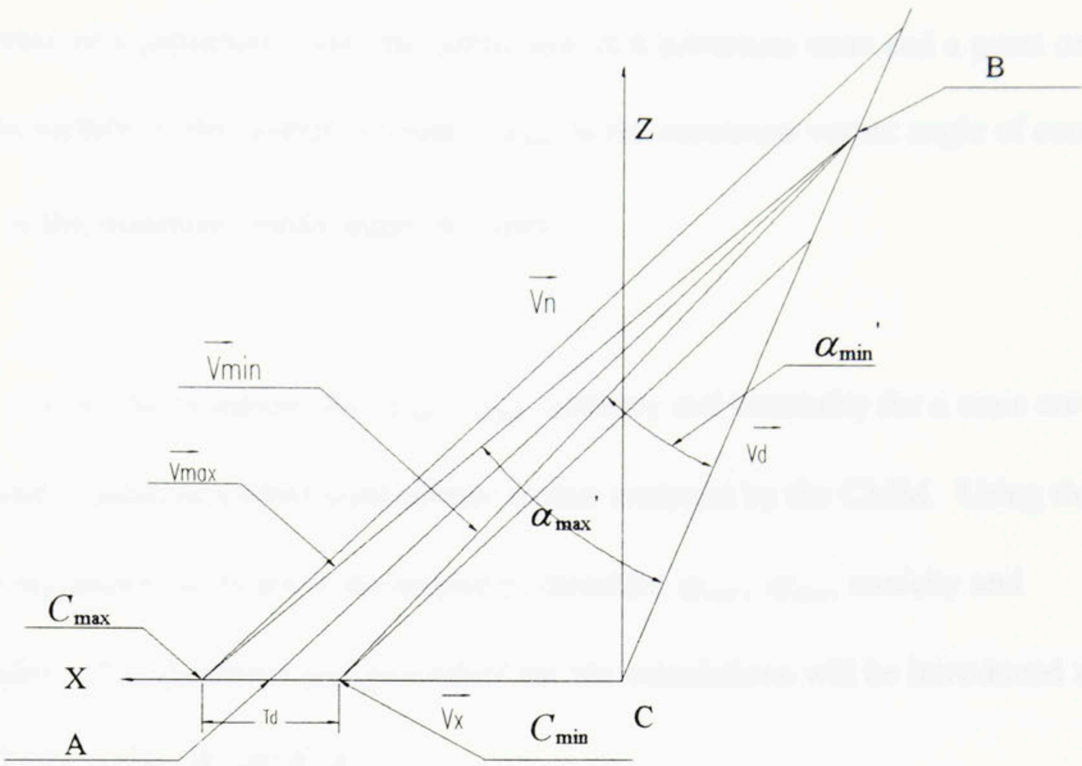


Figure 12. The analysis of α_{max} and α_{min}

$$\cos\alpha_{max}' = \frac{\vec{V}_{max} \bullet \vec{V}d}{|\vec{V}_{max}||\vec{V}d|}$$

$$\cos\alpha_{min}' = \frac{\vec{V}_{min} \bullet \vec{V}d}{|\vec{V}_{min}||\vec{V}d|} \text{ ---------------------------------[21]}$$

$$\alpha_{max} = 2 * \alpha_{max}'$$

$$\alpha_{min} = 2 * \alpha_{min}'$$

where Td is the deviation from the innermost cone to the outermost cone, the innermost cone and the outermost cone have the same slope as the hypotenuse of the substitute cone. $\vec{V}_d$ is a vector from point B to point C. $\vec{V}_{max}$ is a vector from point B to point C_{max}. $\vec{V}_{min}$ is a vector from point B to point C_{min}. α_{max}' is the angle which consists of the vertex of a substitute cone, the center axis of a substitute cone and a point on the bottom of the outermost cone. α_{min}' is the angle which consists of the vertex of a substitute cone, the center axis of a substitute cone and a point on the bottom surface of the innermost cone. α_{max} is the maximum vertex angle of cones. α_{min} is the minimum vertex angle of cones.

Now, the equations for α_{max}, α_{min}, conicity and coaxiality for a cone are prepared to analyze a tilted cone similar to that analyzed by the CMM. Using the above equations, software is developed to calculate α_{max}, α_{min}, conicity and coaxiality. The algorithm and procedure for the calculations will be introduced with flowcharts in the next section.

4.4. Flow Charts for Procedures of Analysis

The calculations of the above equations for the analysis of a tilted cone are done by developed programs and three commercial software programs: SAS, LINDO and GINO. An overview is shown through three self-explanatory flow charts illustrated in Figures 13, 14, and 15.

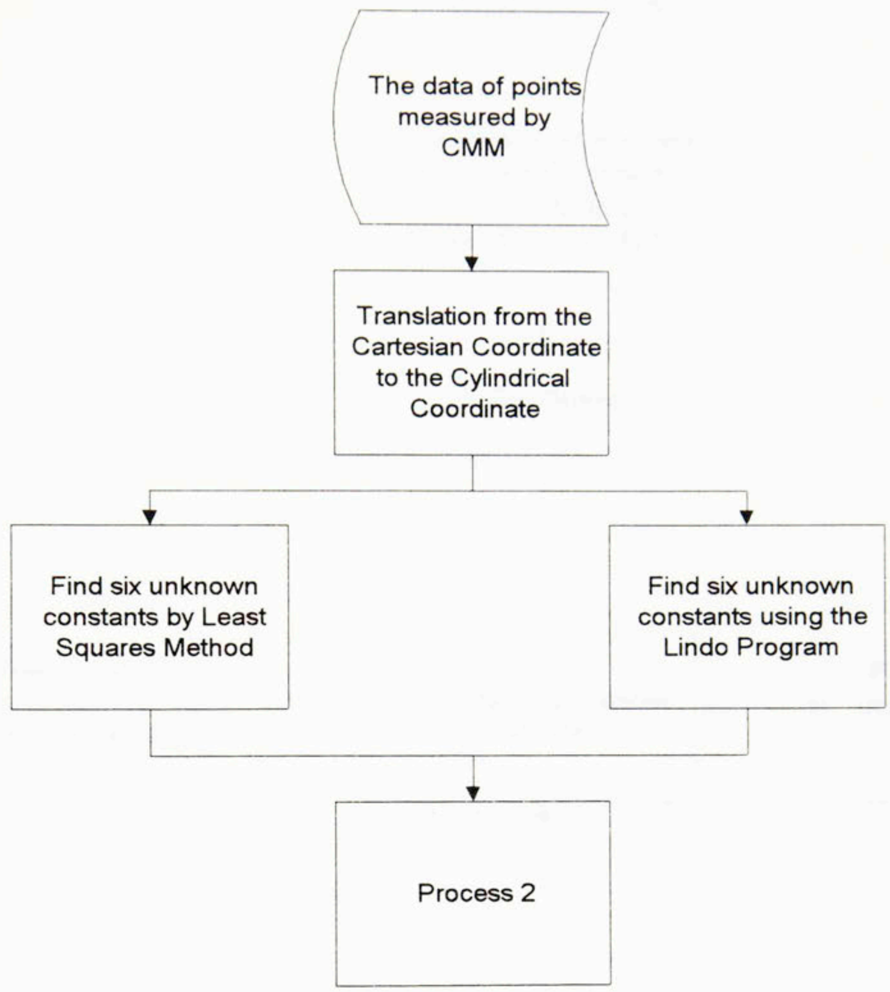


Figure 13. The flow chart using the procedure for the linear equation of a cone

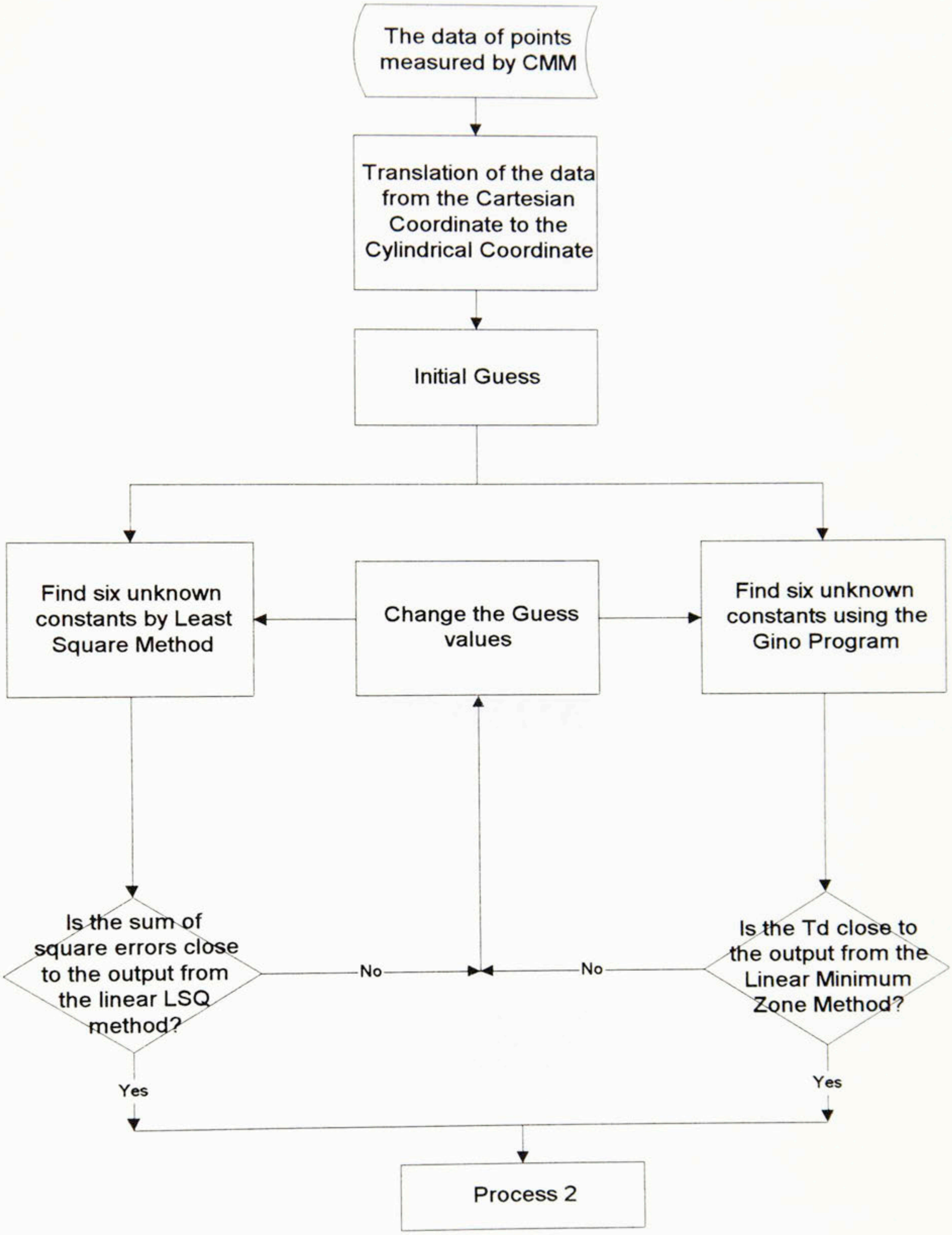


Figure 14. The flow chart using the procedure for the nonlinear equation of a cone

Process 2

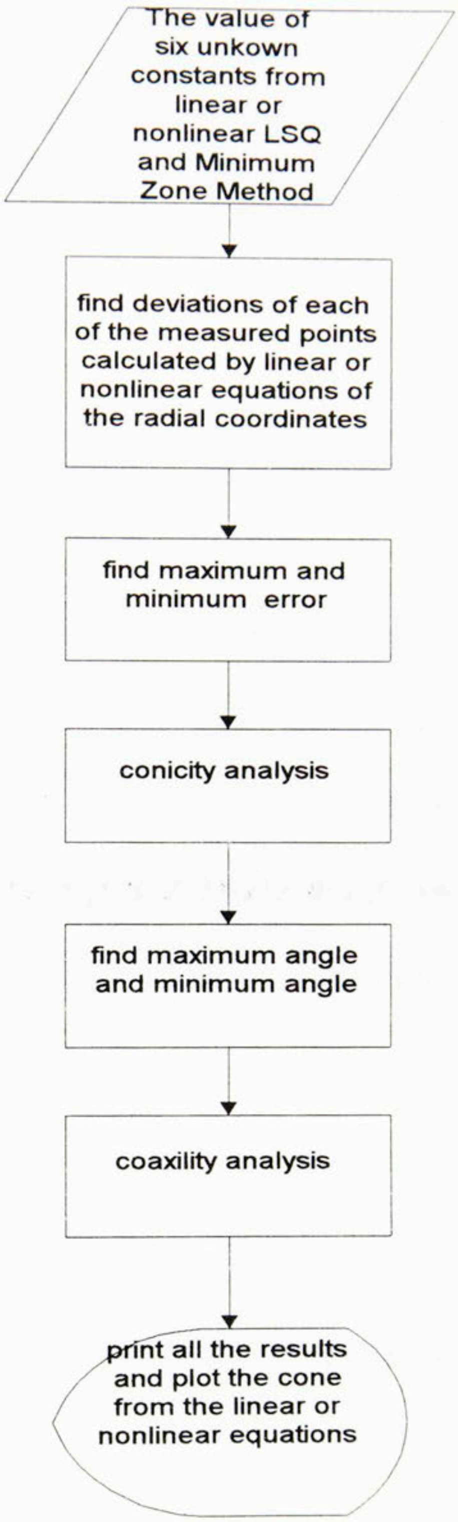


Figure 15. Flow chart of the common parts to both the linear and non-linear representation

CHAPTER 5.

THE LEAST SQUARES METHOD AND MINIMUM ZONE METHOD FOR CONES USING LINEAR EQUATIONS

5.1. The Least Squares Method with Linear Equations for Cones

The determination of constants in equation [5] in Chapter 3 depends on the sum of square errors from the substitute cone to the measured points being minimized. The Least Squares Method employed is frequently used in metrology and other disciplines. This method finds the unknown constants for the best fit cone. To decide the shape of the cone that minimizes the sum of square errors, the sum of square errors for the tilted cone are represented as:

$$\sum_i e_i^2 = \sum_i (r_i - (R_0 + S\,Z_i + X_0 Cos\theta_i + l_0 Z_i Cos\theta_i + Y_0 \sin\theta_i + m_0 Z_i Sin\theta_i))^2 \text{---------[22]}$$

where r_i is the radial coordinates of the measured points in the Cylindrical Coordinate System.

To minimize equation [22], above, the partial differential coefficients with respect to each unknown constant are given as:

$$\frac{\partial e_i}{\partial R_0}: Z_i\, S + R_0 + Cos\theta_i\, X_0 + Sin\theta_i\, Y_0 + Z_i\, l_0\, Cos\theta_i + Z_i\, m_0\, Sin\theta_i = r_i \text{ --------------[23]}$$

$$\frac{\partial e_i}{\partial S}: Z_i^2\, S + Z_i\, R_0 + Z_i\, Cos\theta_i\, X_0 + Z_i\, Sin\theta_i\, Y_0 + Z_i^2\, l_0\, Cos\theta_i + Z_i^2\, m_0\, Sin\theta_i = r_i\, Z_i \text{ --------------------------[24]}$$

$$\frac{\partial e_i}{\partial X_0}: Z_i\, Cos\theta_i\, S + Cos\theta_i\, R_0 + Cos^2\theta_i\, X_0 + Cos\theta_i\, Sin\theta_i\, Y_0 + Z_i\, Cos^2\theta_i\, l_0 + Z_i\, Cos\theta_i\, Sin\theta_i\, m_0 = r_i\, Cos\theta_i \text{ -------------[25]}$$

$$\frac{\partial e_i}{\partial l_0}: Z_i^2\, Cos\theta_i\, S + Z_i\, Cos\theta_i\, R_0 + Z_i\, Cos^2\theta_i\, X_0 + Z_i\, Cos\theta_i\, Sin\theta_i\, Y_0 + Z_i^2\, Cos^2\theta_i\, l_0 + Z_i^2\, Cos\theta_i\, Sin\theta_i\, m_0 = r_i\, Z_i\, Cos\theta_i \text{ ---------------------[26]}$$

$$\frac{\partial e_i}{\partial Y_0}: Z_i\, Sin\theta_i\, S + Sin\theta_i\, R_0 + Cos\theta_i\, Sin\theta_i\, X_0 + Sin^2\theta_i\, Y_0 + Z_i\, Cos\theta_i\, Sin\theta_i\, l_0 + Z_i\, Sin^2\theta_i\, m_0 = r_i\, Sin\theta_i \text{ ------------------------------[27]}$$

$$\frac{\partial e_i}{\partial m_0}: Z_i^2\, Sin\theta_i\, S + Z_i\, Sin\theta_i\, R_0 + Z_i\, Cos\theta_i\, Sin\theta_i\, X_0 + Z_i\, Sin^2\theta_i\, Y_0 + Z_i^2\, Cos\theta_i\, Sin\theta_i\, l_0 + Z_i^2\, Sin^2\theta_i\, m_0 = r_i\, Z_i\, Sin\theta_i \text{ --------------------[28]}$$

The above partial differential coefficients may be expressed in a matrix form such as:

$$A * B = C$$

where

A=

$$\begin{bmatrix}
\sum_i Z_i & \sum_i 1 & \sum_i Cos\theta_i & \sum_i Sin\theta_i & \sum_i Z_i Cos\theta_i & \sum_i Z_i Sin\theta_i \\
\sum_i Z_i^2 & \sum_i Z_i & \sum_i Z_i Cos\theta_i & \sum_i Z_i Sin\theta_i & \sum_i Z_i^2 Cos\theta_i & \sum_i Z_i^2 Sin\theta_i \\
\sum_i Z_i Cos\theta_i & \sum_i Cos\theta_i & \sum_i Cos^2\theta_i & \sum_i Cos\theta_i Sin\theta_i & \sum_i Z_i Cos^2\theta_i & \sum_i Z_i Cos\theta_i Sin\theta_i \\
\sum_i Z_i^2 Cos\theta_i & \sum_i Z_i Cos\theta_i & \sum_i Z_i Cos^2\theta_i & \sum_i Z_i Cos\theta_i Sin\theta_i & \sum_i Z_i^2 Cos^2\theta_i & \sum_i Z_i^2 Cos\theta_i Sin\theta_i \\
\sum_i Z_i Sin\theta_i & \sum_i Sin\theta_i & \sum_i Cos\theta_i Sin\theta_i & \sum_i Sin^2\theta_i & \sum_i Z_i Cos\theta_i Sin\theta_i & \sum_i Z_i Sin^2\theta_i \\
\sum_i Z_i^2 Sin\theta_i & \sum_i Z_i Sin\theta_i & \sum_i Z_i Cos\theta_i Sin\theta_i & \sum_i Z_i Sin^2\theta_i & \sum_i Z_i^2 Cos\theta_i Sin\theta_i & \sum_i Z_i^2 Sin^2\theta_i
\end{bmatrix}$$

$$B=\begin{bmatrix} S \\ R_0 \\ X_0 \\ Y_0 \\ l_0 \\ m_0 \end{bmatrix}, \qquad C=\begin{bmatrix} \sum_i r_i \\ \sum_i r_i Z_i \\ \sum_i r_i Cos\theta_i \\ \sum_i r_i Z_i Cos\theta_i \\ \sum_i r_i Sin\theta_i \\ \sum_i r_i Z_i Sin\theta_i \end{bmatrix} \text{---------------------------------------[29]}$$

For a subset of equation [22], the sum of square errors for the moved cone might be expressed as:

$$\sum_i (r_i - (R_0 + S\ Z_i + X_0\ Cos\theta_i + Y_0\ Sin\theta_i))^2 \text{ --[30]}$$

Corresponding partial derivatives with respect to each unknown constants are given by:

$$\frac{\partial e_i}{\partial R_0}: Z_i S + R_0 + Cos\theta_i X_0 + Sin\theta_i Y_0 = r_i \text{ ---------------} [31]$$

$$\frac{\partial e_i}{\partial S}: Z_i^2 S + Z_i R_0 + Z_i Cos\theta_i X_0 + Z_i Sin\theta_i Y_0 = r_i Z_i \text{ ---------------} [32]$$

$$\frac{\partial e_i}{\partial X_0}: Z_i Cos\theta_i S + Cos\theta_i R_0 + Cos^2\theta_i X_0 + Cos\theta_i Sin\theta_i Y_0 = r_i Cos\theta_i \text{ ---------------} [33]$$

$$\frac{\partial e_i}{\partial Y_0}: Z_i Sin\theta_i S + Sin\theta_i R_0 + Cos\theta_i Sin\theta_i X_0 + Sin^2\theta_i Y_0 = r_i Sin\theta_i \text{ ---------------} [34]$$

These equations can be expressed in a matrix form as:

$$\begin{bmatrix} \sum_i Z_i & \sum_i 1 & \sum_i Cos\theta_i & \sum_i Sin\theta_i \\ \sum_i Z_i^2 & \sum_i Z_i & \sum_i Z_i Cos\theta_i & \sum_i Z_i Sin\theta_i \\ \sum_i Z_i Cos\theta_i & \sum_i Cos\theta_i & \sum_i Cos^2\theta_i & \sum_i Cos\theta_i Sin\theta_i \\ \sum_i Z_i Sin\theta_i & \sum_i Sin\theta_i & \sum_i Cos\theta_i Sin\theta_i & \sum_i Sin^2\theta_i \end{bmatrix} * \begin{bmatrix} S \\ R_0 \\ X_0 \\ Y_0 \end{bmatrix} =$$

$$\begin{bmatrix} \sum_i r_i \\ \sum_i r_i Z_i \\ \sum_i r_i Cos\theta_i \\ \sum_i r_i Sin\theta_i \end{bmatrix} \text{ ---------------} [35]$$

If the cone to be estimated is assumed to be a right cone, equation [7] will reduce to that of a right cone (equation [6]). The only difference between equation [6] and the equation of a line is that the right cone lies on the X-Z plane instead of X-Y plane for the equation of a line. Since the Least Squares Method for a line is well known, the corresponding matrix is not illustrated in this work.

5.2. Minimum Zone Methods with Linear Equations of Cones

The minimum zone is calculated by a linear optimization method. To find optimal unknown constants of the linear equation of the cone, the optimization program LINDO is used, and the required outputs for conicity, α_{max}, and α_{min} are found by developed programs using the constants derived using LINDO. Basically, the LINDO program uses the Simplex Method which is well known for optimization. The basic theory and the algorithm of the Simplex Method are described in Bazaraa (1990) and Ravindran (1987).

5.2.1. Formulation of the Minimum Zone Method for a Tilted Cone

Typically, the tolerance zone of a cone is the summation of the maximum deviation and the minimum deviation from a substitute cone. However, all articles in the literature review indicate that the Least Squares Method can not give a minimum zone. Therefore, in this section, a way to guarantee minimum zone is introduced for cones.

The technique using half of the form tolerance was introduced by Chetwynd (1985) to get roundness of a circle. Since a cone has circular sections, a similar concept may be applied to a cone.

Objective function for this optimization is written as:

minimize hTd--[36]

subject to

constraint 1 is expressed as:

$|e_i| \le hTd$ where, hTd is the half of Td

or $\left| r_i - (R_0 + S\,Z_i + X_0 Cos\,\theta_i + l_0 Z_i Cos\,\theta_i + Y_0 Sin\,\theta_i + m_0 Z_i Sin\,\theta_i) \right| \le hTd$ ---------[37]

where Td is the radial distance between the outermost cone and the innermost cone, and $hTd = \dfrac{Td}{2}$.

or the above constraint may be split into two constraints as:

$$\begin{aligned} hTd + (R_0 + S\,Z_i + X_0 Cos\,\theta_i + l_0 Z_i Cos\,\theta_i + Y_0 Sin\,\theta_i + m_0 Z_i Sin\,\theta_i) \ge r_i \\ -hTd + (R_0 + S\,Z_i + X_0 Cos\,\theta_i + l_0 Z_i Cos\,\theta_i + Y_0 Sin\,\theta_i + m_0 Z_i Sin\,\theta_i) \le r_i \end{aligned} \quad \text{----------[38]}$$

The above constraints result in the error between the measured data and the substitute cone being less than the absolute value of the radial distance which is half of Td.

Constraint 2 is expressed as:

$hTd \geq 0$ --[39]

Since half of Td is the distance between the substitute cone and the measured points, it can not have a negative value.

Constraint 3 is written as:

$R_0 \geq 0$ --[40]

The initial radius might be restricted to be a positive value in the case of a cone-shaped hole inspected in this thesis. Practically, it is a characteristic of the radial coordinate is greater than zero, when $Z_i = 0$. Since the initial radius is the distance from the tilted cone axis to points on the surface of the cone at the minimum Z_i, the above inequality should be added into the constraints. Thus,

In the case of a right cone:

$$R_i = R_0 + SZ_i$$

$$\text{at } Z_i = 0$$

$$\text{thus, } R_i = R_0$$

where R_i are radial coordinates in the Cylindrical Coordinate System. By definition, $R_i \geq 0$. Therefore; $R_0 \geq 0$

In the case of a tilted cone:

$$R_i = R_0 + S\,Z_i + (X_0 + l_0 Z_i)Cos\theta + (Y_0 + m_0 Z_i)Sin\theta$$

at $Z_i = 0$

thus, $R_i = R_0 + X_0 Cos\theta + Y_0 Sin\theta$

Since we assume that the center axis is well aligned with the Z axis,

$$|R_0| >> |X_0 Cos\theta + Y_0 Sin\theta|$$

Therefore; if R_0 is negative, R_i must be negative. This can not happen in the Cylindrical Coordinate System

Constraint 4 is shown as:

$$R_0 + S \quad Z_i \geq 0 \text{---[41]}$$

The radius according to the increase of Z_i value must be restricted to positive values. The above equation implies that the radius from the tilted cone axis throughout the range of Z_i remains positive.

Since the four above constraints are repeated for each of the seven measured points, the Minimum Zone Method consists of one objective function and twenty three constraints, fourteen from constraint 1, one from constraint 2, one from constraint 3 and seven from constraint 4. These constraints are derived from the geometrical meaning of the cone itself.

Since an ideal cone can be tilted in negative directions, the constants to control the movement of the cone axis and the changing rate of the radius according to the increase of Z_i should be allowed to have negative values in LINDO, which initially assumes for all variables to be positive. To allow the constants to have negative values, the constants can be replaced with the following linear equation:

$$\begin{aligned} &S = S_1 - S_2 \\ &X_0 = X_1 - X_2 \\ &l_0 = l_1 - l_2 \\ &y_0 = y_1 - y_2 \\ &m_0 = m_1 - m_2 \\ &R_0, S_1, S_2, X_1, X_2, l_1, l_2, y_1, y_2, m_1, m_2, hTd \geq 0 \end{aligned} \quad \text{--[42]}$$

Due to the above characteristics, equations [37], [38] and [41] must be revised and allowed to have a negative value for the conicity.

$$\left| \begin{aligned} &r_i - (R_0 + S_1 \; Z_i - S_2 \, Z_i + X_1 \, Cos\,\theta_i - X_2 \, Cos\,\theta_i + l_1 \, Z_i \, Cos\,\theta_i \\ &- l_2 \, Z_i \, Cos\,\theta_i + Y_1 \, Sin\,\theta_i - Y_2 \, Sin\theta_i + m_1 \, Z_i \, Sin\,\theta_i - m_2 \, Z_i \, Sin\,\theta_i) \end{aligned} \right| \leq hTd \text{ --------------[43]}$$

or

$$hTd + (R_0 + S_1 Z_i - S_2 Z_i + X_1 Cos\theta_i - X_2 Cos\theta_i + l_1 Z_i Cos\theta_i$$
$$- l_2 Z_i Cos\theta_i + Y_1 Sin\theta_i - Y_2 Sin\theta_i + m_1 Z_i Sin\theta_i - m_2 Z_i Sin\theta_i) \geq r_i$$

------------------[44]

$$- hTd + (R_0 + S_1 Z_i - S_2 Z_i + X_1 Cos\theta_i - X_2 Cos\theta_i + l_1 Z_i Cos\theta_i$$
$$- l_2 Z_i Cos\theta_i + Y_1 Sin\theta_i - Y_2 Sin\theta_i + m_1 Z_i Sin\theta_i - m_2 Z_i Sin\theta_i) \leq r_i$$

$$R_0 + S_1 Z_i - S_2 Z_i \geq 0 \text{ --[45]}$$

5.2.2. The Minimum Zone Method for a Moved Cone

The approach used in the previous section and the objective function are similar to the equations in this section. The difference is that several variables such as l_0 and m_0 are eliminated from the constraints due to the different expressions of R_i of the cone. Dropping these variables causes the center axis to be parallel to the Z axis. The modified constraints and objective function are given as:

minimize hTd

subject to

Constraint 1 is expressed as:

$$\left| \begin{array}{l} r_i - (R_0 + S_1 Z_i - S_2 Z_i + X_1 Cos\theta_i - X_2 Cos\theta_i \\ + Y_1 Sin\theta_i - Y_2 Sin\theta_i) \end{array} \right| \leq hTd \text{ ------------------------------[46]}$$

or

$$hTd + (R_0 + S_1 Z_i - S_2 Z_i + X_1 Cos\theta_i - X_2 Cos\theta_i + Y_1 Sin\theta_i - Y_2 Sin\theta_i) \geq r_i$$

-------------------------------[47]

$$-hTd + (R_0 + S_1 Z_i - S_2 Z_i + X_1 Cos\theta_i - X_2 Cos\theta_i + Y_1 Sin\theta_i - Y_2 Sin\theta_i) \leq r_i$$

Constraint 2 is shown as

$hTd \geq 0$ ---[48]

Constraint 3 is written as:

$R_0 \geq 0$ ---[49]

Constraint 4 is expressed as:

$R_0 + S_1 Z_i - S_2 Z_i \geq 0$ --[50]

The above constraints have the same meanings as before.

5.3. Results of the Least Squares Method and the Minimum Zone Method using Linear Equations for Cones.

To find the values of the unknown constants in equations [5], [6] and [7] for each type of cone, LINDO, a noted optimization software developed by Schrage (1991), was used with constraint [36] through [50], derived in the previous section.

The calculations of Td, conicity, coaxiality, α_{max} and α_{min} are done by equation [15] through equation [21] in Chapter 4.

In the analysis of various cones with the Least Squares Method, the unknown constants in equations [5] and [7] may be found by solving matrices [29] and [35]. After finding the unknown constants in equation [5] and equation [7] with the Least Squares Method, the same procedure explained in Chapter 4 is applied to find the values of Td, conicity, α_{max} and α_{min}.

The results for Td, conicity, α_{max}, and α_{min} are obtained using Matlab, a mathematical calculation program, and are organized in Table 1 and Table 2. Table 1 illustrates the errors between a substitute cone and each measured point and the sum of the square errors. Also, the Td, conicity, angle of a substitute cone, maximum angle and minimum angle generated by the Least Squares Method and Minimum Zone Method using each type of cone are presented in Table 3.

To conduct a preliminary test of the equations [5], [6] and [7], the Coordinate Measuring Machine (Browne&Sharpe, PFx 454 System) was used to measure the coordinates of measured points and compute conicity for comparison with the angle of the vertex and conicity developed in this thesis. The Minimum Zone Method using the Simplex Method will be referred to as the Simplex Method, for convenience, hereinafter.

Sample analysis is presented using data collected from a preliminary test (data sample available in Appendix A). Table 1. shows the different characteristics between the Least Squares Method and the Simplex Method for a sample set of data. The Least Squares Method minimizes the sum of square errors. Thus, the deviations of all measured points from the substitute cone affects the calculations in finding unknown constants in equations [5], [6] and [7]. However, the Minimum Zone Method seeks to minimize the sum of the maximum and the minimum deviations as low as possible.

Table 1. The Comparison of Deviations from the Measured Points

unit: inch

	RIGHT CONE		MOVED CONE
Method	LSQ	SIMPLEX	LSQ
Deviations	-0.00011146	0.00043896	-0.00049225
	0.00016873	0.00071916	0.00050745
	0.0003421	0.00089253	0.00070552
	0.00013735	0.00040305	-0.00021079
	-0.00133943	-0.00107404	-0.0010294
	0.00121713	0.00107389	0.00053282
	-0.00041443	-0.00107434	-0.00001336
Td	0.00255656	0.00214823	0.00173492
SES	0.000003624	0.000005130	0.000002386
	MOVED CONE	TILTED CONE	
Method	SIMPLEX	LSQ	SIMPLEX
Deviations	-0.00071211	0.00031013	0.00036702
	0.00071281	0.00027351	0.00036648
	0.00071243	0.00010737	0.00036684
	-0.00031058	-0.00054485	-0.00036761
	-0.00071150	-0.00045454	-0.00036818
	0.00071259	0.00017138	0.00036650
	0.00050196	0.00013699	0.00036656
Td	0.00142492	0.00085498	0.00073520
SES	0.000002885	0.0000007341	0.0000009430

where SES is Sum of Error Squares.

Toru Kanada (1995) suggests that the extreme points (maximum and minimum deviations) measured by the Least Squares Method are the same extreme points measured by the Minimum Zone Method. Based on our initial experiments, if may state that for the given data, this trend was different in our analysis.

The comparison of deviations for the tilted cone provides a good example for the different characteristics between the Least Squares Method and the Simplex Method. Even though the Least Squares Method shows a lesser value for summation of square deviations, the Simplex Method shows a smaller Td than the Least Squares Method. All the types of cones illustrated the same trend as the tilted cone.

Table 2 illustrates a comparison of the conicity and the angle of vertex calculated by the software offered by the Browne & Sharpe (Tutor) and the conicity, angle of vertex, α_{max}, and α_{min} obtained by the program developed in this thesis. Since the CMM is not able to provide α_{max}, and α_{min}, these are not listed in the table.

Table 2. The Comparison of the Least Squares Method and the Simplex Method
Td and conicity are measured in inches, angles are measured in degrees

		Td	conicity	angle of a substitute cone	max. angle	min. angle
Tilted cone	CMM	*	0.0009	20.134753	*	*
Tilted cone	L.S.Q	0.00085498	0.00084222	20.143489	20.160715	20.113224
Tilted cone	Simplex	0.00073520	0.00072422	20.098651	20.119011	20.078225
Moved cone	L.S.Q	0.00173492	0.00170862	19.979709	20.018638	19.922903
Moved cone	Simplex	0.00142492	0.00140346	19.909636	19.948886	19.87042
Right cone	L.S.Q	0.00255656	0.0025178	19.978837	20.04601	19.904899
Right cone	Simplex	0.00214823	0.00214823	20.153334	20.212987	20.093645

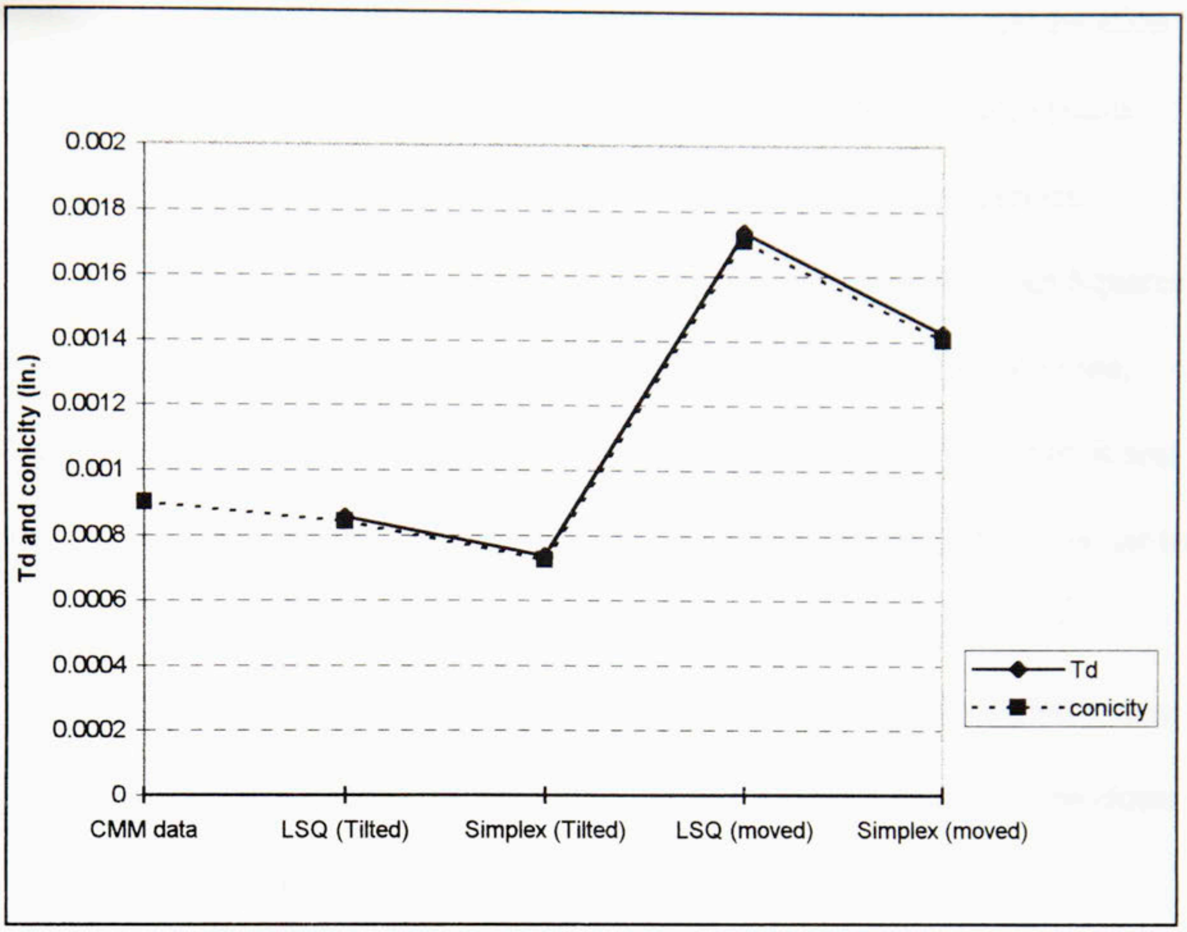


Figure 16. The comparison of Td and conicity calculated by each method.

The most important observation is that the Simplex Method always gives a smaller Td and a smaller conicity than the Least Squares Method. In the case of a tilted cone while using my developed program, the conicity and Td computed using the Simplex Method is approximately 14% better than that obtained with the Least Squares Method. In the case of a moved cone, the conicity and Td computed using the Simplex Method is approximately 17.8% better than those obtained with the Least Squares Method. If the Td and conicity of the tilted cone are compared with the Td and conicity of the moved cone, those of the tilted cone have smaller values than those of the moved cone. This may be explained through a consideration of the tilted cone.

Since the center axis of a tilted cone can be tilted in X and Y direction, the equation of a tilted cone can be constructed with smaller deviations from the measured points.

Since the type of the cone used by the Coordinate Measuring Machine (Browne & Sharpe, Model PFx 454 System) is a tilted cone, using the Least Squares Method, the Least Squares Method in the developed program using a tilted cone, should show a close agreement with the CMM's computed results. However, it was observed that the conicity calculated by the program offered by the CMM is closer to the Td value calculated by the developed program written in Matlab program language. This means that the definition of conicity, so identified by the CMM, may be different than the definition of conicity developed in this thesis, and may be closer to the definition of Td developed in this thesis.

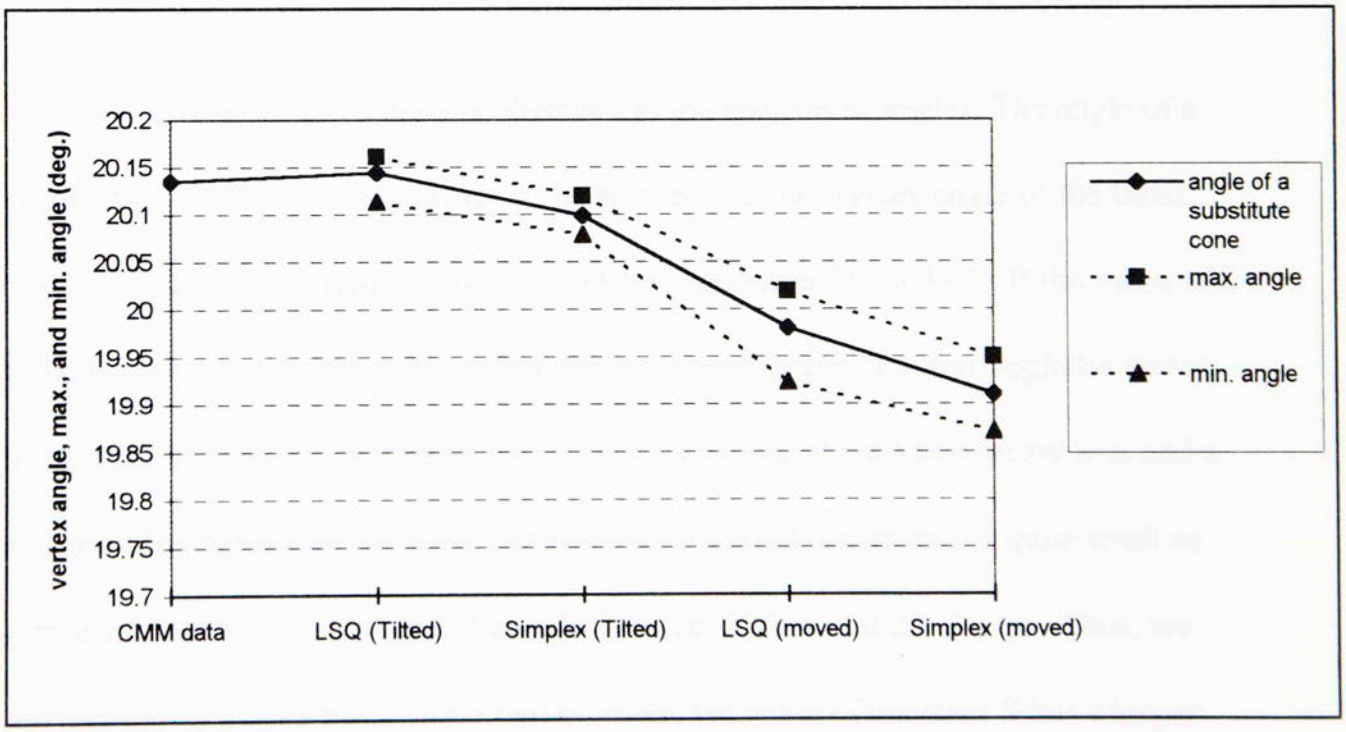


Figure 17. The comparison of vertex angle , α_{max} , and α_{min} .

Another interesting observation from the plot provided in Figure 17 is in the comparison of the difference between the maximum angle and the angle of a substitute element, and the difference between the angle of a substitute element and the minimum angle. When the Least Squares Method is applied to find an optimized substitute cone, the above two differences are not the same. In other words, the substitute cone generated by the Least Squares Method does not lie in the middle of the outermost and the innermost cone. The opinion that the Least Squares Method does not find a substitute element to guarantee a minimum tolerance zone is proved again by the above conclusion. On the other hand, the vertex angle of the cone analyzed by the Simplex Method lies in the middle of α_{max} and α_{min}. This means that the substitute cone lies in the middle of the two cones so that the minimum tolerance zone can be constructed.

From Figure 17, a trend is observed in the analysis of angles. The angle of a moved cone might have a tendency to be smaller than the vertex angle of the tilted cone. This trend is caused by the constant S in equations [5] and [7]. If the value of S in the equation is a larger value, it has a wider vertex angle. Even though the vertex angle in a tilted cone is related to two vectors, a vector of the tilted center axis and a vector on the surface of the cone, the angle of the tilted center axis is quite small as compared to the angle between the vector on the surface and the Z axis. Thus, we may say that if a cone has a wider vertex angle, the value of constant S has a larger value in this experiment. Another interesting observation is that vertex angle generated by the CMM is a smaller angle than that generated by the Least Squares Method with the developed program.

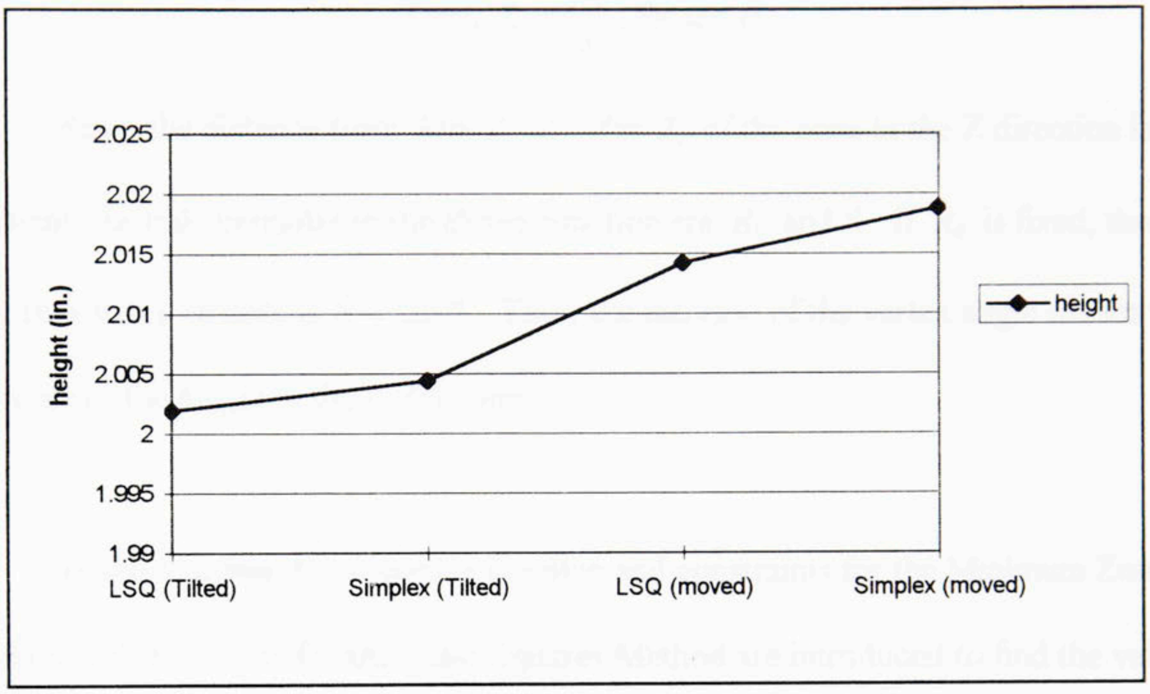


Figure 18. The comparison of height in the Z direction.

Figure 18 represents the height in the Z direction generated by the different types of cones, using the different methods. Comparing Figures 17 and 18, an interesting trend may be noted. With a decrease in the vertex angle of the cone, an apparent increase in the height of the cone is noted. The above observation seems to be quite reasonable in the case of a three dimensional cone. If the initial radius is unchanged, the angle is a factor that controls the height of the cone. The following logic is used to find the height in the Z direction for each cone.

$$\text{height} = Z_v + |Max\,Z_i - Min\,Z_i|$$

Since $|Max\,Z_i - Min\,Z_i|$ is constant, Z_v is the only factor that controls height.

Where, Z_v is Z_i at Radius = 0. Radius = $R_0 + S\ Z_i$.

$$\text{Thus, } Z_i = \frac{R_0}{S}, \text{ S= } \tan\theta$$

Since the distance from $Min\,Z_i$ to $Max\,Z_i$ of the cone in the Z direction is a constant, the only variables in the above equation are R_0 and S. If R_0 is fixed, the only remaining variable is $S = \tan\theta$. Thus, the increase of the vertex angle causes the decrease of the height in the Z direction.

In this Chapter, the objective function and constraints for the Minimum Zone Method and the Matrix for the Least Squares Method are introduced to find the values of unknown constants in equation [5]. Using these constraints and matrix, Td, conicity, α_{max}, and α_{min} are found by each method. In the next Chapter, Td, conicity α_{max}, and α_{min} will be found by a nonlinear equation for a tilted cone.

CHAPTER 6

NONLINEAR OPTIMIZATION OF CONES

Getting a global optimization by the nonlinear optimization technique is almost impossible in that the nonlinear equation might have several local optimal points. However, if a close initial value to the globally optimized point is known, it is possible for the nonlinear optimization technique to find a local optimization point close to the global optimal. In other words, the nonlinear optimization technique can lead to a wrong answer without a good initial guess of the unknown constants of the equation.

Four of the six unknown constants of equation [5] may be used from the previous chapter's linear optimization results. These could serve as the initial guess to obtain the optimal point. To guess the two remaining unknown constants (Z_0 and A), guessing equations have been developed as:

$$\begin{aligned} &for \quad R_0 + S\,Z = 0 \\ &Z_0 = Z \end{aligned} \qquad \text{[51]}$$

Equation [51] means that the guess value for Z_0 is determined by Radius = 0. The Z_0 determines the values of Z as the hypotenuse of the cone meets the center axis of a cone. Using X_0, Y_0, l_0, and m_0 from the previous chapter's linear optimization results, Z_0 from equation [51], and equation [11], A may be found as:

$$A = \frac{(X-(X_0+l_0Z))^2 + (Y-(Y_0+m_0Z))^2}{(Z-Z_0)^2}$$

where --[52]

$$A = \left(\frac{a}{c}\right)^2 \textit{ and } \quad A \geq 0$$

The values of X, Y and Z are the Cartesian points on the optimal cone and are obtained using the programs in Appendix B-1 named as 'Revise.m' and 'Lpcone1.m'. These programs generate the X, Y and Z coordinates of points on the cone optimized by the Linear Simplex Method and using the Linear Least Squares Method. The calculation is based on the assumption that the X, Y and Z values of the points on the surface of a cone generated by the nonlinear equation of the cone might have a location similar to the points on the surface of a cone calculated by the linear equation of the cone. The above assumption is based on Chetwynd's experiments (1979) on circles. Chetwynd states that the radial variation between the limacon and the circle is at most a fraction of a percent of the total signal caused by eccentricity. Even though the object is not a circle but a cone which is a three dimensional object, the radial variations between a linear equation using limacon approximation and a nonlinear equation for the cone will be close to one percent. Based on the above idea, the unknown constants in the nonlinear equation may be set close to the unknown constants in the linear equation. Coordinates of two points on the cone with two different Z coordinates generated by the linear equation are used as X, Y and Z values of equation [52].

6.1. The Least Squares Method for the Nonlinear Representation of the Cone

To solve the nonlinear least square problem for the cone, a procedure called "NLIN" in the SAS software package is used. In the first step, the guessed values of each unknown constant in the nonlinear equation and the partial differential coefficients with respect to them should be provided. Using this information, a local optimal point is found by five iterative methods given in SAS. To find the minimum sum of square errors, the following five iterative methods are given in the SAS program:

- the Steepest-Descent or Gradient Method
- the Newton Method
- the Modified Gauss-Newton Method
- the Marquardt Method
- the Multivariate Secant of False Position Method (also called the DUD method)

These algorithms are explained in SAS (1990), Rao (1996) and Reklaitis (1985).

Four of the methods (except the Newton method) are used to find optimal values of the six unknown constants in the nonlinear equation of the cone. Each iterative method requires at least three kinds of information: the names and starting values of the parameters, the model of nonlinear equation to be optimized, and the partial derivatives of the function with respect to each dependent variable. Since our model require minimization of hTd in the radial direction of the Cylindrical Coordinate system, the model to be estimated in the cone is the cone's radial coordinates, of which the dependent variables are X_0, Y_0, m_0, l_0, Z_0, and A. The radial coordinates of the surface of a cone are expressed in the following form:

$$R_i = \{X_0\cos\theta_i + l_0 Z_i\cos\theta_i + Y_0\sin\theta_i + m_0 Z_i\sin\theta_i\} + \sqrt{D_i}$$
$$where\ D_i = \{X_0\cos\theta_i + l_0 Z_i\cos\theta_i + Y_0\sin\theta_i + m_0 Z_i\sin\theta_i\}^2 \text{ ----------[53]}$$
$$- \{X_0^2 + 2X_0 l_0 Z_i + l_0^2 Z_i^2 + m_0^2 Z_i^2 + Y_0^2 + 2Y_0 m_0 Z_i - AZ_i^2 - AZ_0^2 + 2AZ_0 Z_i\}$$

The partial differential coefficients with respect to each unknown constants are needed to find the optimal point, and are given as:

$$\frac{\partial R_i}{\partial X_0} = \frac{\{X_0 Cos^2\theta_i + l_0 Z_i Cos^2\theta_i + Y_0 Cos\ \theta_i\ Sin\ \theta_i + m_0 Z_i Cos\ \theta_i\ Sin\ \theta_i - X_0 - l_0 Z_i\}}{\sqrt{D_i}} + Cos\theta \text{ ------------------------[54]}$$

$$\frac{\partial R_i}{\partial Y_0} = \frac{\{Y_0 Sin^2\theta_i + m_0 Z_i Sin^2\theta_i + X_0 Cos\ \theta_i\ Sin\ \theta_i + l_0 Z_i Cos\ \theta_i\ Sin\ \theta_i - Y_0 - m_0 Z_i\}}{\sqrt{D_i}} + Sin\ \theta \text{ -----------------------[55]}$$

$$\frac{\partial R_i}{\partial l_0} = \frac{\{l_0 Z_i^2 Cos^2\theta_i + X_0 Z_i Cos^2\theta_i + Y_0 Z_i Cos\ \theta_i\ Sin\ \theta_i + m_0 Z_i^2 Cos\ \theta_i\ Sin\ \theta_i - X_0 Z_i - l_0 Z_i^2\}}{\sqrt{D_i}} + Z_i Cos\ \theta \text{ -------------[56]}$$

$$\frac{\partial R_i}{\partial m_0} = \frac{\{m_0 Z_i^2 Sin^2 \theta_i + Y_0 Z_i Sin^2 \theta_i + X_0 Z_i Cos\ \theta_i\ Sin\ \theta_i + l_0 Z_i^2 Cos\ \theta_i\ Sin\ \theta_i - Y_0 Z_i - m_0 Z_i^2\}}{\sqrt{D_i}} + Z_i Sin\ \theta \text{----------------[57]}$$

$$\frac{\partial R_i}{\partial A} = \frac{\{Z_i^2 + Z_0^2 - 2 Z_0 Z_i\}}{2\sqrt{D_i}} \text{--[58]}$$

$$\frac{\partial R_i}{\partial Z_0} = \frac{\{Z_0 A - A Z_i\}}{\sqrt{D_i}} \text{--[59]}$$

$$where\ D_i = \{X_0 \cos\theta + l_0 Z \cos\theta + Y_0 \sin\theta + m_0 Z \sin\theta\}^2 - \{X_0^2 + 2 X_0 l_0 Z + l_0^2 Z^2 + m_0^2 Z^2 + Y_0^2 + 2 Y_0 m_0 Z - A Z^2 - A Z_0^2 + 2 A Z_0 Z\}$$

$$A = \left(\frac{c}{b}\right)^2 \quad thus, A \geq 0$$

Since the constant A in the nonlinear equation for a cone should be greater than zero, the SAS program offers a method to bound A's range by setting it to be greater or equal to zero. The algorithms of SAS provides flexibility to the user through the proposition of a range for each unknown constant to change the initial guess value. However, this increases the computational time.

After obtaining the values of the unknown constants, the program appended in Appendix B-2 is attached to calculate Td, angle of the substitute cone , α_{max} , α_{min} and conicity.

6.2. The Objective and constraints for the Nonlinear Minimum Zone Method

The mathematical expression of the nonlinear minimum zone for nonlinear cone is developed. The available software and algorithms suitable to solve the above problem are then found.

6.2.1. Formulation of the Nonlinear Minimum Zone Problem

The basic concept is the same as used in the linear minimum zone approximation. The only difference is that the radius equation is expressed in a nonlinear form in the constraint equations. Therefore, the equation is reformulated as follows:

minimize hTd

subject to

$$|e_i| \le hTd \text{ --[60]}$$

If equation [14] for cones is substituted into the above constraint, it will be expressed with 6 unknown constants as:

$$\left| r_i - \left\{ (X_0\cos\theta_i + l_0 Z_i\cos\theta_i + Y_0\sin\theta_i + m_0 Z_i\sin\theta_i) + \sqrt{\begin{matrix}(X_0\cos\theta_i + l_0 Z_i\cos\theta_i + Y_0\sin\theta_i + m_0 Z_i\sin\theta_i)^2 \\ -(X_0^2 + 2X_0 l_0 Z_i + l_0^2 Z_i^2 + Y_0^2 + m_0^2 Z_i^2 + 2Y_0 m_0 Z_i - AZ_i^2 - AZ_0^2 + 2AZ_0 Z_i)\end{matrix}} \right\} \right| \le hTd$$

The above constraint is divided into:

Constraint 1

$$hTd + ((X_0\cos\theta_i + l_0 Z_i\cos\theta_i + Y_0\sin\theta_i + m_0 Z_i\sin\theta_i) + \sqrt{\begin{matrix}(X_0\cos\theta_i + l_0 Z_i\cos\theta_i + Y_0\sin\theta_i + m_0 Z_i\sin\theta_i)^2 \\ -(X_0^2 + 2X_0 l_0 Z_i + l_0^2 Z_i^2 + Y_0^2 + m_0^2 Z_i^2 + 2Y_0 m_0 Z_i - AZ_i^2 - AZ_0^2 + 2AZ_0 Z_i)\end{matrix}}) \geq r_i$$

--[61]

$$-hTd + ((X_0\cos\theta_i + l_0 Z_i\cos\theta_i + Y_0\sin\theta_i + m_0 Z_i\sin\theta_i) + \sqrt{\begin{matrix}(X_0\cos\theta_i + l_0 Z_i\cos\theta_i + Y_0\sin\theta_i + m_0 Z_i\sin\theta_i)^2 \\ -(X_0^2 + 2X_0 l_0 Z_i + l_0^2 Z_i^2 + Y_0^2 + m_0^2 Z_i^2 + 2Y_0 m_0 Z_i - AZ_i^2 - AZ_0^2 + 2AZ_0 Z_i)\end{matrix}}) \leq r_i$$

The second constraint is derived from equation [61] itself. The constraint means that to express the radius of cone, the inside value of root can not be negative. The second constraint has the following form:

Constraint 2 is expressed as:

$$\begin{matrix}(X_0\cos\theta_i + l_0 Z_i\cos\theta_i + Y_0\sin\theta_i + m_0 Z_i\sin\theta_i)^2 \\ -(X_0^2 + 2X_0 l_0 Z_i + l_0^2 Z_i^2 + Y_0^2 + m_0^2 Z_i^2 + 2Y_0 m_0 Z_i - AZ_i^2 - AZ_0^2 + 2AZ_0 Z_i) \geq 0\end{matrix}$$

----[62]

The third constraint is derived similar to the derivation of equation [14]. The radial coordinates of the reference cone is restricted from having a negative value in the Cylindrical Coordinate System. Thus, the third constraint is:

Constraint 3 is written as:

$$(X_0\cos\theta_i + l_0 Z_i\cos\theta_i + Y_0\sin\theta_i + m_0 Z_i\sin\theta_i) + \sqrt{\begin{matrix}(X_0\cos\theta_i + l_0 Z_i\cos\theta_i + Y_0\sin\theta_i + m_0 Z_i\sin\theta_i)^2 \\ -(X_0^2 + 2X_0 l_0 Z_i + l_0^2 Z_i^2 + Y_0^2 + m_0^2 Z_i^2 + 2Y_0 m_0 Z_i - AZ_i^2 - AZ_0^2 + 2AZ_0 Z_i)\end{matrix}} \geq 0 \text{----[63]}$$

The fourth constraint is extracted from the first constraint and the geometry of the radial error in the cone. hTd, the absolute value of error between the measured points and substitute cone, must not be less than zero in cylinder coordinates. The constraint will be:

Constraint 4 is shown as:

$$hTd \geq 0 \text{--[64]}$$

Since the first three constraints are repeated for each of the seven measured points, a total of twenty two constraints result, including the fourth constraint.

6.2.2. Software and Algorithm

GINO, a nonlinear optimization software developed by The Scientific Press, is utilized to find the value of half of Td and the six unknown constants. Since the GINO software is not a package to find a globally optimized point in all ranges of the

Cylindrical Coordinate System, it might find a local optimum rather than a global optimum, leading us to a wrong optimized result. However, if GINO finds a global optimal point as a local optimal point, the values are just what the Minimum Zone Method should find. Therefore, the answer that we can get from Gino is highly dependent on our initial guess values for the six unknown constants. To get more accurate information, the number of trials should be increased. Also, the problem to solve should be understood well, and trials with different search techniques available in GINO can help to obtain a more exact answer.

A version of the generalized reduced gradient, so-called GRG, is used in GINO software to solve a nonlinear optimization problem. The GRG algorithm is presented in Liebman (1986). The Nonlinear Minimum Zone Method using GINO is referred to as GRG after this point.

GINO uses two different methods to find a search direction; the Broyden-Fletcher-Goldfarb-Shanno (BFGS) Quasi Newton Method and the Conjugate Gradient Method. The BFGS is used as a default, but the other method is a good choice in the case of many variables used in the constraints and objective function. In the Conjugate Gradient Method, there are five different sub-methods according to searching techniques; Flecher-Reeves, Polak-Ribiere, Perry, 1-step Davidon-Fletcher-Powell Method (1-step DFP), and 1-step Broyden-Fletcher-Goldfarb-Shanno Method (1step

BFGS). These the five Conjugate Gradient methods and the BFGS Quasi Newton Method are used to find optimal constants in the nonlinear equation for a cone.

A table of the direction search formula of the most effective conjugate gradient methods is offered in Liebman (1986) as:

Table 3. The Methods to Find Search Directions in the Conjugate Gradient Method (Liebman, 1986)

Method name	The formula to find search directions
FLETCHER-REEVES	$D = -\nabla F + a_1 d_{prev}$
POLAK-RIBIERE	$D = -\nabla F + a_2 d_{prev}$
1-STEP BFGS	$D = -\nabla F + a_3(a_4 s + a_5 y)$

$$where,\ a_1 = \nabla F^T \nabla F / (\nabla F_{prev})^T (\nabla F_{prev})$$
$$a_2 = \nabla F^T y / (\nabla F_{prev})^T (\nabla F_{prev})$$
$$a_3 = 1 / s^T y$$
$$a_4 = -(1 + y^T y / s^T y) s^T \nabla F + y^T \nabla F$$
$$a_5 = s^T \nabla F$$

If the line searches are exact, the first two techniques generate down hill search directions along which the objective function decreases. Each search technique is used every time the guess values are changed. The source code of a program which is developed in order to generate the nonlinear constraints and objective function is included Appendix B.

To get proper values of unknown constants and the objective function with minimum zone, several parameters are changed. For example, the parameter to restrict maximum iteration without converging is changed from a default of 10 to 100. The default fractional change and the stopping criteria in the objective function, are changed. With these changes, GINO is used to solve the nonlinear Minimum Zone Method for a cone.

6.3. The Result using the Nonlinear Optimization Technique

Additional discussion on obtaining guess values may help in understanding the procedure for nonlinear optimization techniques used in this thesis.

First, the bounding techniques about unknown constants is applied. The bounding technique can be summarized as:

Step1. Give any values to the unknown constants except one variable. Allocate two guess values to that variable.

Step 2. Evaluate the sum of square errors for the two settings of guess values with SAS. Choose the better guess value based on the sum of square errors. Repeat three or four times. Find the two best guess values among them.

Step 3. Repeat Step 2 with the other unknown constants.

The above procedure can be beneficial in finding the boundary of unknown constants in the case of one or two variables in the objective and constraints.

However, it was recognized as time consuming for the cone equation with six unknown constants.

Second, to save evaluation time, this previous procedure is revised. Since the cone generated by a nonlinear equation may be close to the cone generated by a linear equation, the constants estimated from the results of the linear optimization in the previous chapter can be used as guess values for the unknown constants in the nonlinear equation. Using the previous chapter's results which are the results of the Least Squares Method and Simplex Method in a tilted cone, only two constants A and Z_0 remain unknown. Steps 1, 2, and 3, outlined earlier, are performed for these two constants. Hence, the boundaries for these constants can be determined. Any values within these boundaries can be used as guess values for these unknown constants A and Z_0.

Third, equations [51] and [52] are used to find guess values for A and Z_0. These equations use the values of X_0, l_0, m_0 and Y_0 from the previous chapter's results of the Linear Least Squares Method and the Linear Minimum Zone Method. A point on their surface generated by equation [5] is used as X, Y, and Z for equation [52].

Using these different combinations of guess values for the unknown constants, SAS and GINO are used to find the minimum sum of square errors and the minimum tolerance zone. After obtaining the results from SAS and GINO for different combinations of guess values, the results are compared with each other. SAS and GINO give the values of unknown constants, as well as the value of the objective function. With these values of unknown constants, the Td, conicity, α_{max}, α_{min} and vertex angle of the cone are found by a developed program with Matlab. For example, six trials and data collected from a preliminary test are illustrated in Appendix A.

Compared with the nonlinear optimization using SAS, the nonlinear optimization using GINO seems less sensitive to guess values of unknown constants in the nonlinear equation. The reason may be that the nonlinear optimization in GINO is a constrained nonlinear optimization, but SAS is a unconstrained nonlinear optimization.

Table 4. The Comparison of the Radial Deviations of Each Measured Point and Sum of Square Errors unit: inch

	TILTED CONE	
	LEAST SQUARE (GAUSS)	MINIMUM ZONE (GRG)
Errors of each measured points	0.00030810 0.00027840 0.00010967 -0.00054425 -0.00046273 0.00017289 0.00013791	0.00037010 0.00036947 0.00036973 -0.00036846 -0.00036899 0.00036998 0.00037024
Sum of error squres	0.0000007436969145	0.00000096

Table 4 illustrates the same trend as the linear optimization in the previous chapter. If the sums of square errors are compared, the Least Squares Method shows smaller values than the Minimum Zone Method.

The values given in Table 5 are from the results of the developed program using the results of SAS and GINO. In the case of the Nonlinear Least Squares Method, the results of 6th trial of the Gauss Method are used with the results showing a minimum sum of square errors. The reason that the 6th trial of the Gauss Method shows the smallest sum of square errors might be due to the guess values. The guess values are based on the results of linear optimization, as mentioned before, and the guess value for Z_0 is given as a range instead of as a fixed value.

Table 5. Results of the Least Squares Method and Minimum Zone Method Using the Nonlinear Optimization

unit: inch for Td and conicity, angle for other

	Td	conicity	angle of substitute cone	max. angle	min. angle
LSQ	0.00085235	0.00083963	20.14472	20.161835	20.114486
GRG	0.00073923	0.00072819	20.099637	20.120177	20.079165

where the model of the cone is a tilted cone. LSQ is the Least Squares Method and GRG is the Minimum Zone Method using the General Reduced Gradient Algorithm.

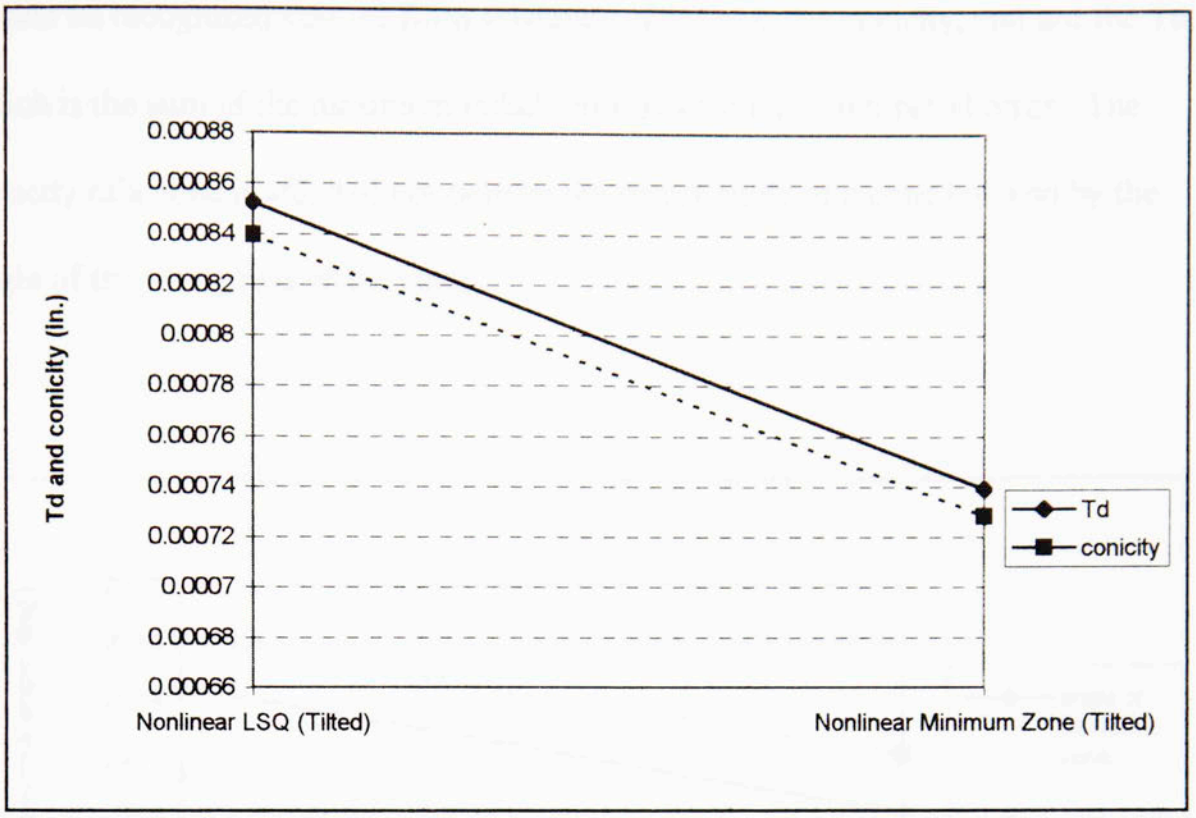


Figure 19. The comparison of Td and conicity with a nonlinear equation.

Figure 19 confirms the most important trend in the nonlinear optimization. As Figure 16 showed in the previous chapter, the Td and conicity using the Nonlinear Minimum Zone Method are approximately 13.3% better than those obtained with the Nonlinear Least Squares Method. Due to this, it is proposed that using the Minimum Zone Method always results in the smaller tolerance zone than while using the Least Squares Method. In other words, the Least Squares Method can not guarantee the minimum tolerance zone using either a linear or nonlinear equation.

Another interesting observation concerns the Td and conicity for each method. In each case of using linear and nonlinear equations, the conicity is less than Td. It should be recognized that the form tolerance of cones is the conicity, and not the Td which is the sum of the maximum radial error and the minimum radial error. The conicity of a cone is affected not only by the vertex angle of a cone but also by the angle of the tilting axis of the cone.

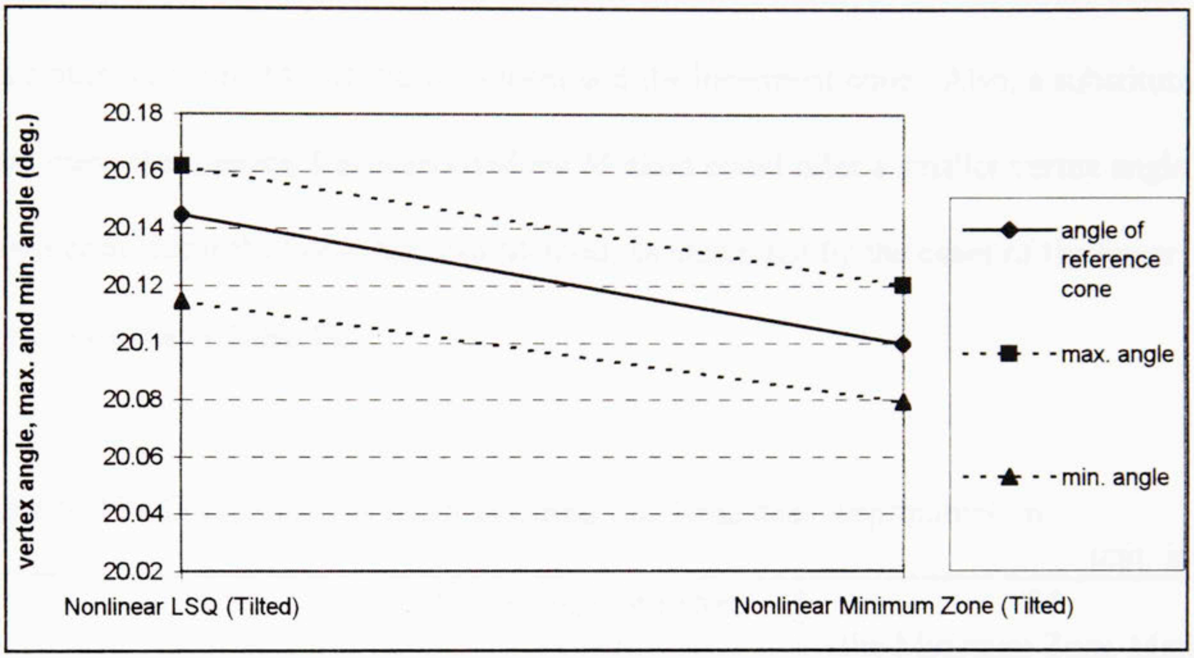


Figure 20. The comparison of angles generated with a nonlinear equation

The trends in Figure 20 are similar to those obtained in the linear cone. The vertex angle using the Minimum Zone Method with a nonlinear equation lies in the middle of α_{max} and α_{min}. However, the vertex angle using the Least Squares Method is not in the middle of them. Since a circle having a minimum zone should lie in the middle of α_{max} and α_{min}, the vertex angle of the cone having a minimum zone should also lie in the middle of the outermost cone and the innermost cone. This characteristic of a cone, having a minimum zone, can be derived from the fact that a section of a cone at a specific Z_i is a circle.

The above observation means that the Least Squares Method might not guarantee the determination of Td and conicity with a minimum tolerance zone. However, in the case of the General Reduced Gradient Method, the substitute cone lies almost in the middle of the outermost and the innermost cone. Also, a substitute cone using the General Reduced Gradient Method could offer a smaller vertex angle than a cone using the Least Squares Method, as illustrated by the cases of the linear approximation in Table [2].

Table 6. The Comparison of Error in Linear and Nonlinear Approximation

unit: inch

	Sum of Square Errors (LSQ)	Td (the Minimum Zone Method)
Linear Equation	0.0000007341181	0.0007352
Nonlinear Equation	0.0000007436969	0.0007392
the Percentage of Difference	1.288%	0.541%

The sum of square errors using a linear equation is approximately 1.29% less than that using a nonlinear equation. The Minimum Zone using a linear equation is 0.54% less than that, using a nonlinear equation. These results are similar to Chetwynd's (1979) circle work and Lin's (1995) cylinder estimation.

Hence, Table 6 confirms the above articles in that the linear equations using limacon approximation for circular objects such as circles, cylinders and cones may express surfaces of circular objects better than the nonlinear equations when their center axis is well-aligned with the Z axis.

Chapter 7

CONCLUDING REMARKS

This thesis has presented linear and nonlinear equations for cones. Based on the two representations, a comparison between the Minimum Zone Method and the Least Squares Method is attempted. The following summarizes this works and assumptions of this thesis:

1. Linear and nonlinear equations for tilted cones are developed consistent with those employed by the Coordinate Measuring Machine (CMM) used Browne and Sharpe (Model: PFx 454 System).
2. Linearization of a nonlinear equation of the cone is applied. The Least Squares Method (LSM) and the Simplex Method using the linearized equation for cones are employed for calculation of Td and Conicity. The Least Squares Method and minimum zone calculation are also done for a non-linear representation of the cone.
3. CMMs only analyze the vertex angle of a substitute cone, and do not focus on determination of angles such as α_{max} and α_{min}, that are computed here.
4. The nonlinear least square optimization techniques such as Gradient, Gauss, DUD and Marquadt are then used to find a substitute cone using the Nonlinear Least Squares Method. The constrained nonlinear optimization technique, the General

Reduced Gradient Method, is applied for the analysis of the form tolerance of cones using the Nonlinear Minimum Zone Method.

Some preliminary observations are made based on a sample set of data obtained using the above CMM and fitted to validate the equations. There are:

1. The Minimum Zone Method yielded a smaller Td and conicity than the Least Squares Method in Figure 21. This trend has been proven before by many researches while evaluating cylindricity, circularity, flatness, and straightness. It is difficult to make conclusive statement in the conicity case, without additional data sets. Further work must concentrate on careful experimental design and analysis to verify this claim.
2. The preliminary test and analysis confirmed that the results using linear equations are more precise than the results using nonlinear equations in the case of the tilted cones of which the eccentricity ratio is less than 0.01, as observed by Chetwynd (1985) and Lin (1995). But these differences are very small (0.541%) to make any conclusions. It would be interesting to investigate this issue with various data sets. Development of data sets and selection of representative data and detailed experimental designs is the subject of future work.
3. A comparison of the Least Squares Method and the Simplex Method in different types of cones using the linear equation and nonlinear equation is presented for assessing angles, height and form tolerances. Figure 21 through Figure 23 may be used to summarize the comparison. Since Td and conicity using the nonlinear and linear equations for the tilted cone are very close to each other, the height and

vertex angle should be close to each other. The Figures 22 and 23 show that the tilted cones using the linear and nonlinear equations have similar height and vertex angle. Since the conicity using the moved cone in Figure 21 are not close to the conicity from the Coordinate Measuring Machine (CMM), we may conclude that the type of cone used in CMM is the tilted cone as the definition of ISO. Figure 22 and Figure 23 show that the increase of height is directly related to the decrease of the vertex angle.

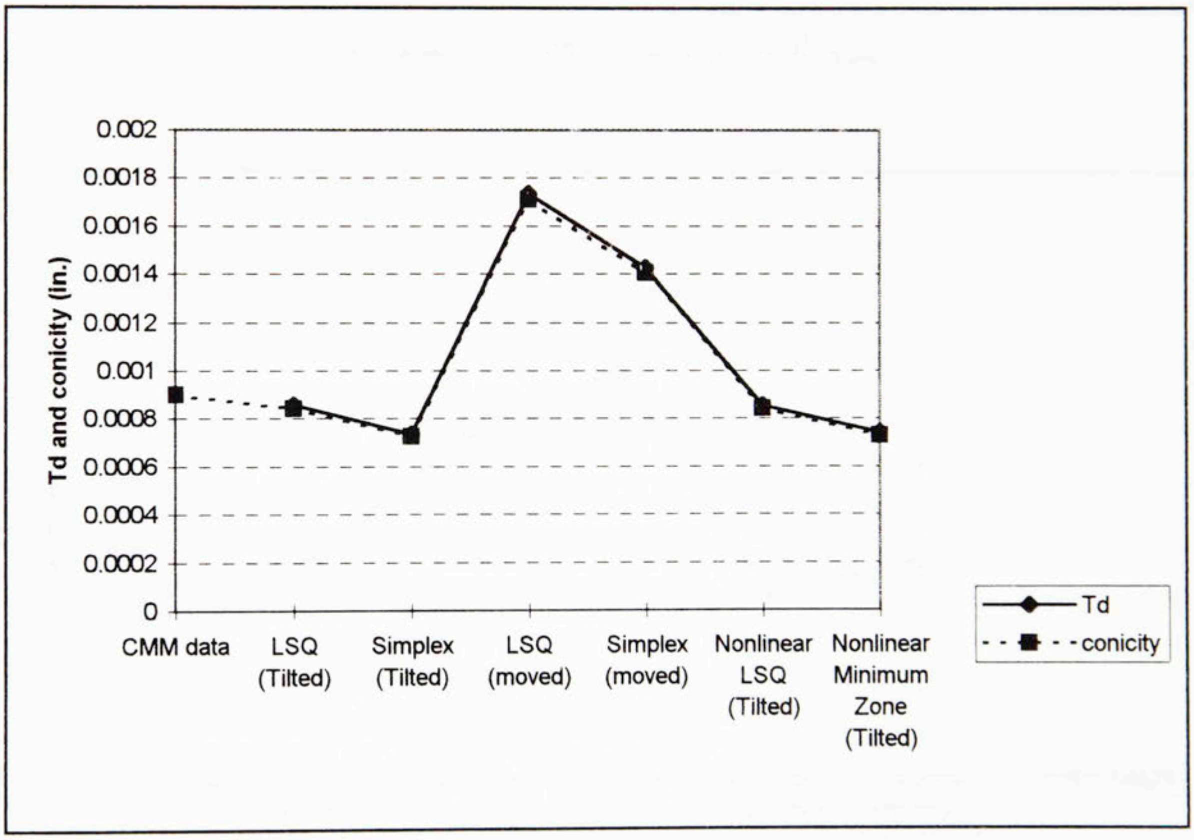


Figure 21. Td and conicity with the linear and nonlinear equations.

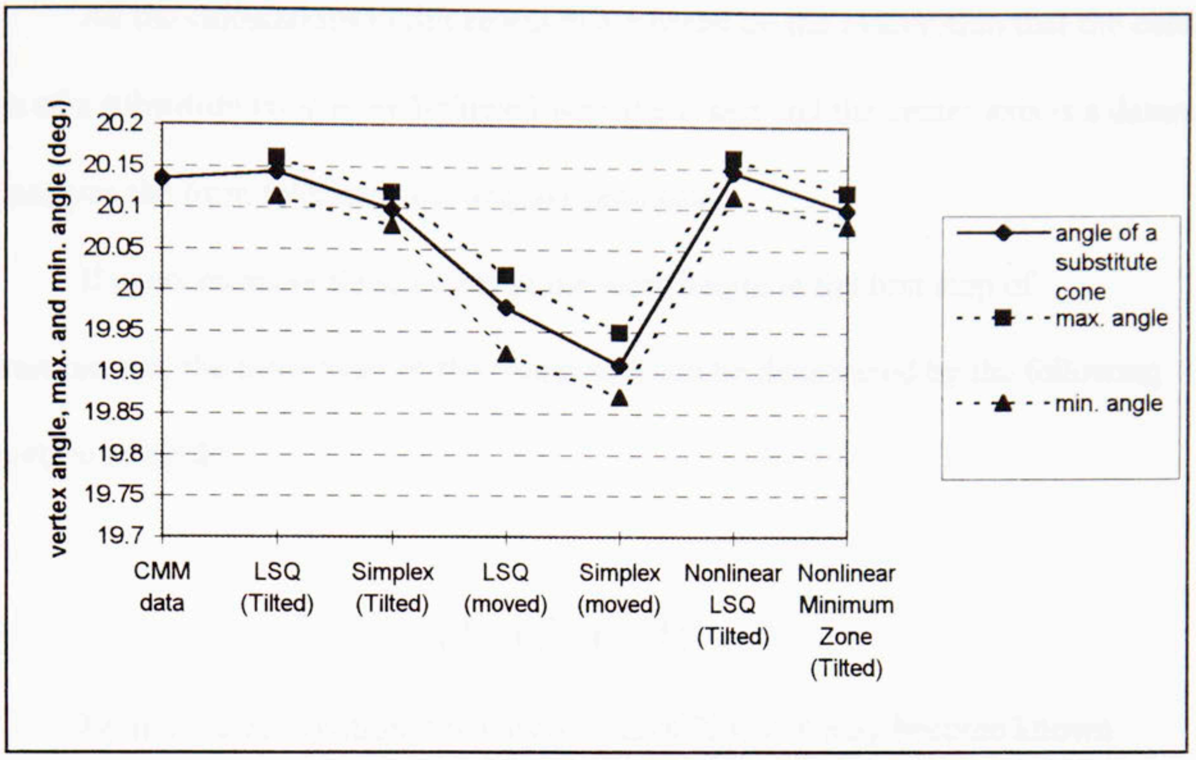


Figure 22. Vertex angle, $\alpha_{\max}$ and $\alpha_{\min}$

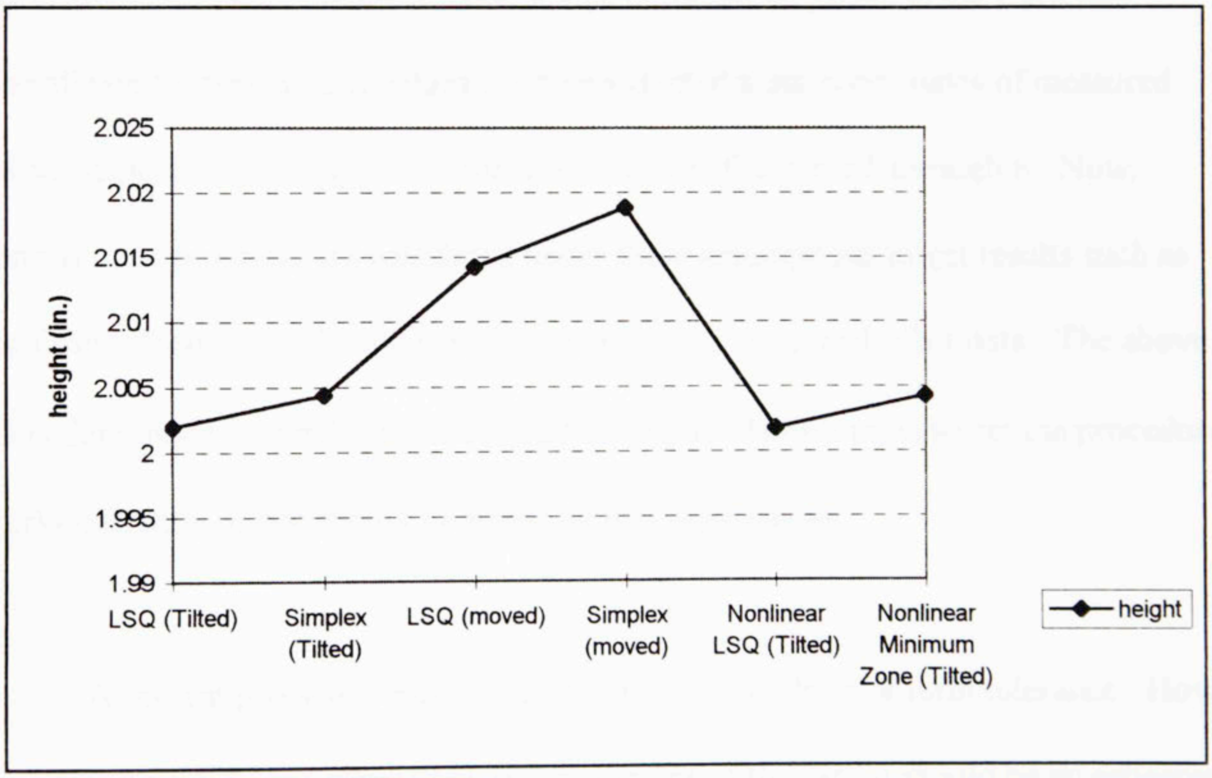


Figure 23. The height in the Z direction by each method

All the calculations in this research are based on the assumption that the center axis of a substitute cone is well-aligned with the Z axis and the center axis is a datum to analyze the form tolerance and angular problems.

If users measure three points at the same height in the first step of measurement, the movement of the center axis can be determined by the following equation of circle:

$$(X-a)^2 + (Y-b)^2 = r^2$$

By the above equation, the movements of X and Y may become known variables. The same Z value of the three points is considered to be the movement of the center axis. After obtaining coordinates of measured points in the Cartesian Coordinate System, a, b, Z values can be subtracted from coordinates of measured points respectively to apply the assumption used in Chapters 3 through 6. Now, general cone problems are calculated under these assumptions to get results such as the form tolerance, angle of a substitute cone, α_{mzx}, $\alpha_{\min}$ and other data. The above procedure might be similar to those used in CMMs. However, whether the procedure works correctly or not should be tested by real experiments.

A second problem concerns the datum in the analysis of form tolerance. How the conicity and Td are affected by the movement of the datum should be investigated in the case of cones. Also, when the datum has a tolerance such as a position tolerance, the conicity of a cone might be affected by the accumulation of tolerances.

The formulation of a cone including the above consideration should be investigated in the future.

Extensions to three dimensions must also be investigated. Choice of proper measuring points is yet another area of research that needs adequate consideration in CMM research. Proper choice of data impacts the inspection and data- fitting significantly. To state otherwise, the same object, sampled differently can yield different data which when fitted can result in different minimum zones and different least square fits. This in itself points do the need for more comprehensive data collection and analysis for the future.

The another method for the Minimum Zone Method should be investigated to be compared with the results of the Minimum Zone Method in this thesis. The formulas can be written as :

Using a linear equation for cones

$$\begin{aligned}
&Min(Max(e_i) - Min(e_i)) \\
&where\ e_i = r_i - (R_o + S\,Z_i + X_0 Cos\,\theta_i + l_0 Z_i Cos\,\theta_i + Y_0 Sin\,\theta_i + m_0 Z_i Sin\,\theta_i) \\
&Subject\ to \\
&R_0 + S\,Z_i \geq 0 \\
&R_0 \geq 0
\end{aligned}$$

Using a nonlinear equation for cones

$$MIN\ (MAX(e_i) - MIN(e_i))$$

Where $e_i = r_i - R_i$

$$R_i = \{X_0 \cos\theta_i + l_0 Z_i \cos\theta_i + Y_0 \sin\theta_i + m_0 Z_i \sin\theta_i\}$$
$$+ \sqrt{D_i}$$
$$where,\ D_i = \{X_0 \cos\theta_i + l_0 Z_i \cos\theta_i + Y_0 \sin\theta_i + m_0 Z_i \sin\theta_i\}^2$$
$$- \{X_0^2 + 2\,X_0 l_0 Z_i + l_0^2 Z_i^2 + Y_0^2 + m_0^2 Z_i^2 + 2\,Y_0 m_0 Z_i - A Z_i^2 - A Z_0^2 + 2 A Z_0 Z_i\}$$
$$and\ D_i \geq 0$$

Subject to

$$\{X_0 \cos\theta_i + l_0 Z_i \cos\theta_i + Y_0 \sin\theta_i + m_0 Z_i \sin\theta_i\}^2$$
$$- \{X_0^2 + 2\,X_0 l_0 Z_i + l_0^2 Z_i^2 + Y_0^2 + m_0^2 Z_i^2 + 2\,Y_0 m_0 Z_i - A Z_i^2 - A Z_0^2 + 2 A Z_0 Z_i\} \geq 0$$
$$R_i \geq 0$$

REFERENCE

[1]. ASME (1988), Dimensioning and Tolerancing, ASME, New York.

[2] Bazaraa, M. S., Jarvis,J. J. and Sherali, H. D. (1990), Linear programming and Network flows, 2nd Ed., John Wiley & Sons,p91-95, New York.

[3] Carr, K., and Ferreira, P. (1995), "Verification of form tolerances Part I: Basic issues, flatness, and straightness," Precision Engineering, Vol. 17. No. 2, PP. 131-143.

[4] Carr, K., and Ferreira, P. (1995), "Verification of form tolerances Part II: Cylindricity and straightness of a median line", Precision Engineering, Vol. 17. No. 2, PP. 144-156.

[5] Chetwynd, D. G. (1979), "Roundness measurement using limacons" , Precision Engineering, Vol. 1, PP.137-141.

[6] Chetwynd, D. G. (1985),"Applications of linear programming to engineering metrology", Proceedings of the Institution of Mechanical Engineers., Vol. 199, No. B2, PP. 93-100.

[7] De Bruin, W. and Meijer, J. (1981), "Some results of flatness measurement", Precision Engineering, Vol. 3, PP. 23-26

[8] De Bruin, W. and Meijer, J. (1981), "Determination of flatness from straightness measurements and characterization of the surface by four parameters", Precision Engineering, Vol. 3, PP. 7-22

[9] Dowling, M. M., Griffin, P. M., Tsui, K. -L., and Zhou, C. (1995), "A Comparison of the Orthogonal least Squares and Minimum Enclosing Zone Methods for Form Error Estimation", Manufacturing Review, Vol. 8, No. 2., PP. 120-137

[10] Elmaraghy, W. H. , Elmaragy, H. A. and Wu, Z. (1990), "Determination of actual geometric deviations using coordinate measuring machine Data", Manufacturing Review, Vol. 3, No. 1, March., PP. 32-39.

[11] El-Sayed, A. A. and El-Hakeem, H. M. (1986), "Uncertainties of two analytical methods used for the determination of straightness error", International Journal of Production Research, Vol. 24, No. 5, PP. 1193-1202

[12] Henzold (1996), G., Handbook of Geometrical Tolerancing: design, manufacturing and inspection, John Wiley & Sons, New York.

[13] Hong, J.-T. , Liao, Y.-S. and Fan, K.-C. (1991), "A computational Geometry Approach to minimum zone evaluation of straightness", Journal of the Chinese Institute of Engineers, Vol. 14, No. 5, PP. 463-470.

[14] Huang, S. T. , Fan, K. C. and Wu, J. H. (1993), "A New Minimum Zone Method for Evaluating Straightness Errors", Precision Engineering, Vol. 15, No, 3, July., PP. 25-32.

[15] Huang, S. T., Wu, J. H., and Fan, K. C. (1993), "A minimum zone method for evaluating flatness error of gage blocks measured by phase-shifting interferometry", Journal of the Chinese Institute of Engineers, Vol. 16, No. 5, p. 641-650.

[16] Kanada, T. (1995),"Evaluation of spherical form errors-Computation of sphericity by means of minimum zone method and some examinations with using simulated data", Precision Engineering, Vol. 17, PP. 281-289.

[17] Kanada, T. and Suzuki S. (1993), "Evaluation of minimum zone flatness by means of nonlinear optimization techniques and its verification", Precision Engineering, Vol. 15 , PP. 93-99.

[18] Le, V.-B and Lee, D. T. (1991), "Out-of-Roundness Problem Revisited (1991)", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 3, March., PP. 217- 223.

[19]Lee, G., Mou, J., and Shen, Y. (1997), "Sampling Strategy Design for Dimensional Measurement of geometric features using Coordinate Measuring Machine", International Journal of Machining Tools Manufacturing, Vol. 37, no.7, PP. 917-934

[20] Liebman, J., Lasdon, L., Schrage, L., and Waren, A. (1986), Modeling and Optimization with GINO, The Scientific Press, P142-182,South San Francisco.

[21] Lin, S. -S., Varghese, P., Zhang, C., and Wang, H. -P. B. (1995), " A comparative analysis of CMM Form-Fitting Algorithms", Manufacturing Review, Vol. 8, No. 1, March., PP. 47-58.

[22] Menq, C. -H., Yau, H. -T., Lai, G. -Y. (1992), "Automated Precision Measurement of Surface Profile in CAD-Directed Inspection", IEEE Transactions on Robotics and Automation, Vol. 8, No. 2, p. 268-278.

[23] Menq, C. -H., Yau, H. -T., Lai, G. -Y., and Miller, R. A. (1990), "Statistical evaluation of form tolerances using discrete measurement data", in Advances in integrated product design and manufacturing:, presented at the Winter Annual Meeting of the ASME, Dallas, Texas, Nov., PP. 25-30.

[24] Mou, J. and Liu, C. R., "A method for enhancing the accuracy of CNC machine tools for on-machine inspection, Journal of Manufacturing Systems", Vol. 11, no.4, PP.229-237

[25] Mou, J., Domez, M. A. and Cetinkunt, S. (1995), "An adpative methodology using features-based analysis techniques for machine error correction: part II-Experimental Verification", ASME Transactions, Journal of Engineering for Industry, Vol. 117, no.4, PP. 591-600

[26] Murthy, T. S. R. and Abdin, S. Z. (1980), "Minimum zone evaluation of surfaces", International Journal of Machine Tool Design and Research, Vol. 20, PP. 123-136.

[27] Olmsted, Hubbell, M. J. (1947), Solid Analytic Geometry, D. Appleton-Century Company, Inc., New York.

[28] Orady, E. A., Li, S., and Chen, Y. (1996), "Evaluation of Minimum Zone Strainghtness by nonlinear optimization method", Manufacturing Science and Engineering, MED-Vol. 4., PP. 413-423.

[29] Rao, S. S. (1996), Engineering Optimization, Theory and practice ,second Edition, A Wiley-Interscience Publication., New York.

[30] Ravindran, A. , Phillips D. T. , and Solberg J. J. (1987), Operations Research Principles and Practice, 2nd Ed, John Wiley & Sons, P35-37., New York.

[31] Reklaitis, G. V., Ravindran, A. , and Ragsdell, K. M. (1983), Engineering Optimization Methods and Applications, Wiley Interscience, New York.

[32] Roy, U. and Zhang, X. (1994), "Development and Application of Voronoi Diagrams in the assessment of roundness error in an industrial environment", Computers and Industrial Engineering, Vol. 26, No. 1, PP.11-26.

[33] SAS (1990), SAS/STAT User's Guide, 4th Ed., SAS Institute Inc., Vol.2, PP. 1136-1193.

[34] Schrage, L. (1991), Lindo User's manual, An International Thomson Publishing Company, New York.

[35] Shunmugam, M. S. (1986), "Establishing Reference Figures for Form Evaluation of Engineering Surfaces", Journal of Manufacturing Systems, Vol. 10., No. 4, PP. 314-321.

[36] Shumugam, M. S. (1986), "On assessment of geometric errors" , International Journal of Production Research., Vol. 24, No. 2, PP. 413-425.

[37] Shunmugam, M. S. (1987), "New approach for evaluating form errors of engineering surfaces", Computer-Aided Design, Vol. 19, No. 7., PP. 368-374.

[38] Trabnd, M.T., Joshi, S., Wysk R. A., and Cavalier, T. M. (1989), "Evaluation of Straightness and Flatness Tolerances using the minimum zone", Manufacturing Review, Vol. 2, No. 3, September., PP.189-195.

[39] Wang, Y. (1992), "Minimum Zone Evaluation of Form Tolerances", Manufacturing Review, Vol. 5, No. 3, September., PP. 213-220.

[40] Woo, T. C., Lian, R., Lee, N. K. and Hsieh, C. C. (1995), "Efficient Sampling for Surface Measurment", Journal of Manufacturing Systems, Vol. 14, no. 5, PP. 345-354

Appendix A. The outputs of the Nonlinear Least Squares Method and the Nonlinear Minimum Zone Method

There are no constraints on A in the 1st and 2nd trial in the SAS program. The other trials include the bound statement for constant A.

Table 1. The result of Td , conicity and the angle of a substitute cone

	1st Trial		2nd Trial		3rd Trial	
	Td and conicity	angle of substitute cone	Td and conicity	angle of substitute cone	Td and conicity	angle of substitute cone
Gradient	Td: 0.05331794 conicity: 0.05236780	21.9595178 6	Td: 0.05311694 conicity: 0.05247091	18.2283642 4	Td: 0.00085235 conicity: 0.00083963	20.1506753 3
Gauss	Td: 0.00085235 conicity: 0.00083963	20.1506753 3	Td: 0.00085235 conicity: 0.00083964	20.1506756 5	Td: 0.00085235 conicity: 0.00083963	20.1506753 3
DUD	Td: 0.00083696 conicity: 0.00082447	20.1542276 5	Td: 0.00083848 conicity: 0.00082596	20.1546838 1	Td: 0.00083696 conicity: 0.00082447	20.1542276 5
Marquadt	Td: 0.00085235 conicity: 0.00083964	20.1506756 5	Td: 0.00085235 conicity: 0.00083964	20.1506756 5	Td: 0.00085235 conicity: 0.00083964	20.1506756 5

	4th Trial		5th Trial		6th Trial	
	Td and conicity	angle of substitute cone	Td and conicity	angle of substitute cone	Td and conicity	angle of substitute cone
Gradient	Td: 0.05311694 conicity: 0.05247091	18.2283642 4	Td: 0.05434306 conicity: 0.05337062	21.9527306 8	Td: 0.05296937 conicity: 0.05232189	18.2152321 2
Gauss	Td: 0.00085235 conicity: 0.00083964	20.1506756 5	Td: 0.00085235 conicity: 0.00041939	20.1506756 5	Td: 0.00085235 conicity: 0.00083964	20.1506756 5
DUD	Td: 0.00083848 conicity: 0.00082596	20.1546838 1	Td: 0.00085168 conicity: 0.00083897	20.1507928 2	Td: 0.00291621 conicity: 0.00287277	20.1814855 5
Marquadt	Td: 0.00085235 conicity: 0.00083964	20.1506756 5	Td: 0.00085235 conicity: 0.00083964	20.1506756 5	Td: 0.00085235 conicity: 0.00083964	20.1506756 5

	7TH TRIAL		8TH TRIAL	
	Td and conicity	angle of substitute cone	Td and conicity	angle of substitute cone
DUD	Td: 0.00083886 conicity: 0.00082634	20.15361072	Td: 0.00294802 conicity: 0.00290411	20.18122566
GAUSS	Td: 0.00085235 conicity: 0.00083964	20.15067565	Td: 0.00085235 conicity: 0.00083964	20.15067565
GRADIENT	Td: 0.05342035 conicity: 0.05246853	21.95920631	Td: 0.05427906 conicity: 0.05330775	21.95298152
MARQUARDT	Td: 0.00085235 conicity: 0.00083964	20.15067565	Td: 0.00085235 conicity: 0.00083964	20.15067565

Table 2. The sum of square errors of the Gauss Method

GAUSS	1ST TRIAL	2ND TRIAL	3RD TRIAL
Errors of each measured points	0.00030810 0.00027840 0.00010966 -0.00054425 -0.00046274 0.00017289 0.00013791	0.00030810 0.00027840 0.00010966 -0.00054425 -0.00046274 0.00017289 0.00013790	0.00030810 0.00027840 0.00010966 -0.00054425 -0.00046274 0.00017289 0.00013791
Sum of square errors	0.00000074370397	0.0000007437012178	0.0000007437039759
GAUSS	4TH TRIAL	5TH TRIAL	6TH TRIAL
Errors of each measured points	0.00030810 0.00027840 0.00010966 -0.00054425 -0.00046274 0.00017289 0.00013790	0.00030810 0.00027839 0.00010966 -0.00054426 -0.00046274 0.00017288 0.00013790	0.00030810 0.00027840 0.00010967 -0.00054425 -0.00046273 0.00017289 0.00013791
Sum of square errors	0.00000074370121	0.000000743703773	0.0000007436969145

GAUSS	7th trial	8th trial
Errors of each measured points	0.00030810 0.00027840 0.00010966 -0.00054425 -0.00046274 0.00017289 0.00013790	0.00030810 0.00027840 0.00010966 -0.00054425 -0.00046274 0.00017289 0.00013790
Sum of square errors	0.0000007437012178	0.000000743712178

Table 3. The sum of square errors of the Gradient Method

GRADIENT	1^{ST} TRIAL	2^{ND} TRIAL	3^{RD} TRIAL
Errors of each measured points	-0.02002275 -0.01985158 -0.02067165 -0.02389108 -0.02352004 -0.02640358 -0.03346636	0.01999384 0.01986027 0.01988234 0.02228427 0.02223708 0.02721784 0.03325667	0.00030810 0.00027840 0.00010966 -0.00054425 -0.00046274 0.00017289 0.00013791
Sum of square errors	0.00416343	0.00402738	0.0000007437039759
GRADIENT	4^{TH} TRIAL	5^{TH} TRIAL	6^{TH} TRIAL
Errors of each measured points	0.01999384 0.01986027 0.01988234 0.02228427 0.02223708 0.02721784 0.03325667	-0.02050073 -0.02030066 -0.02095188 -0.02431888 -0.02402979 -0.02690190 -0.03404240	0.02059817 0.01945451 0.02025415 0.02371212 0.02097617 0.02750169 0.03351486
Sum of square errors	0.00402738	0.00432281	0.00409485

GRADIENT	7^{th} trial	8^{th} trial
Errors of each measured points	-0.02007512 -0.01990915 -0.02072599 -0.02395002 -0.02356880 -0.02646338 -0.03351120	-0.02046780 -0.02026767 -0.02091906 -0.02428637 -0.02399720 -0.02686972 -0.03401139
Sum of square errors	0.00418135	0.00431177

Table 4. The sum of square errors of the DUD Method

DUD	1ST TRIAL	2ND TRIAL	3RD TRIAL
Errors of each measured points	0.00028009 0.00031687 0.00010367 -0.00052009 -0.00048923 0.00016647 0.00014221	0.00027262 0.00032533 0.00010471 -0.00051314 -0.00049743 0.00016428 0.00014369	0.00028009 0.00031687 0.00010367 -0.00052009 -0.00048923 0.00016647 0.00014221
Sum of square errors	0.000000747380019	0.0000007495094564	0.000000747380019
DUD	4TH TRIAL	5TH TRIAL	6TH TRIAL
Errors of each measured points	0.00027262 0.00032533 0.00010471 -0.00051314 -0.00049743 0.00016428 0.00014369	0.00030826 0.00027893 0.00011072 -0.00054342 -0.00046251 0.00016699 0.00014709	0.00099478 -0.00029414 0.00011102 0.00091487 -0.00192143 0.00034451 0.00044750
Sum of square errors	0.000000749509456	0.0000007438270156	0.00000594

DUD	7th trial	8th trial
Errors of each measured points	0.00027429 0.00032071 0.00010578 -0.00051815 -0.00049219 0.00016792 0.00014164	0.00100891 -0.00031191 0.00012689 0.00092954 -0.00193911 0.00032493 0.00046267
Sum of square errors	0.0000007482687512	0.00000608

Table 5. The sum of square errors of the Marquadt Method

MARQUADT	ALL TRIALS GIVE THE SAME RESULTS
Errors of each measured points	0.00030810 0.00027840 0.00010966 -0.00054425 -0.00046274 0.00017289 0.00013790
Sum of square errors	0.0000007437012178

Table 6. The four initial guessed values for the nonlinear minimum zone method and the results of variables

Variable	The 1st set of Initial	The 2nd set of Initial	The 3rd set of Initial	The 4th set of Initial	Answer
X0	0.00103398	0.00075	0.00103398	0.00075	0.000745
Y0	-0.00089145	-0.00084	-0.000891455	-0.00084	-0.000839
l0	-0.00281865	-0.00237	-0.00281865	-0.00237	-0.002367
m0	0.000518612	0.00492	0.000518612	0.00492	0.00492
Z0	-1.08943694	-1.092264	-1.08974648	-1.09207926	-1.091928
A	0.031522003	0.031366	0.031506217	0.031374858	0.031383

Table 7. The Data Set Used in the linear and the non-linear optimization

Measured order	X	Y	Z
1	0.0350	0.2160	0.1416
2	0.0349	-0.2163	0.1416
3	0.2159	0.0383	0.1416
4	0.0541	0.2451	0.3229
5	0.0260	-0.2482	0.3231
6	-0.2968	0.0261	0.5833
7	0.3533	0.0261	0.9123

Appendix B. Mfile

APPENDIX B-1

These programs are performed in Pentium 200MHz and this language is a language offered by Matlab. First, these program should be saved as separate files in the same directory. Second, run Matlab and type mfile's names in the Matlab. Third, follow the instruction offered in the matlab.

Revise.m

```
%This function calculate R value of cone object which is depended on
%height and center of cone section and R value of right cone
%The equation of radius of the tilted cone is that
%Radius=R0+Ram*Zval+X0*cos(theta)+l0*Zval*cos(theta)+Y0*sin(theta)+m0*Zval*sin(theta)

%Input section
[m,n]=size(Data);
Newcoord=zeros(m,n);
A=zeros(6,6);
C=zeros(6,1);

t0=clock;

%data transformation loop from x,y,z coordinate to cylinder coordinate
[TH,R,Z]=CART2POL(Data(:,1),Data(:,2),Data(:,3))
Newcoord(:,1)=R;
Newcoord(:,2)=TH;
Newcoord(:,3)=Z;
theta=TH;

%the calculation of tilted cone
[A,B,C,R0,Ram,X0,l0,Y0,m0,INVE]=tcone(R,Z,theta,m);

%finding optimal points of tilted cone
[Data2,Zver,xyz]=optimal(Data,R0,Ram,X0,l0,Y0,m0);

%find error
[sizez,a2]=size(Z)
Err=zeros(sizez,1);
Rid=zeros(sizez,1);
for c1=1:sizez
Rid(c1,1)=R0+Ram*Newcoord(c1,3)+ X0*cos(Newcoord(c1,2)) + l0*Newcoord(c1,3) *
cos(Newcoord(c1,2))+Y0 * sin(Newcoord(c1,2))+m0*Newcoord(c1,3) * sin(Newcoord(c1,2));
Err(c1,1)=Newcoord(c1,1)-Rid(c1,1);
```

```
end
%find maximum and minimum
maxError=max(Err)
minError=min(Err)

[l1,j1]=size(xyz);
slice=zeros((l1/36),3);

nc=1;
for na=1:36:396
 slice(nc,1)=xyz(na,1);
 slice(nc,2)=xyz(na,2);
 slice(nc,3)=xyz(na,3);
 nc=nc+1;
end

ti='This calculation is for conicity of tilted cone'

Tdti=abs(maxError)+abs(minError)

%coaxility analysis
diam=input('what is the diameter of tolerance');%diam is short for coaxility
 decision=Zver(:,1) .^ 2+Zver(:,2) .^ 2
[de,de1]=size(decision)
for che=1:de
if decision(che,de1)>(diam/2)^2
  s='The center is out of tolerance'
end
end

Rideal(:,3)=Newcoord(:,3);

time=etime(clock,t0)

%equation saving
%data output
file=input('What is the output file name that you want?','s')
fn=fopen(file,'w');
case1='tilted cone with using least squares method';
case2='right cone with using least squares method';

text4='original data';
fprintf(fn,'%s\n',text4);
[m,n]=size(Data);
  for l=1:m
   fprintf(fn,'%12.8f   %12.8f   %12.8f\n',Data(l,1),Data(l,2),Data(l,3));
  end

text23='the converted r theta z of original data';
fprintf(fn,'%s\n',text23);
  for l=1:m
   fprintf(fn,'%12.8f   %12.8f   %12.8f\n',R(l,1),TH(l,1),Z(l,1));
  end
```

```
e=' A matrix';
fprintf(fn,'%s \n',e);
[w,r]=size(A);
for l2=1:w
  fprintf(fn,'%12.8f %12.8f %12.8f %12.8f %12.8f
%12.8f\n',A(l2,1),A(l2,2),A(l2,3),A(l2,4),A(l2,5),A(l2,6));
 end

c='Inverse A matrix';
fprintf(fn,'%s \n',c);
[q,r]=size(INVE);
for l1=1:q
  fprintf(fn,'%12.8f %12.8f %12.8f %12.8f %12.8f
%12.8f\n',INVE(l1,1),INVE(l1,2),INVE(l1,3),INVE(l1,4),INVE(l1,5),INVE(l1,6));
end

p='B matrix';
fprintf(fn,'%s \n',p);
[q1,r1]=size(INVE);
for l7=1:q1
  fprintf(fn,'%12.8f\n',B(l7,1));
end

t='C matrix';
fprintf(fn,'%s \n',t);
[q2,r2]=size(INVE);
for l8=1:q2
  fprintf(fn,'%12.8f\n',C(l8,1));
end

writap(fn,R0,Ram,X0,l0,Y0,m0,Data2,xyz,Zver,Err,maxError,minError,slice,Newcoord,case1,Tdti)

fclose(fn);

%Preparation for linear programming analysis
lindo(Newcoord)

%plotting operation
plotap(Data,xyz,Zver,slice)
```

This is subfunction “plotap”.

```
function plotap(Data,xyz,Zver,slice)
%plotting operation
%plotting x and y value

plot(Data(:,1),Data(:,2),'r+')
hold on
plot(xyz(:,1),xyz(:,2),'g*')
```

```
title('Check whether the 2 dimensional representation is right or wrong')
xlabel('X')
ylabel('Y')
gtext('* are optimal cone data and + is original coordinate data')
grid

%plotting x,y and z value
figure
plot3(Data(:,1),Data(:,2),Data(:,3),'r+')
hold on
plot3(xyz(:,1),xyz(:,2),xyz(:,3),'g*')
hold on
plot3(Zver(:,1),Zver(:,2),Zver(:,3),'c-')
title('3-dimensional representation for measured and ideal cone data')
text(-0.4,-0.4,min(xyz(:,3))-0.35,'line is center axis')
xlabel('X')
ylabel('Y')
zlabel('Z')
axis('auto')
grid

%line representation for ideal cone
figure
plot3(Data(:,1),Data(:,2),Data(:,3),'r+')
hold on
plot3(xyz(:,1),xyz(:,2),xyz(:,3),'g-')
title(' line representation for ideal cone')
text(-0.6,0,min(xyz(:,3))-0.45,'green line is surface of tilted cone  ')
xlabel('X')
ylabel('Y')
zlabel('Z')
axis('auto')
grid


%'Front view of line representation for ideal cone';
figure
plot(Data(:,1),Data(:,3),'r+')
hold on
plot(xyz(:,1),xyz(:,3),'g*')
hold on
plot(Zver(:,1),Zver(:,3),'c-')
title('Front  view of line representation for ideal cone')
xlabel('X')
ylabel('Z')
gtext('red +:original data, green *:optimal cone data,cyan line:center axis')
axis('auto')
grid

%show the contour in the xy plane

figure
plot(slice(:,1),slice(:,3),'r+')
```

```
hold on
plot(Zver(:,1),Zver(:,3),'c-')
title('The contour of cone in xyplane')
xlabel('X')
ylabel('Z')
gtext('red + tilted cone, cyan line: center axis')
zoom
grid
```

This is a subfunction tcone.m.

```
function [A,B,C,R0,Ram,X0,l0,Y0,m0,INVE]=tcone(R,Z,theta,m)
%let's assume that the cone is tilted
%find matrixes and the coefficients of the tilted cone
%'A' matrix calculation
cosine=cos(theta);
sine=sin(theta);
t=Z .^2;
co=cos(theta) .^2;
si=sin(theta) .^2;
A(1,1)=sum(Z);
A(1,2)=m;
A(1,3)=sum(cos(theta));
A(1,4)=sum(sin(theta));
A(1,5)=sum(Z .* cos(theta));
A(1,6)=sum(Z .* sin(theta));
A(2,1)=sum(t);
A(2,2)=sum(Z);
A(2,3)=sum(Z .* cos(theta));
A(2,4)=sum(Z .* sin(theta));
A(2,5)=sum(t .* cos(theta));
A(2,6)=sum(t .* sin(theta));
A(3,1)=sum(Z .* cos(theta));
A(3,2)=sum(cos(theta));
A(3,3)=sum(co);
A(3,4)=sum(cos(theta) .* sin(theta));
A(3,5)=sum(Z .* co);
A(3,6)=sum(Z .* cos(theta) .* sin(theta));
A(4,1)=sum(t .* cos(theta));
A(4,2)=sum(Z .* cos(theta));
A(4,3)=sum(Z .* co);
A(4,4)=sum(Z .* cos(theta) .* sin(theta));
A(4,5)=sum(t .* co);
A(4,6)=sum(t .* cos(theta) .* sin(theta));
A(5,1)=sum(Z.* sin(theta));
A(5,2)=sum(sin(theta));
A(5,3)=sum(cos(theta) .* sin(theta));
A(5,4)=sum(si);
A(5,5)=sum(Z .* cos(theta) .* sin(theta));
A(5,6)=sum(si .* Z);
A(6,1)=sum(t .* sin(theta));
```

```
A(6,2)=sum(Z .* sin(theta));
A(6,3)=sum(Z .* cos(theta) .* sin(theta));
A(6,4)=sum(si .* Z);
A(6,5)=sum(t .* cos(theta) .* sin(theta));
A(6,6)=sum(t .* si);

%'C' matrix calculation
C(1,1)=sum(R);
C(2,1)=sum(R .* Z);
C(3,1)=sum(R .* cos(theta));
C(4,1)=sum(R .* Z .* cos(theta));
C(5,1)=sum(R .* sin(theta));
C(6,1)=sum(R .* Z .* sin(theta));

%find inverse matrix and variables in equation
INVE=inv(A);
B=INVE*C;
Ram=B(1,1);
R0=B(2,1);
X0=B(3,1);
Y0=B(4,1);
l0=B(5,1);
m0=B(6,1);
```

This is a subfunction "optimal.m".

```
function [Data2,Zver,xyz]=optimal(Data,R0,Ram,X0,l0,Y0,m0)
%let's find the optimal cone and the centers in each sections of cone
%input Data , the coefficient of equation for cone(R0,Ram,X0,l0,Y0,m0)
row=1;
hms=1;
Data2=zeros(360,3);
Zval=min(Data(:,3));
Zmax=max(Data(:,3));
Zmin=min(Data(:,3));
Gap=abs(Zmax-Zmin);
Zinc=Gap/10;
%while row<=360
while Zval<=Zmax & Zval>=Zmin
 for inc=0:10:350
  theinc=inc*pi/180;
  Radius=R0+Ram*Zval+X0*cos(theinc)+l0*Zval*cos(theinc)+Y0*sin(theinc)+m0*Zval*sin(theinc);
  Data2(row,1)=theinc;
  Data2(row,2)=Radius;
  Data2(row,3)=Zval;
  row=row+1;
 end
 Zval=Zval+Zinc;
 hms=hms+1;%how many sections are in the cone
end
hms=hms-1;
```

```
Zver=zeros(hms,3);
Zval=min(Data(:,3));
for ver=1:hms
  Ycen=Y0+m0*Zval;
  Xcen=X0+l0*Zval;
  Zver(ver,1)=Xcen;
  Zver(ver,2)=Ycen;
  Zver(ver,3)=Zval;
  Zval=Zval+Zinc;
end
%converting cylinder coordinate to x-y-z coordinate
[v,b]=size(Data2);
xyz=zeros(v,b);
[nx,ny,nz]=pol2cart(Data2(:,1),Data2(:,2),Data2(:,3));
xyz(:,1)=nx;
xyz(:,2)=ny;
xyz(:,3)=nz;
```

This is a subfuction "lindo.m".

```
function lindo(Newcoord)
%Newcoord: r theta z coords of original data Preparation for linear programming analysis
%writing progam file for lindo Newcoord()
filelin=input('What is the Input file name for lindo that you want?','s')
flin=fopen(filelin,'w');
lin='min';
lin1='h';
lin5='subject to';
lin
fprintf(flin,'%s %s\n',lin,lin1)
fprintf(flin,'%s\n',lin5)
[lin3,lin4]=size(Newcoord)
for linor=1:lin3
fprintf(flin,'R01 -R02+%12.7f Ram1-%12.7f Ram2 +%12.7f X01-%12.7f X02+%12.7f l01-%12.7f
l02\n',Newcoord(linor,3),Newcoord(linor,3),cos(Newcoord(linor,2)),cos(Newcoord(linor,2)),Newcoor
d(linor,3)*cos(Newcoord(linor,2)),Newcoord(linor,3)*cos(Newcoord(linor,2)));
fprintf(flin,' + %12.7f  Y01-%12.7f  Y02 +%12.7f  m01-%12.7f
m02+h>=%12.7f\n',sin(Newcoord(linor,2)),sin(Newcoord(linor,2)),Newcoord(linor,3)*sin(Newcoord
(linor,2)),Newcoord(linor,3)*sin(Newcoord(linor,2)),Newcoord(linor,1));
fprintf(flin,'R01 -R02+%12.7f Ram1-%12.7f Ram2 +%12.7f X01-%12.7f X02+%12.7f l01-%12.7f
l02\n',Newcoord(linor,3),Newcoord(linor,3),cos(Newcoord(linor,2)),cos(Newcoord(linor,2)),Newcoor
d(linor,3)*cos(Newcoord(linor,2)),Newcoord(linor,3)*cos(Newcoord(linor,2)));
fprintf(flin,' + %12.7f  Y01-%12.7f  Y02 +%12.7f  m01-%12.7f  m02-
h<=%12.7f\n',sin(Newcoord(linor,2)),sin(Newcoord(linor,2)),Newcoord(linor,3)*sin(Newcoord(linor,
2)),Newcoord(linor,3)*sin(Newcoord(linor,2)),Newcoord(linor,1));

end
fclose(flin);
```

This main function is used to find conicity, Td, angles for a tileted cone

```
%This function calculate R value of cone object which is depended on
%the ouput value of variables from nonlinear programming(Gino or SAS)
%To use  this function,open this m file and change the value of
%R0,Ram,X0,l0,m0,Y0 at first and then this function will calculate
%conicity and Td for you Gnewton.m
format long e;
%Input section
[m,n]=size(Data);
Newcoord=zeros(m,n);
A=zeros(6,6);
C=zeros(6,1);

t0=clock;

%data transformation loop from x,y,z coordinate to cylinder coordinate
[TH,R,Z]=CART2POL(Data(:,1),Data(:,2),Data(:,3))
Newcoord(:,1)=R;
Newcoord(:,2)=TH;
Newcoord(:,3)=Z;
theta=TH;
[A,B,C,R0,Ram,X0,l0,Y0,m0,INVE]=tcone(R,Z,theta,m);
file=input('What is the output file name that you want?','s')
fn=fopen(file,'w');
%finding optimal points of tilted cone
[Data2,Zver,xyz,height,last,Xcenm,Ycenm,Xcenv,Ycnv]=opt12(fn,Data,R0,Ram,X0,l0,Y0,m0);

%find error
[a1,a2]=size(Z)
Err=zeros(a1,1);
for c1=1:a1

Rid(c1,1)=R0+Ram*Newcoord(c1,3)+ X0*cos(Newcoord(c1,2)) + l0*Newcoord(c1,3) *
cos(Newcoord(c1,2))+Y0 * sin(Newcoord(c1,2))+m0*Newcoord(c1,3) * sin(Newcoord(c1,2));
Err(c1,1)=Newcoord(c1,1)-Rid(c1,1);
end
%find maximum and minimum
maxError=max(Err)
minError=min(Err)

%slice the cone in accordance with x-y plane

%conicity analysis(tilted cone)
%find theta between vectors
ti='This calculation is for conicity of tilted cone'
Vn=zeros(1,3);
Vn(1,1)=xyz(2,1)-xyz(1,1);
Vn(1,2)=xyz(2,2)-xyz(1,2);
Vn(1,3)=xyz(2,3)-xyz(1,3);
Vc1=zeros(1,3);
Vc1(1,1)=Zver(2,1)-Zver(1,1);
```

```
Vc1(1,2)=Zver(2,2)-Zver(1,2);
Vc1(1,3)=Zver(2,3)-Zver(1,3);

Vc2=zeros(1,3);
Vc2(1,1)=Zver(1,1)-Zver(2,1);
Vc2(1,2)=Zver(1,2)-Zver(2,2);
Vc2(1,3)=Zver(1,3)-Zver(2,3);
Vx=zeros(1,3);
Vx(1,1)=xyz(2,1)-Zver(2,1);
Vx(1,2)=xyz(2,2)-Zver(2,2);
Vx(1,3)=xyz(2,3)-Zver(2,3);
absvn=sqrt(Vn(1,1)^2+Vn(1,2)^2+Vn(1,3)^2);
absvc1=sqrt(Vc1(1,1)^2+Vc1(1,2)^2+Vc1(1,3)^2);
absvc2=sqrt(Vc2(1,1)^2+Vc2(1,2)^2+Vc2(1,3)^2);
absvx=sqrt(Vx(1,1)^2+Vx(1,2)^2+Vx(1,3)^2);
Vnn=[Vn(1,1),Vn(1,2),Vn(1,3)];
Vnc1=[Vc1(1,1),Vc1(1,2),Vc1(1,3)];
Vnc2=[Vc2(1,1),Vc2(1,2),Vc2(1,3)];
Vnx=[Vx(1,1),Vx(1,2),Vx(1,3)];
over=dot(Vnx,Vnc2);
over1=dot(Vnn,Vnc1);
thetadash=acos(over/(absvc2*absvx));
thetadash=thetadash-0.5*pi;
thetaone=acos(over1/(absvn*absvc1));
thetato=thetaone+thetadash;
angleti=(thetaone*180*2)/pi
%calculate Td(distance of maximum cone and minimum cone)
Td=abs(maxError)+abs(minError)
%calculate conicity
conicity=(Td)*cos(thetato)
%find angle max and angle min

%The end of conicity analysis for tilted cone
angxyz(1,1)=xyz(2,1)+maxError;
angxyz(1,2)=xyz(2,2);
angxyz(1,3)=xyz(2,3);
angxyz(2,1)=xyz(2,1)+minError;
angxyz(2,2)=xyz(2,2);
angxyz(2,3)=xyz(2,3);
Vna=zeros(2,3);
Vna(1,1)=angxyz(1,1)-xyz(1,1);
Vna(1,2)=angxyz(1,2)-xyz(1,2);
Vna(1,3)=angxyz(1,3)-xyz(1,3);
Vna(2,1)=angxyz(2,1)-xyz(1,1);
Vna(2,2)=angxyz(2,2)-xyz(1,2);
Vna(2,3)=angxyz(2,3)-xyz(1,3);
Vnnax=[Vna(1,1),Vna(1,2),Vna(1,3)];
Vnnan=[Vna(2,1),Vna(2,2),Vna(2,3)];
%find maximum angle
absvnx=sqrt(Vna(1,1)^2+Vna(1,2)^2+Vna(1,3)^2);
over2=dot(Vnnax,Vnc1);
thetaonex=acos(over2/(absvnx*absvc1));
maximumang=(thetaonex*2*180)/pi
```

```
%find minimum angle
absvnn=sqrt(Vna(2,1)^2+Vna(2,2)^2+Vna(2,3)^2);
over3=dot(Vnnan,Vnc1);
thetaonen=acos(over3/(absvnn*absvc1));
minimumang=(thetaonen*2*180)/pi
gap1=maximumang-angleti
gap2=angleti-minimumang

%coaxility analysis
diam=input('what is the diameter of tolerance');%diam is short for coaxility
 decision=Zver(:,1) .^ 2+Zver(:,2) .^ 2
[de,de1]=size(decision)
for che=1:de
if decision(che,de1)>(diam/2)^2
  s='The center is out of tolerance'
end
end

over7=dot(Vnnax,Vnnan);
thet=acos(over7/(absvnx*absvnn));
gapang=(thet*2*180)/pi;

time=etime(clock,t0)

%equation saving
%data output

pergap=(gap2-gap1)/gap2
pererr=(abs(minError)-maxError)/abs(minError)
```

This is a subfunction "opt12.m"

```
function
[Data2,Zver,xyz,height,last,Xcenm,Ycenm,Xcenv,Ycenv]=opt10(fn,Data,R0,S,X0,l0,Y0,m0)
%let's find the optimal cone and the centers in each sections of cone
%the equation is a nonlinear form.
%input Data , the coefficient of equation for cone(A,X0,l0,Y0,m0)
format long e;
row=1;
hms=1;
last=-R0/S%((A*Z0+Y0*m0)+sqrt((A*Z0+Y0*m0)^2-(A-m0^2)*(A*Z0^2-Y0^2)))/(A-m0^2)%Z0
height=abs(last)+0.9123;
Data2=zeros(2,3);
Zval=last;  %min(Data(:,3))
Zmax=max(Data(:,3));
Zmin=last;  %min(Data(:,3))
Gap=abs(Zmax-Zmin);
Zinc=Gap/10;
fprintf(fn,'R0=%12.8f\n',R0);
fprintf(fn,'S=%12.8f\n',S);
fprintf(fn,'X0=%12.8f\n',X0);
```

```
fprintf(fn,'Y0=%12.8f\n',Y0);
fprintf(fn,'l0=%12.8f\n',l0);
fprintf(fn,'m0=%12.8f\n',m0);
er='the inside value of root is negative';

Zvalv=last
Zvalm=Zmax

Zver=zeros(2,3);

 Ycenv=Y0+m0*Zvalv;
 Xcenv=X0+l0*Zvalv;
  Zver(1,1)=Xcenv;
  Zver(1,2)=Ycenv;
  Zver(1,3)=Zvalv;

 Ycenm=Y0+m0*Zvalm;
 Xcenm=X0+l0*Zvalm;
  Zver(2,1)=Xcenm;
  Zver(2,2)=Ycenm;
 Zver(2,3)=Zvalm;

the=0
 Radius=R0+S*Zvalv+ X0*cos(the) + l0*Zvalv * cos(the)+Y0 * sin(the)+m0*Zvalv * sin(the);
 Data2(row,1)=the;
  Data2(row,2)=Radius;
  Data2(row,3)=Zvalv;
%end
%prospect the maximum value
max=zeros(1,3)
max(1,3)=Zvalm
max(1,2)=Ycenm
max(1,1)=Xcenm+R0+S*Zvalm
[the1,r,Zvalm]=cart2pol(max(1,1),max(1,2),max(1,3))
Radius1=R0+S*Zvalm+ X0*cos(the1) + l0*Zvalm * cos(the1)+Y0 * sin(the1)+m0*Zvalm *
sin(the1);
[comx,comy]=pol2cart(the1,Radius1)
comxol=comx
delta=1
while delta >0,

 the1=the1+0.00000001
 Radius1=R0+S*Zvalm+ X0*cos(the1) + l0*Zvalm * cos(the1)+Y0 * sin(the1)+m0*Zvalm *
sin(the1);
 [comxnew,comynew]=pol2cart(the1,Radius1);
 delta=comxnew-comxol;
 ccomxol=comxnew;
end
 the1=the1-0.00000001
Radius1=R0+S*Zvalm+ X0*cos(the1) + l0*Zvalm * cos(the1)+Y0 * sin(the1)+m0*Zvalm *
sin(the1);

row=2
```

```
Data2(row,1)=the1
Data2(row,2)=Radius1
Data2(row,3)=Zvalm

%converting cylinder coordinate to x-y-z coordinate
[v,b]=size(Data2);
xyz=zeros(v,b);
[nx,ny,nz]=pol2cart(Data2(:,1),Data2(:,2),Data2(:,3))
xyz(:,1)=nx
xyz(:,2)=ny
xyz(:,3)=nz
xyz(1,2)=Ycenv
```

LPCONE1.M

```
%This function calculate R value of cone object which is depended on
%the ouput value of variables from linear programming
%To use  this function,open this m file and change the value of
%R0,Ram,X0,l0,m0,Y0 at first and then this function will calculate
%conicity and Td for you

%Input section
[m,n]=size(Data);
Newcoord=zeros(m,n);
A=zeros(6,6);
C=zeros(6,1);

t0=clock;

%data transformation loop from x,y,z coordinate to cylinder coordinate
[TH,R,Z]=CART2POL(Data(:,1),Data(:,2),Data(:,3))
Newcoord(:,1)=R;
Newcoord(:,2)=TH;
Newcoord(:,3)=Z;
theta=TH;

R0=0.193443
Ram=0.177134
X0=0.000746
l0=-0.002373
Y0=-0.000839
m0=0.00492

%finding optimal points of tilted cone
[Data2,Zver,xyz]=optimal(Data,R0,Ram,X0,l0,Y0,m0);

%find error
[a1,a2]=size(Z)
Err=zeros(a1,1);
for c1=1:a1
Err(c1,1)=Newcoord(c1,1)-R0-Ram*Newcoord(c1,3)- X0*cos(Newcoord(c1,2)) - l0*Newcoord(c1,3)
* cos(Newcoord(c1,2))-Y0 * sin(Newcoord(c1,2))-m0*Newcoord(c1,3) * sin(Newcoord(c1,2));
```

```
end
%find maximum and minimum
maxError=max(Err)
minError=min(Err)

%slice the cone in accordance with x-y plane
[l1,j1]=size(xyz);
slice=zeros((l1/36),3);

nc=1;
for na=1:36:396
 slice(nc,1)=xyz(na,1);
 slice(nc,2)=xyz(na,2);
 slice(nc,3)=xyz(na,3);
 nc=nc+1;
end

%conicity analysis(tilted cone)
%find theta between vectors
ti='This calculation is for conicity of tilted cone'
Vn=zeros(1,3);
Vn(1,1)=xyz(361,1)-xyz(1,1);
Vn(1,2)=xyz(361,2)-xyz(1,2);
Vn(1,3)=xyz(361,3)-xyz(1,3);
Vc1=zeros(1,3);
Vc1(1,1)=Zver(11,1)-Zver(1,1);
Vc1(1,2)=Zver(11,2)-Zver(1,2);
Vc1(1,3)=Zver(11,3)-Zver(1,3);
Vc2=zeros(1,3);
Vc2(1,1)=Zver(1,1)-Zver(11,1);
Vc2(1,2)=Zver(1,2)-Zver(11,2);
Vc2(1,3)=Zver(1,3)-Zver(11,3);
Vx=zeros(1,3);
Vx(1,1)=xyz(361,1)-Zver(11,1);
Vx(1,2)=xyz(361,2)-Zver(11,2);
Vx(1,3)=xyz(361,3)-Zver(11,3);
absvn=sqrt(Vn(1,1)^2+Vn(1,2)^2+Vn(1,3)^2);
absvc1=sqrt(Vc1(1,1)^2+Vc1(1,2)^2+Vc1(1,3)^2);
absvc2=sqrt(Vc2(1,1)^2+Vc2(1,2)^2+Vc2(1,3)^2);
absvx=sqrt(Vx(1,1)^2+Vx(1,2)^2+Vx(1,3)^2);
Vnn=[Vn(1,1),Vn(1,2),Vn(1,3)];
Vnc1=[Vc1(1,1),Vc1(1,2),Vc1(1,3)];
Vnc2=[Vc2(1,1),Vc2(1,2),Vc2(1,3)];
Vnx=[Vx(1,1),Vx(1,2),Vx(1,3)];
over=dot(Vnx,Vnc2);
over1=dot(Vnn,Vnc1);
thetadash=acos(over/(absvc2*absvx));
thetadash=thetadash-0.5*pi;
thetaone=acos(over1/(absvn*absvc1));
thetato=thetaone+thetadash;
angleti=(thetaone*180*2)/pi
%calculate Td(distance of maximum cone and minimum cone)
Td=abs(maxError)+abs(minError)
```

```
%calculate conicity
conicity=(Td)*cos(thetato)
%find angle max and angle min

%The end of conicity analysis for tilted cone
angerror=zeros(2,3);
angerror(1,2)=Data2(361,2)+maxError;
angerror(1,1)=Data2(361,1);
angerror(1,3)=Data2(361,3);
angerror(2,2)=Data2(361,2)+minError;
angerror(2,1)=Data2(361,1);
angerror(2,3)=Data2(361,3);
angxyz=zeros(2,3);
[a1,a2,a3]=pol2cart(angerror(1,1),angerror(1,2),angerror(1,3));
angxyz(1,1)=a1;
angxyz(1,2)=a2;
angxyz(1,3)=a3;
[a4,a5,a6]=pol2cart(angerror(2,1),angerror(2,2),angerror(2,3));
angxyz(2,1)=a4;
angxyz(2,2)=a5;
angxyz(2,3)=a6;
Vna=zeros(2,3);
Vna(1,1)=angxyz(1,1)-xyz(1,1);
Vna(1,2)=angxyz(1,2)-xyz(1,2);
Vna(1,3)=angxyz(1,3)-xyz(1,3);
Vna(2,1)=angxyz(2,1)-xyz(1,1);
Vna(2,2)=angxyz(2,2)-xyz(1,2);
Vna(2,3)=angxyz(2,3)-xyz(1,3);
Vnnax=[Vna(1,1),Vna(1,2),Vna(1,3)];
Vnnan=[Vna(2,1),Vna(2,2),Vna(2,3)];
%find maximum angle
absvnx=sqrt(Vna(1,1)^2+Vna(1,2)^2+Vna(1,3)^2);
over2=dot(Vnnax,Vnc1);
thetaonex=acos(over2/(absvnx*absvc1));
maximumang=(thetaonex*2*180)/pi
%find minimum angle
absvnn=sqrt(Vna(2,1)^2+Vna(2,2)^2+Vna(2,3)^2);
over3=dot(Vnnan,Vnc1);
thetaonen=acos(over3/(absvnn*absvc1));
minimumang=(thetaonen*2*180)/pi

%coaxility analysis
diam=input('what is the diameter of tolerance');%diam is short for coaxility
 decision=Zver(:,1) .^ 2+Zver(:,2) .^ 2
[de,de1]=size(decision)
for che=1:de
if decision(che,de1)>(diam/2)^2
  s='The center is out of tolerance'
end
end

time=etime(clock,t0)
```

```
%equation saving
%data output
file=input('What is the output file name that you want?','s')
fn=fopen(file,'w');
case1='tilted cone with using linear programming technique';
case2='right cone';

text4='original data';
fprintf(fn,'%s\n',text4);
[m,n]=size(Data);
 for l=1:m
  fprintf(fn,'%12.8f    %12.8f   %12.8f\n',Data(l,1),Data(l,2),Data(l,3));
 end

text23='the converted r theta z of original data';
fprintf(fn,'%s\n',text23);
 for l=1:m
  fprintf(fn,'%12.8f    %12.8f   %12.8f\n',R(l,1),TH(l,1),Z(l,1));
 end

writing(fn,R0,Ram,X0,l0,Y0,m0,Data2,xyz,Zver,Err,maxError,minError,slice,Newcoord,case1,Td,co
nicity,angleti,maximumang,minimumang)

fclose(fn);

%plotting operation
plot1(Data,xyz,Zver,slice)
```

THIS IS A SUBFUNCTION "PLOT1"

```
function plot1(Data,xyz,Zver,slice)
%plotting operation
%plotting x and y value

plot(Data(:,1),Data(:,2),'r+')%original data
hold on
plot(xyz(:,1),xyz(:,2),'g*')%optimal cone data
title('Check whether the 2 dimensional representation is right or wrong')
gtext('* are optimal cone data and + is original coordinate data')
xlabel('X')
ylabel('Y')
grid

%plotting x,y and z value
figure
plot3(Data(:,1),Data(:,2),Data(:,3),'r+')
hold on
plot3(xyz(:,1),xyz(:,2),xyz(:,3),'g*')
hold on
plot3(Zver(:,1),Zver(:,2),Zver(:,3),'c-')%center
```

```
title('3-dimensional representation for measured and ideal cone data')
text(-0.4,-0.4,min(xyz(:,3))-0.35,'line descrbe the center axis')
xlabel('X')
ylabel('Y')
zlabel('Z')
axis('auto')
grid

%line representation for ideal cone
figure
plot3(Data(:,1),Data(:,2),Data(:,3),'r+')
hold on
plot3(xyz(:,1),xyz(:,2),xyz(:,3),'g-')
title(' line representation for ideal cone')
text(-0.4,-0.4,min(xyz(:,3))-0.35,'+ are original data')
xlabel('X')
ylabel('Y')
zlabel('Z')
axis('auto')
grid

%'Front top view of line representation for ideal cone';
figure
plot3(Data(:,1),Data(:,2),Data(:,3),'r+')
hold on
plot3(xyz(:,1),xyz(:,2),xyz(:,3),'g*')
hold on
plot3(Zver(:,1),Zver(:,2),Zver(:,3),'c-')
title('Front  view of line representation for ideal cone')
xlabel('X')
ylabel('Y')
zlabel('Z')
axis('auto')
view([0,-1,0])
grid

%show the contour in the xy plane
figure
plot(slice(:,1),slice(:,3),'r*')
hold on
plot(Zver(:,1),Zver(:,3),'c-')
title('The representation of contour of cone')
gtext('* are optimal cone data and line is cone axis')
xlabel('X')
ylabel('z')
zoom
grid
```

This is a subfunction "writing.m"

```
function
writing(fn,R0,Ram,X0,l0,Y0,m0,Data2,xyz,Zver,Err,maxError,minError,slice,Newcoord,case,Td,con
icity,angleti,maximumang,minimumang)
%equation saving and data output
%writing(Data,R,TH,Z,R0,Ram,X0,l0,Y0,m0,Data2,xyz,Zver,Err,slice,Newcoord)
%fn:file name Data: original data R,TH,Z:converted coordinate(Newcoord)
%R0,Ram,X0,l0,Y0,m0:the coefficients of equation Data2,xyz:ideal cone Zver:center
%Err: the distance of error, slice:contour of cone

fprintf(fn,'%s\n',case)

m='The equation for cone is';
fprintf(fn,'%s \n',m);
fprintf(fn,'%10.5f+%10.5f * Zi+ %10.5f * cos theta+%10.5f * Zi*cos theta\n',R0,Ram,X0,l0);
fprintf(fn,'+%10.5f * sin (theta)+ %10.5f * Zi* sin (theta)\n',Y0,m0);

ex='Ideal data generation(theta R Z)';
fprintf(fn,'%s\n',ex);
[g,h]=size(Data2);
 for l12=1:g
  fprintf(fn,'%d %12.8f    %12.8f   %12.8f\n',l12,Data2(l12,1),Data2(l12,2),Data2(l12,3));
 end

text11='Ideal data conversion to xyz coordinate';
fprintf(fn,'%s\n',text11);
 for l12=1:g
  fprintf(fn,'%d %12.8f    %12.8f   %12.8f\n',l12,xyz(l12,1),xyz(l12,2),xyz(l12,3));
 end

y='Center of ideal cone sections only for tilted cone';
fprintf(fn,'%s \n',y);
[b,r]=size(Zver);
for l31=1:b
  fprintf(fn,'%12.8f %12.8f\n',Zver(l31,1),Zver(l31,2));
end

y2='The error';
fprintf(fn,'%s \n',y2);
[b1,r1]=size(Err);
for l32=1:b1
  fprintf(fn,'%12.8f\n',Err(l32,1));
end

y32='The residual sums of square(sum of error squres)';
fprintf(fn,'%s \n',y32);
sumsq=sum(Err(:,1) .^2)
  fprintf(fn,'%15.13f\n',sumsq);%sumsq is the summation of error squares

y2='The maximum and minimum error';
  fprintf(fn,'%s \n',y2);
  fprintf(fn,'%12.8f\n',maxError);
  fprintf(fn,'%12.8f\n',minError);
y2='the conicity and Td';
```

```
y21='conicity';
y22='Td';
y23='angle of reference cone';
y24='maximum angle';
y25='minimum angle';
  fprintf(fn,'%s \n',y2);
  fprintf(fn,'%s:%12.8f\n',y22,Td);
  fprintf(fn,'%s:%12.8f\n',y21,conicity);
  fprintf(fn,'%s:%12.8f\n',y23,angleti);
if maximumang~=0 | minimumang ~=0
  fprintf(fn,'%s:%12.8f\n',y24,maximumang);
  fprintf(fn,'%s:%12.8f\n',y25,minimumang);
end

yr3='The contour coords of  cone which is ';
fprintf(fn,'%s %s\n',yr3,case);
[br2,r1]=size(slice);
for l33=1:br2
  fprintf(fn,'%12.8f %12.8f %12.8f\n',slice(l33,1),slice(l33,2),slice(l33,3));
end
```

Coneli11.m

```
%This function calculate R value of cone object which is depended on
%the ouput value of variables from nonlinear programming(Gino or SAS)
%To use  this function,open this m file and change the value of
%R0,Ram,X0,l0,m0,Y0 at first and then this function will calculate
%conicity and Td for you Gnewton.m
format long;
%Input section
[m,n]=size(Data);
Newcoord=zeros(m,n);
A=zeros(6,6);
C=zeros(6,1);

t0=clock;

%data transformation loop from x,y,z coordinate to cylinder coordinate
[TH,R,Z]=CART2POL(Data(:,1),Data(:,2),Data(:,3))
Newcoord(:,1)=R;
Newcoord(:,2)=TH;
Newcoord(:,3)=Z;
theta=TH;
R0=0.193443
Ram=0.177134
X0=0.000746
l0=-0.002373
Y0=-0.000839
m0=0.00492

%[A,B,C,R0,Ram,X0,l0,Y0,m0,INVE]=tcone(R,Z,theta,m);
file=input('What is the output file name that you want?','s')
```

```
fn=fopen(file,'w');
%finding optimal points of tilted cone
[Data2,Zver,xyz,height,last,Xcenm,Ycenm,Xcenv,Ycenv]=opt12(fn,Data,R0,Ram,X0,l0,Y0,m0);

%find error
[a1,a2]=size(Z)
Err=zeros(a1,1);
for c1=1:a1

Err(c1,1)=Newcoord(c1,1)-R0-Ram*Newcoord(c1,3)- X0*cos(Newcoord(c1,2)) - l0*Newcoord(c1,3)
* cos(Newcoord(c1,2))-Y0 * sin(Newcoord(c1,2))-m0*Newcoord(c1,3) * sin(Newcoord(c1,2));
end
%find maximum and minimum
maxError=max(Err)
minError=min(Err)

%slice the cone in accordance with x-y plane

%conicity analysis(tilted cone)
%find theta between vectors
ti='This calculation is for conicity of tilted cone'
Vn=zeros(1,3);
Vn(1,1)=xyz(2,1)-xyz(1,1);
Vn(1,2)=xyz(2,2)-xyz(1,2);
Vn(1,3)=xyz(2,3)-xyz(1,3);
Vc1=zeros(1,3);
Vc1(1,1)=Zver(2,1)-Zver(1,1);
Vc1(1,2)=Zver(2,2)-Zver(1,2);
Vc1(1,3)=Zver(2,3)-Zver(1,3);

Vc2=zeros(1,3);
Vc2(1,1)=Zver(1,1)-Zver(2,1);
Vc2(1,2)=Zver(1,2)-Zver(2,2);
Vc2(1,3)=Zver(1,3)-Zver(2,3);
Vx=zeros(1,3);
Vx(1,1)=xyz(2,1)-Zver(2,1);
Vx(1,2)=xyz(2,2)-Zver(2,2);
Vx(1,3)=xyz(2,3)-Zver(2,3);
absvn=sqrt(Vn(1,1)^2+Vn(1,2)^2+Vn(1,3)^2);
absvc1=sqrt(Vc1(1,1)^2+Vc1(1,2)^2+Vc1(1,3)^2);
absvc2=sqrt(Vc2(1,1)^2+Vc2(1,2)^2+Vc2(1,3)^2);
absvx=sqrt(Vx(1,1)^2+Vx(1,2)^2+Vx(1,3)^2);
Vnn=[Vn(1,1),Vn(1,2),Vn(1,3)];
Vnc1=[Vc1(1,1),Vc1(1,2),Vc1(1,3)];
Vnc2=[Vc2(1,1),Vc2(1,2),Vc2(1,3)];
Vnx=[Vx(1,1),Vx(1,2),Vx(1,3)];
over=dot(Vnx,Vnc2);
over1=dot(Vnn,Vnc1);
thetadash=acos(over/(absvc2*absvx));
thetadash=thetadash-0.5*pi;
thetaone=acos(over1/(absvn*absvc1));
thetato=thetaone+thetadash;
angleti=(thetaone*180*2)/pi
```

```
%calculate Td(distance of maximum cone and minimum cone)
Td=abs(maxError)+abs(minError)
%calculate conicity
conicity=(Td)*cos(thetato)
%find angle max and angle min

%The end of conicity analysis for tilted cone
angxyz(1,1)=xyz(2,1)+maxError;
angxyz(1,2)=xyz(2,2);
angxyz(1,3)=xyz(2,3);
angxyz(2,1)=xyz(2,1)+minError;
angxyz(2,2)=xyz(2,2);
angxyz(2,3)=xyz(2,3);
Vna=zeros(2,3);
Vna(1,1)=angxyz(1,1)-xyz(1,1);
Vna(1,2)=angxyz(1,2)-xyz(1,2);
Vna(1,3)=angxyz(1,3)-xyz(1,3);
Vna(2,1)=angxyz(2,1)-xyz(1,1);
Vna(2,2)=angxyz(2,2)-xyz(1,2);
Vna(2,3)=angxyz(2,3)-xyz(1,3);
Vnnax=[Vna(1,1),Vna(1,2),Vna(1,3)];
Vnnan=[Vna(2,1),Vna(2,2),Vna(2,3)];
%find maximum angle
absvnx=sqrt(Vna(1,1)^2+Vna(1,2)^2+Vna(1,3)^2);
over2=dot(Vnnax,Vnc1);
thetaonex=acos(over2/(absvnx*absvc1));
maximumang=(thetaonex*2*180)/pi
%find minimum angle
absvnn=sqrt(Vna(2,1)^2+Vna(2,2)^2+Vna(2,3)^2);
over3=dot(Vnnan,Vnc1);
thetaonen=acos(over3/(absvnn*absvc1));
minimumang=(thetaonen*2*180)/pi
gap1=maximumang-angleti
gap2=angleti-minimumang
%maximumang=20.15067565+gap1
%minimumang=20.15067565-gap2

%coaxility analysis
diam=input('what is the diameter of tolerance');%diam is short for coaxility
 decision=Zver(:,1) .^ 2+Zver(:,2) .^ 2
[de,de1]=size(decision)
for che=1:de
if decision(che,de1)>(diam/2)^2
  s='The center is out of tolerance'
end
end

over7=dot(Vnnax,Vnnan);
thet=acos(over7/(absvnx*absvnn));
gapang=(thet*2*180)/pi;

time=etime(clock,t0)
```

```
%equation saving
%data output

pergap=(gap2-gap1)/gap2
pererr=(abs(minError)-maxError)/abs(minError)
confidential=abs(pergap-pererr)/max(abs(pergap),abs(pererr))
```

APPENDIX B-2

CONENO10

```
%This function calculate R value of cone object which is depended on
%the ouput value of variables from nonlinear programming(Gino or SAS)
%To use  this function,open this m file and change the value of
%R0,Ram,X0,l0,m0,Y0 at first and then this function will calculate
%conicity and Td for you Gnewton.m
format long;
%Input section
[m,n]=size(Data);
Newcoord=zeros(m,n);
A=zeros(6,6);
C=zeros(6,1);

t0=clock;

%data transformation loop from x,y,z coordinate to cylinder coordinate
[TH,R,Z]=CART2POL(Data(:,1),Data(:,2),Data(:,3))
Newcoord(:,1)=R;
Newcoord(:,2)=TH;
Newcoord(:,3)=Z;
theta=TH;
%[A,B,C,R0,Ram,X0,l0,Y0,m0,INVE]=tcone(R,Z,theta,m);
Z0=-1.089436924
A=0.031522003
X0=0.00103398
l0=-0.002818651
Y0=-0.000891455
m0=0.005186123

file=input('What is the output file name that you want?','s')
fn=fopen(file,'w');
%finding optimal points of tilted cone
[Data2,Zver,xyz,height,last,A01,A11,Xcenm,Ycenm,Xcenv,Ycenv]=opt15(fn,Data,A,Z0,X0,l0,Y0,m
0);
```

```
%find error
[a1,a2]=size(Z)
Err=zeros(a1,1);
for c1=1:a1

A1=(X0*cos(Newcoord(c1,2))+l0*Newcoord(c1,3)*cos(Newcoord(c1,2))+Y0*sin(Newcoord(c1,2))+
m0*Newcoord(c1,3)*sin(Newcoord(c1,2)));
 A0=A1^2-
(X0^2+2*X0*l0*Newcoord(c1,3)+l0^2*Newcoord(c1,3)^2+Y0^2+m0^2*Newcoord(c1,3)^2+2*Y0*
m0*Newcoord(c1,3)-A*Newcoord(c1,3)^2-A*Z0^2+2*A*Z0*Newcoord(c1,3));

 Err(c1,1)=Newcoord(c1,1)-A1-sqrt(A0);
 end
%find maximum and minimum
maxError=max(Err)
minError=min(Err)

%slice the cone in accordance with x-y plane

%conicity analysis(tilted cone)
%find theta between vectors
ti='This calculation is for conicity of tilted cone'
Vn=zeros(1,3);
Vn(1,1)=xyz(2,1)-xyz(1,1);
Vn(1,2)=xyz(2,2)-xyz(1,2);
Vn(1,3)=xyz(2,3)-xyz(1,3);
Vc1=zeros(1,3);
Vc1(1,1)=Zver(2,1)-Zver(1,1);
Vc1(1,2)=Zver(2,2)-Zver(1,2);
Vc1(1,3)=Zver(2,3)-Zver(1,3);

Vc2=zeros(1,3);
Vc2(1,1)=Zver(1,1)-Zver(2,1);
Vc2(1,2)=Zver(1,2)-Zver(2,2);
Vc2(1,3)=Zver(1,3)-Zver(2,3);
Vx=zeros(1,3);
Vx(1,1)=xyz(2,1)-Zver(2,1);
Vx(1,2)=xyz(2,2)-Zver(2,2);
Vx(1,3)=xyz(2,3)-Zver(2,3);
absvn=sqrt(Vn(1,1)^2+Vn(1,2)^2+Vn(1,3)^2);
absvc1=sqrt(Vc1(1,1)^2+Vc1(1,2)^2+Vc1(1,3)^2);
absvc2=sqrt(Vc2(1,1)^2+Vc2(1,2)^2+Vc2(1,3)^2);
absvx=sqrt(Vx(1,1)^2+Vx(1,2)^2+Vx(1,3)^2);
Vnn=[Vn(1,1),Vn(1,2),Vn(1,3)];
Vnc1=[Vc1(1,1),Vc1(1,2),Vc1(1,3)];
Vnc2=[Vc2(1,1),Vc2(1,2),Vc2(1,3)];
Vnx=[Vx(1,1),Vx(1,2),Vx(1,3)];
over=dot(Vnx,Vnc2);
over1=dot(Vnn,Vnc1);
thetadash=acos(over/(absvc2*absvx));
thetadash=thetadash-0.5*pi;
thetaone=acos(over1/(absvn*absvc1));
thetato=thetaone+thetadash;
```

```
angleti=(thetaone*180*2)/pi
%calculate Td(distance of maximum cone and minimum cone)
Td=abs(maxError)+abs(minError)
%calculate conicity
conicity=(Td)*cos(thetato)
%find angle max and angle min

%The end of conicity analysis for tilted cone
angxyz=zeros(2,3);
angxyz(1,1)=xyz(2,1)+maxError;
angxyz(1,2)=xyz(2,2);
angxyz(1,3)=xyz(2,3);
angxyz(2,1)=xyz(2,1)+minError;
angxyz(2,2)=xyz(2,2);
angxyz(2,3)=xyz(2,3);
Vna=zeros(2,3);
Vna(1,1)=angxyz(1,1)-xyz(1,1);
Vna(1,2)=angxyz(1,2)-xyz(1,2);
Vna(1,3)=angxyz(1,3)-xyz(1,3);
Vna(2,1)=angxyz(2,1)-xyz(1,1);
Vna(2,2)=angxyz(2,2)-xyz(1,2);
Vna(2,3)=angxyz(2,3)-xyz(1,3);
Vnnax=[Vna(1,1),Vna(1,2),Vna(1,3)];
Vnnan=[Vna(2,1),Vna(2,2),Vna(2,3)];
%find maximum angle
absvnx=sqrt(Vna(1,1)^2+Vna(1,2)^2+Vna(1,3)^2);
over2=dot(Vnnax,Vnc1);
thetaonex=acos(over2/(absvnx*absvc1));
maximumang=(thetaonex*2*180)/pi
%find minimum angle
absvnn=sqrt(Vna(2,1)^2+Vna(2,2)^2+Vna(2,3)^2);
over3=dot(Vnnan,Vnc1);
thetaonen=acos(over3/(absvnn*absvc1));
minimumang=(thetaonen*2*180)/pi
gap1=maximumang-angleti
gap2=angleti-minimumang

%coaxility analysis
diam=input('what is the diameter of tolerance');%diam is short for coaxility
 decision=Zver(:,1) .^ 2+Zver(:,2) .^ 2
[de,de1]=size(decision)
for che=1:de
if decision(che,de1)>(diam/2)^2
   s='The center is out of tolerance'
end
end

over7=dot(Vnnax,Vnnan);
thet=acos(over7/(absvnx*absvnn));
gapang=(thet*2*180)/pi;

time=etime(clock,t0)
```

```
%equation saving
%data output

pergap=abs(gap2-gap1)/gap2
pererr=(abs(minError)-maxError)/abs(minError)
confidential=abs(pergap-pererr)/max(abs(pergap),abs(pererr))
```

This is a subfunction "OPT15.M"

```
function
[Data2,Zver,xyz,height,last,A01,A11,Xcenm,Ycenm,Xcenv,Ycenv,thecenv,rcenv,A01v,A11v,part1,p
art2,error,part3]=opt15(fn,Data,A,Z0,X0,l0,Y0,m0)
%let's find the optimal cone and the centers in each sections of cone
%the equation is a nonlinear form.
%input Data , the coefficient of equation for cone(A,X0,l0,Y0,m0)
format long e;
row=1;
hms=1;
last=Z0
Data2=zeros(2,3);
Zval=last;  %min(Data(:,3))
Zmax=max(Data(:,3));
Zmin=last;  %min(Data(:,3))
Gap=abs(Zmax-Zmin);
Zinc=Gap/10;
fprintf(fn,'A=%12.8f\n',A);
fprintf(fn,'Z0=%12.8f\n',Z0);
fprintf(fn,'X0=%12.8f\n',X0);
fprintf(fn,'Y0=%12.8f\n',Y0);
fprintf(fn,'l0=%12.8f\n',l0);
fprintf(fn,'m0=%12.8f\n',m0);
er='the inside value of root is negative';

Zvalv=last
Zvalm=Zmax

Zver=zeros(2,3);
  Ycenv=Y0+m0*Zvalv;
  Xcenv=X0+l0*Zvalv;
  Zver(1,1)=Xcenv;
  Zver(1,2)=Ycenv;
  Zver(1,3)=Zvalv;

 Ycenm=Y0+m0*Zvalm;
  Xcenm=X0+l0*Zvalm;
  Zver(2,1)=Xcenm;
  Zver(2,2)=Ycenm;
```

```
Zver(2,3)=Zvalm;
row=1
[thecenv,rcenv]=cart2pol(Xcenv,Ycenv)

the=thecenv
A11v=(X0*cos(the)+l0*Zvalv*cos(the)+Y0*sin(the)+m0*Zvalv*sin(the))
part1=A11v^2
part2=(X0^2+2*X0*l0*Zvalv+l0^2*Zvalv^2+Y0^2+m0^2*Zvalv^2+2*Y0*m0*Zvalv)
part3=A*Zvalv^2+A*Z0^2-2*A*Z0*Zvalv
A01v=part1-(part2-part3)
Radius=A11v+sqrt(A01v);%since a0 is almost zero
Data2(row,1)=the;
Data2(row,2)=Radius;
Data2(row,3)=Zvalv;

row=2
max=zeros(1,3)
max(1,3)=Zvalm
max(1,2)=Ycenm
thma=0
A11=(X0*cos(thma)+l0*Zvalm*cos(thma)+Y0*sin(thma)+m0*Zvalm*sin(thma));
A01=A11^2-(X0^2+2*X0*l0*Zvalm+l0^2*Zvalm^2+Y0^2+m0^2*Zvalm^2+2*Y0*m0*Zvalm-
A*Zvalm^2-A*Z0^2+2*A*Z0*Zvalm);
max(1,1)=Xcenm+sqrt(A01)

[the1,r,Zvalm]=cart2pol(max(1,1),max(1,2),max(1,3))
A1=(X0*cos(the1)+l0*Zvalm*cos(the1)+Y0*sin(the1)+m0*Zvalm*sin(the1));
A0=A1^2-(X0^2+2*X0*l0*Zvalm+l0^2*Zvalm^2+Y0^2+m0^2*Zvalm^2+2*Y0*m0*Zvalm-
A*Zvalm^2-A*Z0^2+2*A*Z0*Zvalm);
Radius=A1+sqrt(A0);%since a0 is almost zero
[comx,comy]=pol2cart(the1,Radius)
comxol=comx
delta=1
while delta >0,

the1=the1-1E-9;
A1=(X0*cos(the1)+l0*Zvalm*cos(the1)+Y0*sin(the1)+m0*Zvalm*sin(the1));
A0=A1^2-(X0^2+2*X0*l0*Zvalm+l0^2*Zvalm^2+Y0^2+m0^2*Zvalm^2+2*Y0*m0*Zvalm-
A*Zvalm^2-A*Z0^2+2*A*Z0*Zvalm);
Radius=A1+sqrt(A0);%since a0 is almost zero

[comxnew,comynew]=pol2cart(the1,Radius);
delta=comxnew-comxol
ccomxol=comxnew;
end
the1=the1+1E-9

A1=(X0*cos(the1)+l0*Zvalm*cos(the1)+Y0*sin(the1)+m0*Zvalm*sin(the1));
A0=A1^2-(X0^2+2*X0*l0*Zvalm+l0^2*Zvalm^2+Y0^2+m0^2*Zvalm^2+2*Y0*m0*Zvalm-
A*Zvalm^2-A*Z0^2+2*A*Z0*Zvalm);
Radius=A1+sqrt(A0);%since a0 is almost zero
```

```
Data2(row,1)=the1;
 Data2(row,2)=Radius;
 Data2(row,3)=Zvalm;

%converting cylinder coordinate to x-y-z coordinate
[v,b]=size(Data2);
xyz=zeros(v,b);
[nx,ny,nz]=pol2cart(Data2(:,1),Data2(:,2),Data2(:,3));
xyz(:,1)=nx;
xyz(:,2)=ny;
xyz(:,3)=nz;
error=part1-part2
```

Coneno11

```
%This function calculate R value of cone object which is depended on
%the ouput value of variables from nonlinear programming(Gino or SAS)
%To use  this function,open this m file and change the value of
%R0,Ram,X0,l0,m0,Y0 at first and then this function will calculate
%conicity and Td for you Gnewton.m
format long;
%Input section
[m,n]=size(Data);
Newcoord=zeros(m,n);
A=zeros(6,6);
C=zeros(6,1);

t0=clock;

%data transformation loop from x,y,z coordinate to cylinder coordinate
[TH,R,Z]=CART2POL(Data(:,1),Data(:,2),Data(:,3))
Newcoord(:,1)=R;
Newcoord(:,2)=TH;
Newcoord(:,3)=Z;
theta=TH;
%[A,B,C,R0,Ram,X0,l0,Y0,m0,INVE]=tcone(R,Z,theta,m);

Z0=-1.091928
A=0.031383
X0=0.000745
l0=-0.002367
Y0=-0.000839
m0=0.00492

file=input('What is the output file name that you want?','s')
fn=fopen(file,'w');
%finding optimal points of tilted cone
[Data2,Zver,xyz,height,last,A01,A11,Xcenm,Ycenm,Xcenv,Ycenv]=opt15(fn,Data,A,Z0,X0,l0,Y0,m
0);

%find error
```

```
[a1,a2]=size(Z)
Err=zeros(a1,1);
for c1=1:a1

A1=(X0*cos(Newcoord(c1,2))+l0*Newcoord(c1,3)*cos(Newcoord(c1,2))+Y0*sin(Newcoord(c1,2))+
m0*Newcoord(c1,3)*sin(Newcoord(c1,2)));
 A0=A1^2-
(X0^2+2*X0*l0*Newcoord(c1,3)+l0^2*Newcoord(c1,3)^2+Y0^2+m0^2*Newcoord(c1,3)^2+2*Y0*
m0*Newcoord(c1,3)-A*Newcoord(c1,3)^2-A*Z0^2+2*A*Z0*Newcoord(c1,3));

 Err(c1,1)=Newcoord(c1,1)-A1-sqrt(A0);
 end
%end
%find maximum and minimum
maxError=max(Err)
minError=min(Err)

%slice the cone in accordance with x-y plane

%conicity analysis(tilted cone)
%find theta between vectors
ti='This calculation is for conicity of tilted cone'
Vn=zeros(1,3);
Vn(1,1)=xyz(2,1)-xyz(1,1);
Vn(1,2)=xyz(2,2)-xyz(1,2);
Vn(1,3)=xyz(2,3)-xyz(1,3);
Vc1=zeros(1,3);
Vc1(1,1)=Zver(2,1)-Zver(1,1);
Vc1(1,2)=Zver(2,2)-Zver(1,2);
Vc1(1,3)=Zver(2,3)-Zver(1,3);

 Vc2=zeros(1,3);
 Vc2(1,1)=Zver(1,1)-Zver(2,1);
 Vc2(1,2)=Zver(1,2)-Zver(2,2);
 Vc2(1,3)=Zver(1,3)-Zver(2,3);
 Vx=zeros(1,3);
 Vx(1,1)=xyz(2,1)-Zver(2,1);
 Vx(1,2)=xyz(2,2)-Zver(2,2);
 Vx(1,3)=xyz(2,3)-Zver(2,3);
 absvn=sqrt(Vn(1,1)^2+Vn(1,2)^2+Vn(1,3)^2);
 absvc1=sqrt(Vc1(1,1)^2+Vc1(1,2)^2+Vc1(1,3)^2);
 absvc2=sqrt(Vc2(1,1)^2+Vc2(1,2)^2+Vc2(1,3)^2);
 absvx=sqrt(Vx(1,1)^2+Vx(1,2)^2+Vx(1,3)^2);
 Vnn=[Vn(1,1),Vn(1,2),Vn(1,3)];
 Vnc1=[Vc1(1,1),Vc1(1,2),Vc1(1,3)];
 Vnc2=[Vc2(1,1),Vc2(1,2),Vc2(1,3)];
 Vnx=[Vx(1,1),Vx(1,2),Vx(1,3)];
 over=dot(Vnx,Vnc2);
 over1=dot(Vnn,Vnc1);
 thetadash=acos(over/(absvc2*absvx));
 thetadash=thetadash-0.5*pi;
 thetaone=acos(over1/(absvn*absvc1));
 thetato=thetaone+thetadash;
```

```
angleti=(thetaone*180*2)/pi
%calculate Td(distance of maximum cone and minimum cone)
Td=abs(maxError)+abs(minError)
%calculate conicity
conicity=(Td)*cos(thetato)
%find angle max and angle min

%The end of conicity analysis for tilted cone
angxyz=zeros(2,3);
angxyz(1,1)=xyz(2,1)+maxError;
angxyz(1,2)=xyz(2,2);
angxyz(1,3)=xyz(2,3);
angxyz(2,1)=xyz(2,1)+minError;
angxyz(2,2)=xyz(2,2);
angxyz(2,3)=xyz(2,3);
Vna=zeros(2,3);
Vna(1,1)=angxyz(1,1)-xyz(1,1);
Vna(1,2)=angxyz(1,2)-xyz(1,2);
Vna(1,3)=angxyz(1,3)-xyz(1,3);
Vna(2,1)=angxyz(2,1)-xyz(1,1);
Vna(2,2)=angxyz(2,2)-xyz(1,2);
Vna(2,3)=angxyz(2,3)-xyz(1,3);
Vnnax=[Vna(1,1),Vna(1,2),Vna(1,3)];
Vnnan=[Vna(2,1),Vna(2,2),Vna(2,3)];
%find maximum angle
absvnx=sqrt(Vna(1,1)^2+Vna(1,2)^2+Vna(1,3)^2);
over2=dot(Vnnax,Vnc1);
thetaonex=acos(over2/(absvnx*absvc1));
maximumang=(thetaonex*2*180)/pi
%find minimum angle
absvnn=sqrt(Vna(2,1)^2+Vna(2,2)^2+Vna(2,3)^2);
over3=dot(Vnnan,Vnc1);
thetaonen=acos(over3/(absvnn*absvc1));
minimumang=(thetaonen*2*180)/pi
gap1=maximumang-angleti
gap2=angleti-minimumang

%coaxility analysis
diam=input('what is the diameter of tolerance');%diam is short for coaxility
 decision=Zver(:,1) .^ 2+Zver(:,2) .^ 2
[de,de1]=size(decision)
for che=1:de
if decision(che,de1)>(diam/2)^2
  s='The center is out of tolerance'
end
end

over7=dot(Vnnax,Vnnan);
thet=acos(over7/(absvnx*absvnn));
gapang=(thet*2*180)/pi;

time=etime(clock,t0)
```

```
%equation saving
%data output

pergap=abs(gap2-gap1)/gap2
pererr=abs(abs(minError)-abs(maxError))/abs(minError)
confidential=abs(pergap-pererr)/max(abs(pergap),abs(pererr))
```