

NOTE TO USERS

This reproduction is the best copy available.

UMI[®]

**QUALITY CONTROL METHODOLOGIES FOR 3D GEOMETRIC
PRIMITIVES DERIVED FROM LIDAR DATA**

by

ARTUR FIDERA, B.Eng.
Ryerson University, Toronto, 2004

A thesis

presented to Ryerson University

in partial fulfillment of the
requirements for the degree of

Master of Applied Science

In the Program of

Civil Engineering.

PROPERTY OF
RYERSON UNIVERSITY LIBRARY

Toronto, Ontario, Canada

Artur Fidera 2004 ©

UMI Number: EC52920

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI®

UMI Microform EC52920

Copyright 2008 by ProQuest LLC.

All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest LLC
789 E. Eisenhower Parkway
PO Box 1346
Ann Arbor, MI 48106-1346

BORROWER'S PAGE

Ryerson University requires the signature of all persons using or photocopying this thesis.
Please sign below, and give address and date.

Signature:

Address:

Date:

_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

ABSTRACT

QUALITY CONTROL METHODOLOGIES FOR 3D GEOMETRIC PRIMITIVES

DERIVED FROM LIDAR DATA. Artur Edward Fidera, M.A.Sc., Civil Engineering.

Ryerson University, Toronto, 2004.

The purpose of this study is to develop algorithms with a computational ability to reliably establish with precision and accuracy the critical parameters of a solid object in space. Utilizing a least-squares adjustment method and laser scanned data, a three-dimensional computer assisted drawing (3D CAD) model is created based on algorithms for reverse engineering of the geometric primitives. The derived 3D CAD model of an object (e.g., machinery component) may be then used in the redesign, retrofitting, and updating of technical drawings. This thesis presents a unique approach to point cloud data modeling and visualization as well as numerical analysis based on stability criteria. Several statistical techniques from the literature are reviewed and implemented dealing with numerical methods using the stability of matrices as a criterion. The thesis discusses topics ranging from basic statistical analysis to advanced topics such as Singular Value Decomposition (SVD) and condition numbers. Various theories and techniques of obtaining stability criteria are described and analyzed. Tests of point cloud data revealed that combining standard numerical analysis with Condition Numbers allows for quantifying the goodness-of-fit of the results and for predicting the behaviour of the algorithms.

ACKNOWLEDGMENTS

I would like to acknowledge the enormous assistance and guidance provided by Dr. M. A. Chapman and thank him for his patience, continued support, and knowledge that he passed onto me. I would like to thank Dr. J. Li who directed and helped me to achieve my goals while coding a visual interface for data modeling and visualization. I would like to acknowledge Steven LeBlanc for his enormous help with coding the visual interface and for his assistance in refining the code. Also, thanks to Des Rogan who showed me programming tricks and helped me when there appeared to be no hope...

Finally, I would like to thank my family for the support they have provided me throughout the completion of this research. Above all, this work is dedicated to my fiancée, Barbara, without whose patience and assistance this research could have not been achieved.

TABLE OF CONTENTS

TITLE PAGE	i
DECLARATION OF AUTHORSHIP	ii
BORROWER'S PAGE	iii
ABSTRACT	iv
ACKNOWLEDGMENTS	v
TABLE OF CONTENTS	vi
LIST OF TABLES	ix
LIST OF FIGURES	ix
1.0 INTRODUCTION	x
1.1 Overview	1
1.2 Objectives	2
1.3 Scope of the Study	4
1.4 Thesis Organization	4
2.0 GEOMETRIC FORM FITTING	5
2.1 Cylinder Fitting	7
2.1.1 Approximate Parameter Estimation	8
2.1.2 Least-Squares Adjustment of a Cylinder	18
2.2 Sphere Fitting	19
2.2.1 Calculating Approximates for Sphere	19
2.2.2 Least-Squares Adjustment of a Sphere	22
3.0 ADVANCED NUMERICAL ANALYSIS OF A GOODNESS-OF-FIT	27
3.1 Advanced Analysis using Condition Numbers (Stability) as a Criterion	28

3.1.1 Linear Least Squares Problems.....	29
3.1.2 Condition Number	32
3.1.2 Eigenvalues.....	35
3.1.4 Eigenvectors.....	36
3.1.5 Gershgorin Disks	37
3.1.6 The Cayley-Hamilton Theorem.....	38
3.1.7 Characteristic Equation.....	38
3.1.8 Matrix Norm	40
3.1.9 L1 Matrix Norm.....	41
3.1.10 L2 Matrix Norm.....	42
3.1.11 Frobenius Matrix Norm	42
3.1.12 L Infinity Matrix Norm.....	43
3.1.13 Spectral Radius	43
3.1.14 EISPACK Matrix Norm.....	45
3.1.15 Householder and Givens Transformation Method.....	45
3.2 QL Algorithm - Factorization of A	48
3.3 Singular Value Decomposition	49
3.4. Chapter Summary	50
4.0 EXPERIMENTATION.....	52
4.1 Testing the g parameter	54
4.2 Testing the Influence of the Coverage of the Pipe.....	66
4.3 Summary of Results.....	85

5.0 CONCLUSIONS AND RECOMMENDATIONS	86
5.1 Conclusions.....	86
5.2 Recommendations.....	88
REFERENCES	89
APPENDIX.....	91

LIST OF TABLES

Table 1: G-test-1 Summary Table.....	58
Table 2: G-test-2 Summary Table.....	61
Table 3: G-test-3 Summary Table.....	64
Table 4: Coverage-150 degrees, Summary table	70
Table 5: Coverage-130 degrees, Data Summary	74
Table 6: Coverage-100 degrees, Data Summary	78
Table 7: Coverage-70 degrees, Data Summary	81
Table 8: Coverage-20 degrees, Data Summary	84
Table 9: Results of Diameter Measurements by Two Different Software Packages (Unit: mm)	85

LIST OF FIGURES

Figure 2.1: Flow of logic.	5
Figure 2.2: Direction of stretch in data	8
Figure 2.3: Resolving quadrants.	14
Figure 4.1: Preset for Data Kreator.....	52
Figure 4.2: Preset for Data Flipper.	53
Figure 4.3: G-test-1, Data Kreator.	54
Figure 4.4: G-test-1 translation along z-axis by -5 units.	55
Figure 4.5: G-test-1 visualisation of data.....	56
Figure 4.6: G-test-2, Data Flipper.....	60
Figure 4.7: G-test-3, Data Flipper.....	63
Figure 4.8: Coverage-150 degrees, Data Kreator.	67
Figure 4.9: Coverage-150 degrees, Data Flipper	68
Figure 4.10: Coverage-150 degrees, Data Visualization	69
Figure 4.11: Coverage-130 degrees, Data Kreator	71
Figure 4.12: Coverage-130 degrees, Data Flipper.....	72
Figure 4.13: Coverage-130 degrees, Data Visualization	73
Figure 4.14: Coverage-100 degrees, Data Kreator.	75
Figure 4.15: Coverage-100 degrees, Data Flipper.....	76
Figure 4.16: Coverage-100 degrees, Data Visualization.	77
Figure 4.17: Coverage-70 degrees, Data Kreator.	79
Figure 4.18: Coverage-70 degrees, Data Visualization.	80
Figure 4.19: Coverage-20 degrees, Data Kreator.	82
Figure 4.20: Coverage-20 degrees, Data Visualization.	83

1.0 INTRODUCTION

1.1 Overview

While engineering traditionally involves the design and construction of a product, engineers are occasionally faced with the problem of redesigning existing structures. Reverse engineering is the process of creating a (digital) design model from an existing product. For example, the concept of reverse engineering may be applied in the redesigning of chemical plants, oil refineries, sewage plants, and nuclear plants. Furthermore, reverse engineering may also be used to collect data on existing structures including road networks and urban areas. The process of reverse engineering includes the development of three-dimensional (3D) models, which allows for a realistic representation of original structures. Although reverse engineering is a useful method, it may lack in robustness and reliability. The data used in the process of reverse engineering are often poorly defined and incomplete, which compromises the quality and integrity of the model. Consequently, there is a need for more efficient metrology tools that would address these issues. This thesis presents a mathematical method, which aims to increase the robustness and reliability of the developed model. More specifically, the purpose of this study is to calculate the critical parameters of a geometric primitive (i.e., a cylinder or a sphere) based on partial and poorly distributed data in order to describe the spatial orientation and radius of such forms.

When establishing the critical parameters of a solid object in space, the robustness and convergence of the solid model are compromised by many factors. An uneven distribution of the data is one of the most significant factors that may jeopardize the accuracy of the model. In a real life situation, Light Detection and Ranging (LIDAR)

systems are able to sense only a portion of the surface of the solid. Thus, the estimated geometric parameters are not always close to the actual values of the desired parameters. Furthermore, the processes of shape recognition are not only lengthy but they produce a final product that is difficult to analyze. As such, the derived product is usually not the most precise and accurate model of reality. Human error and an overwhelming amount of data from laser scanners further compromise the accuracy of the model. This study offers an alternative approach to the management, utilization, modeling, and visualization of redundant and voluminous data sets through the use of least-squares adjustment process and automatic solid recognition with reliable numerical analysis related to the calculated parameters.

1.2 Objectives

In this study, algorithms are developed with a computational ability to reliably establish the precision and accuracy of the critical parameters of a solid object in space. Historically, least-squares fitting of primitives other than lines and planes has not been studied in detail. The most commonly studied geometric primitives are ellipsoids, cylinders, spheres, cones, cuboids, and tori (Kwon et al., 2002; Lukacs et al., 1997; Watson, 2000; Werghi et al., 2000; Werghi et al., 2000). In this thesis, the fitting of a cylinder is presented and studied. Other common forms such as sphere can be viewed as trivial cases of a cylinder.

Applying a least-squares adjustment method to terrestrial lidar data, a 3D CAD model of a cylinder is created based on algorithms for reverse engineering of the primitives. The created 3D CAD model may be then used in the redesign, retrofitting,

and updating of technical drawings. In comparison to other methods such as photogrammetry, the use of LIDAR technology allows for efficient data collection. Although the precision and accuracy of LIDAR is inferior to that of photogrammetry, the number of collected points and the rate of scanning using LIDAR are significantly better than in other methods. This thesis presents a unique approach to point cloud data modeling and visualization as well as numerical analysis based on stability criteria. Different methods of calculating matrix norms are reviewed and implemented from the literature dealing with statistics using stability of the matrices as a criterion. Fundamental numerical analysis methods as well as advanced topics such as Singular Value Decomposition (SVD) and condition numbers are studied to establish the quality of the results. Various techniques of obtaining stability criteria, theories and their background are described and analyzed. This study presents a detailed overview of the problem, reviews the major areas of interest in numerical analysis and stability criteria, and details some of the principal algorithms used. It also examines the practical aspects of applied numerical analysis in programming and some obstacles involved in the process. This thesis describes least-squares fitting, modeling and numerical analysis of simple three dimensional primitives such as lines, planes and, above all, spheres and cylinders. The problem can be decomposed into five logical steps: form identification, approximate value generation, least-squares fitting, which is described briefly, data modeling, and numerical analysis based on stability criteria.

1.3 Scope of the Study

This work is an attempt to analyze least-squares adjustment algorithms usable for robust solid fitting in 3D. The rationale is to create a tool that utilizes an efficient and robust least-squares adjustment technique, which allows for the creation of reliable 3D CAD models with minimal human intervention in the process. Numerical analysis and stability criteria are applied to ensure and enforce the calculation of the critical parameters of the solid.

1.4 Thesis Organization

In Chapter 2, the least-squares adjustment of cylinders and spheres is reviewed. Detailed algorithms are presented and calculation of the approximate values of critical parameters (i.e., g -offset, μ -orientation parameter in x-y plane, ϑ -orientation parameter in zenith, α -orientation parameters about normal, and k -curvature) is examined. Chapter 3 presents the theoretical background of quality control methodologies available for statistical analysis. In addition, Singular Value Decomposition and techniques for calculating condition numbers are reviewed. In Chapter 4, tests of critical parameters are performed with distance and coverage of the pipe as the changing variables. It was determined that the condition numbers and the use of standard deviations and correlation-coefficient matrices in a controlled environment permitted the analysis of the solution and the establishment of its behaviour as changes were introduced into the point data cloud.

2.0 GEOMETRIC FORM FITTING

The software presented in this paper has been developed for geometric least-squares fitting of primitives. In particular, methods have been developed for the fitting of spheres and cylinders with numerical analysis. The algorithms for cylinder and sphere adjustments are presented in the following sections.

While the method of least-squares fitting of lines and planes is well understood and documented in the photogrammetry and mathematical literature (Fausett, 1999), the fitting of spheres and cylinders has not been extensively studied. Marshall et al. (2001) applied least-squares fitting to various second order surfaces including spheres, cylinders, cones, and tori. The authors used an angular parameterization of a model in order to describe the spatial orientation and location of the above-mentioned figures. Although the model was useful in fitting the primitives, it required the calculation of the first approximates from the point data cloud, since the point data distribution in the cloud data was uneven and potentially biased.

Various techniques may be applied to the task of surface reconstruction from 3D range data. Figure 1 depicts the flow of some of the methods used in the process of converting an imaged scene into a surface model.

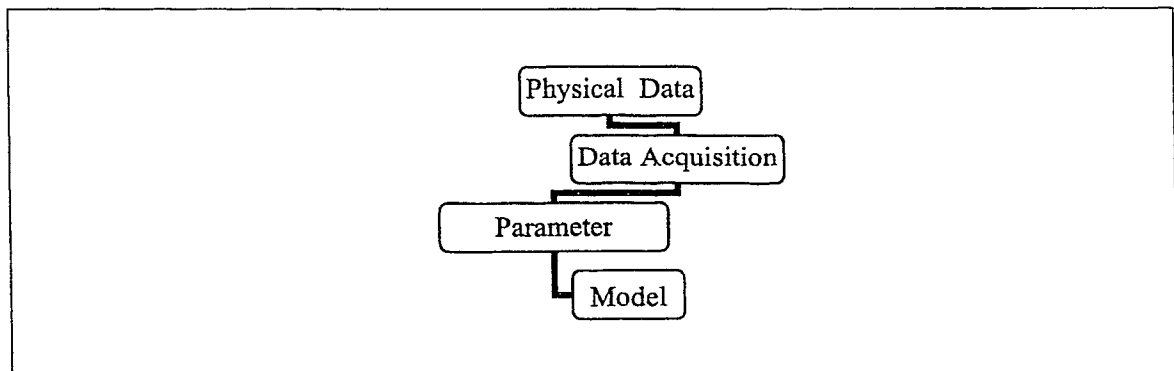


Figure 2.1: Flow of logic.

In the process of converting a scene into a surface model, data are first acquired from the real world using a sensing device, such as a laser scanner. The laser scanner acquires data, which are stored as an x, y, and z component of each point. This study presents a model of least-squares adjustment capable of reconstructing a figure based on partial point cloud data. A medium to high data set is used for this study. The physical object in this trial is a pipe 20mm in diameter and 200mm in length; the data exist for only 40% of the object and consist of approximately 29,000 points. The problem becomes one of extracting an estimate of the sampled surfaces from the limited range of data.

Given the relative lack of research in the area of least-squares adjustments in the field of solids, this thesis presents a robust approach to point cloud data fitting with advanced statistical analysis. The unique contributions of this study include methods for estimating first approximates of the parameters related to cylinders and advanced numerical analysis of the derived parameters.

Mathematical problems and calculations in engineering are plagued with numerous errors. In numerical solutions of linear equations, several sources of the inaccuracy are encountered even when exact methods are used (Faddeev & Faddeeva, 1963). The finite accuracy of the computer introduces truncation errors during the computational processes. In addition, rounding errors are present when the computational limits of the machine are reached. Mathematical models are merely an approximation attempting to model and mimic real world situations. Given that

calculations are frequently plagued with observational errors and blunders, it is often surprising that a set of approximate values can be calculated at all.

The models often used in linear and linearized systems are vectors and matrices. In order to make a statement about computed solutions and errors imbedded in them, it is necessary to describe the relative dependency between solutions using, for example, the condition numbers, norms, and errors or other stability criteria.

2.1 Cylinder Fitting

The approximates of the cylinder are determined by first calculating the center line of rotation, which is the first and most challenging step of this process. Initially, the method of least squares adjustment was applied to approximate the center line of the cylinder. The calculation of the approximate value of the center line was attempted by using parallel projection of the point cloud onto x-y, x-z, and y-z planes. Following this, least-squares adjustments of the line were applied. The equations of the three lines were determined from the corresponding three projections. The equations of the lines were transformed into equations of the planes, which are perpendicular to the corresponding projected planes. Thus, the intersection of the artificially created planes was the actual center line of the cylinder (Anderson et al., 1988). However, this method of calculating the approximates was unsuccessful since the laser scanned data were unevenly distributed over the object of interest. Furthermore, the high correlation between the shape of the figure and laser position distorted the orientation of the center line in 3D space. This caused the approximate of the center line to be highly inaccurate, which prevented the final adjustments from converging.

2.1.1 Approximate Parameter Estimation

Another attempt was made using Principal Component Analysis (PCA). This is a technique used to find the directions in which a cloud of data points is oriented (Jensen 2004).

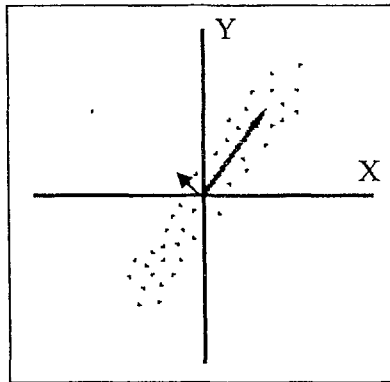


Figure 2.2: Direction of stretch in data

These directions define the basic orientation of the data and are, thus, important for establishing orientation of parameters (e.g., of a line) in space.

PCA is utilized in statistics to extract the main relations in data of high dimensionality. The general method of determining the principal components of a data set is by calculating the eigenvectors of the data correlation matrix. These vectors indicate the directions in which the data cloud is stretched the most. The projections of the data along the eigenvectors are the principal components. The corresponding eigenvalues give an indication of the amount of information the respective principal components represent. Principal components corresponding to large eigenvalues represent a great amount of information in the data set and, thus, are very informative

with regards to the relations between the data points. A 3-D case of PCA was employed to determine the first approximation of the line in 3D space.

PCA is a powerful method of analyzing correlated multidimensional data (Byne et al., 1980). The data from the laser scanner has a certain degree of redundancy (Fung & LeDrew, 1987). However, the method adopted in this project was a special case of A Zero Correlation, Rational Transform form of PCA. The principal components are based on the eigenvectors of the covariance the correlation matrix. The variance-covariance matrix C_x can be defined as:

$$C_x = \frac{1}{N-1} \sum_{i=1}^N (X_i - M) * (X_i - M)^T \quad (2.1)$$

where X is a given set of n -dimensional variables with the mean vector given as \bar{M} . N is the number of points used. Each component and N is the number of a point. Each component Y_i is denoted by

$$Y_i = a_{i1}X_1 + a_{i2}X_2 + a_{i3}X_3 + \dots + a_{in}X_n = a_i^T X \quad (2.2)$$

a_i^T is the transpose of the normalized eigenvectors of the matrix C_x . The complete transformation can be shown as

$$W = A^T * I \quad (2.3)$$

where A is the matrix of eigenvectors which gives the covariance matrix C_Y of W by

$$C_Y = A * C_X * A^T \quad (2.4)$$

The C_Y matrix will be a diagonal matrix, in which the elements are eigenvalues of C_X

$$C_Y = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \lambda_N \end{bmatrix} \quad (2.5)$$

where $\lambda_1 > \lambda_2 > \lambda_3 > \dots > \lambda_N$

The method of principal components transformation was utilized to successfully approximate the center line of the cylinder in 3D space. PCA properties were used to establish the orientation of the scanned cylinder, which enabled the definition of the directions in which the cloud of data points is stretched. The direction of the axis of the cylinder was calculated by normalizing the direction vector of the line. This was based on the center line previously determined from PCA.

$$\vec{r} = \vec{r}_1 + t\vec{m} \quad \text{where } t\vec{m} = (at, bt, ct) \quad (2.6)$$

The \vec{r} represents the vector equation of the line through the point with position vector \vec{r}_1 , in the direction \vec{m} where $t \in \mathbb{R}$.

and the parametric form of the line:

$$x_1 = x_0 + at \quad (2.7)$$

$$y_1 = y_0 + bt \quad (2.8)$$

$$z_1 = z_0 + ct \quad (2.9)$$

and transforming the line into direction vector \vec{a} , where \vec{a} : is a direction vector of a approximated center line of the point cloud data

$$\vec{a} = \vec{m} = (a, b, c) \quad \text{and } |\vec{a}| = 1 \quad (2.10)$$

The next step in calculating the approximates of the parameters for the cylinder involves finding the normal vector to the line going through the origin of the coordinate system.

$$\vec{n} = (\vec{n}_x, \vec{n}_y, \vec{n}_z) \quad \text{and } |\vec{n}| = 1 \quad (2.11)$$

and parametrized vector is:

$$\vec{n} = (\cos \mu * \sin \vartheta, \sin \mu * \sin \vartheta, \cos \vartheta) \quad (2.12)$$

(μ -orientation parameter in x-y plane, ϑ -orientation parameter in zenith)

where Cartesian components of the vector are:

$$\bar{n}_x = x_0 - x_1 - \frac{[a,b,c] \cdot \overline{P_i P_j}}{[a,b,c] \cdot [a,b,c]} * a \quad (2.13)$$

$$\bar{n}_y = y_0 - y_1 - \frac{[a,b,c] \cdot \overline{P_i P_j}}{[a,b,c] \cdot [a,b,c]} * b \quad (2.14)$$

$$\bar{n}_z = z_0 - z_1 - \frac{[a,b,c] \cdot \overline{P_i P_j}}{[a,b,c] \cdot [a,b,c]} * c \quad (2.15)$$

It is important to note that the dot product of vector \vec{a} and vector \vec{n} must be equal to zero to satisfy orthonormality continuous of these vectors.

$$(\vec{a} \cdot \vec{n}) = 0 \quad (2.16)$$

This method is employed to verify the fact that vector \vec{n} is normal to vector \vec{a} and includes in its set of points the origin of the coordinate system. Moreover, \vec{n} is also a unit vector calculated along the shortest distance between direction vector \vec{a} and the origin. Therefore, the next calculated parameter is the shortest distance between the origin and the line.

$$PR = \frac{|\overline{PQ} \cdot \vec{m}|}{|\vec{m}|} \quad (2.17)$$

$$d = QR = \sqrt{(PQ)^2 - (PR)^2} \quad (2.18)$$

where P is a point on the line, point Q is the origin of the coordinate system $Q=(0,0,0)$, and R is the closest point on the line to the origin.

The next step involves using the approximated parameters to calculate their parametrized versions of approximates. The parametrization of the approximated \bar{n} vector is accomplished by converting it from the right-handed Cartesian coordinate system into the polar coordinates system parametrized approximates to allow for polar coordinates system nonlinear least-squares adjustment.

$$r = \sqrt{x^2 + y^2} \quad (2.19)$$

$$\mu = \tan^{-1} \frac{y}{x} \quad \text{or} \quad \mu = \Pi + \tan^{-1} \frac{y}{x} \quad (2.20)$$

and

$$\vartheta = \tan^{-1} \frac{r}{z} \quad \text{or} \quad \vartheta = \Pi + \tan^{-1} \frac{r}{z} \quad (2.21)$$

where Π is used to resolve the angle position if angle is located in 1st, 2nd, 3rd or 4th quadrant.

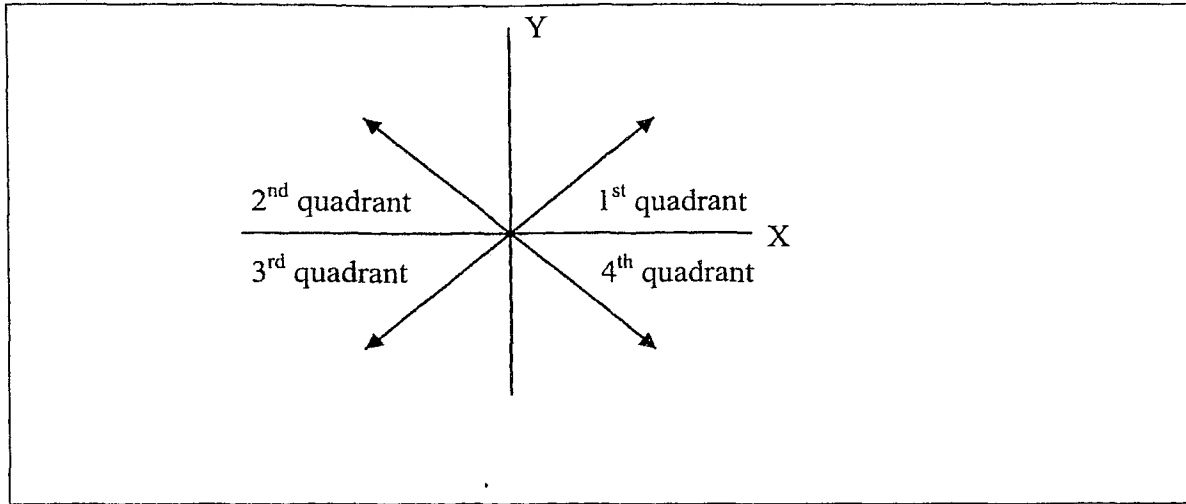


Figure 2.3: Resolving quadrants.

ϑ is the angle between \vec{n} vector and the z-axis and μ is the angle between the projection of vector \vec{n} onto x-y plane and the x-axis.

The process resolves two special angles μ and ϑ out of the five actual parametrized approximates subsequently used in the adjustment process.

Taking the first derivative of \vec{n} from Equation (2.12) with respect to angle μ gives us a n'' vector.

$$n'' = (-\sin \mu * \sin \vartheta, \cos \mu * \sin \vartheta, 0) \quad (2.22)$$

The second derivative of a \vec{n} vector from Equation (2.22) with respect to angle ϑ and then to angle μ produces vector $\overline{n''''}$:

$$\overline{n''''} = (-\sin \mu * \cos \vartheta, \cos \mu * \cos \vartheta, 0) \quad (2.23)$$

Taking a first derivative of \bar{n} from Equation (2.12) with respect to angle ϑ gives us a n^{ϑ} vector.

$$n^{\vartheta} = (\cos \mu * \cos \vartheta, \sin \mu * \cos \vartheta, -\sin \vartheta) \quad (2.24)$$

The second derivative of a \bar{n} vector, from Equation (2.24) with respect to angle ϑ and then to angle ϑ produces vector $\overline{n^{\vartheta\vartheta}}$:

$$\overline{n^{\vartheta\vartheta}} = (\cos \mu * \sin \vartheta, -\sin \mu * \sin \vartheta, -\cos \vartheta) \quad (2.25)$$

The second derivative of a \bar{n} vector from Equation (2.22) with respect to angle μ and then to angle μ produces vector $\overline{n^{\mu\mu}}$:

$$\overline{n^{\mu\mu}} = (-\cos \mu, \sin \mu, 0) \quad (2.26)$$

By equating and manipulating the partial derivatives of vector \bar{n} from from Equations (2.22 - 2.26) the vector $\overline{n^{\mu\vartheta}}$ is produced.

$$\overline{n^{\mu\vartheta}} = (-\sin \mu, \cos \mu, 0) = \frac{n^{\mu\vartheta}}{\sin \vartheta} \quad (2.27)$$

The next step in determining the parametrized approximates for the cylinder is to compute angle α , which is the angle between vector \vec{a} and vector \vec{n}^θ .

$$\alpha = \cos^{-1} \left(\frac{\vec{a} \cdot \vec{n}^\theta}{|\vec{a}| |\vec{n}^\theta|} \right) \quad (2.28)$$

Next, the approximated radius of the cylinder must be calculated. This is accomplished by rotating the data and aligning the approximated center line of the data cloud along the x-axis.

$$\begin{bmatrix} x_j \\ y_j \\ z_j \end{bmatrix} = R_{\nu'} * R_{\mu'} * R_{\alpha'} * \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \quad (2.29)$$

where:

$$R_\alpha = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.30)$$

$$R_\theta = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.31)$$

$$R_{\mu} = \begin{bmatrix} \cos \mu & \sin \mu & 0 \\ -\sin \mu & \cos \mu & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.32)$$

In the next step, the average distance between each point of the data cloud and the x-axis is calculated. The average distance is the average radius of the cylinder. Then, the approximate radius is parametrized by its inverse parameter k (curvature):

$$R = \frac{1}{k} \quad (2.33)$$

Now, by combining the shortest distance between the origin of the coordinate system and the original center line of the data cloud from Equation (2.18) and subtracting the approximated radius Equation (2.33), the last required parametrized approximate may be calculated. This approximate value represents the shortest distance between the origin of the coordinate system and the surface of the cylinder:

$$g = d - R \quad (2.34)$$

Following these steps has provided the parametrized approximates required for the least-squares adjustment of the cylinder. The parametrized approximates in terms of g , μ , ϑ , α , and k have been calculated using Equations (2.34), (2.20), (2.21), (2.28), and (2.33), respectively.

2.1.2 Least-Squares Adjustment of a Cylinder

The least-squares adjustment of a right circular cylinder is completed by first creating a Jacobian matrix A. The derivatives of the parameterized approximates with respect to the distance from the symmetry axis to the surface of the cylinder are given as follows:

$$A(i,0) = \frac{\partial \tilde{d}}{\partial g} = k * (g - (p \cdot \bar{n})) + 1 \quad (2.35)$$

$$A(i,1) = \frac{\partial \tilde{d}}{\partial \mu} = -k * (g * (p \cdot n'') + (p \cdot \bar{a}) * (p \cdot \overline{n'''} * \cos \alpha + \overline{n'''} * \sin \alpha)) - (p \cdot n'') \quad (2.36)$$

$$A(i,2) = \frac{\partial \tilde{d}}{\partial \vartheta} = k * ((p \cdot \bar{a}) * (p \cdot n) \cos \alpha - g * (p \cdot n^\vartheta)) - (p \cdot n^\vartheta) \quad (2.37)$$

$$A(i,3) = \frac{\partial \tilde{d}}{\partial \alpha} = k * (p \cdot \bar{a}) * (p \cdot n^\vartheta * \sin \alpha - \overline{n''} * \cos \alpha) \quad (2.38)$$

$$A(i,4) = \frac{\partial \tilde{d}}{\partial k} = \frac{1}{2} (|p|^2 - 2g(p \cdot \bar{n}) - (p \cdot \bar{a})^2 + g^2) \quad (2.39)$$

where $i=0$ to $N-1$ and N is the number of points.

The p is defined as a point in 3D with coordinates read in from data point.

These are associated with a residual vector:

$$w(i) = -\frac{k}{2} * (|p|^2 - 2g(p \cdot \bar{n}) - (p \cdot \bar{a})^2 + g^2) + g - (p \cdot \bar{n}) \quad (2.40)$$

The parametric least-squares method is used for set of equations as:

$$\partial x = (A^T A)^{-1} A^T w \quad (2.41)$$

where ∂x is the vector of corrections to the approximate values.

The method of least-squares adjustment for a right circular cylinder fitting produced satisfactory results in all trials. This method effectively calculates the parametrized approximates and allows least-squares adjustments to converge successfully. It should be noted at this point that the sphere is the trivial form of the cylinder model.

2.2 Sphere Fitting

In order to proceed with least-squares adjustments for the sphere, the approximates of the center of gravity in 3D space need to be determined. In addition, the radius of the sphere must be computed using parameter k . Once the parameters have been determined, they are used to calculate the parametrized approximates. Finally, the process of least-squares adjustment may be applied. It should be noted that the sphere problem is a trivial form of the cylinder model.

2.2.1 Calculating Approximates for Sphere

The approximates of the sphere are determined by first calculating the center of gravity of the data. The approximate of the center of gravity was determined by calculating the average value for the x, y, and z coordinates of the point cloud data.

$$x_{avg.} = \frac{1}{N} \sum_{i=0}^{N-1} x_i \quad (2.42)$$

$$y_{avg} = \frac{1}{N} \sum_{i=0}^{N-1} y_i \quad (2.43)$$

$$z_{avg} = \frac{1}{N} \sum_{i=0}^{N-1} z_i \quad (2.44)$$

Following this, the approximation of the radius is estimated based on the calculated distance difference between the center of gravity and each point in the point cloud data.

$$R = \frac{1}{N} * \sum_{i=0}^{N-1} \sqrt{(x_i - x_{avg})^2 + (y_i - y_{avg})^2 + (z_i - z_{avg})^2} \quad (2.45)$$

The next step in calculating the approximates of the parameters for the sphere involves finding the shortest distance between the origin of coordinate system and the data cloud.

$$g = \left| \sqrt{(x_i - 0)^2 + (y_i - 0)^2 + (z_i - 0)^2} \right|_{\min} \quad (2.46)$$

The \vec{n} vector which is the vector connecting the origin of the coordinate system and the center of gravity is established by:

$$\vec{n} = (x_{avg}, y_{avg}, z_{avg}) \quad (2.47)$$

The next step involves using the approximated parameters to calculate their parametrized versions of approximates. The parametrization of the approximated \bar{n} vector is accomplished by converting it from a right-handed Cartesian coordinate system into a polar coordinate system parametrized approximates to allow for a polar coordinate system non-linear least-squares adjustment Equations (2.19-2.21).

Taking the first derivative of \bar{n} from Equation (2.47) with respect to angle μ gives us a n'' vector.

$$n'' = (-\sin \mu * \sin \vartheta, \cos \mu * \sin \vartheta, 0) \quad (2.48)$$

Taking a first derivative of \bar{n} from Equation (2.47) with respect to angle ϑ gives us a n^ϑ vector.

$$n^\vartheta = (\cos \mu * \cos \vartheta, \sin \mu * \cos \vartheta, -\sin \vartheta) \quad (2.49)$$

In the next step, the average distance between each point of the data cloud and the x-axis is calculated. The average distance is the average radius of the cylinder. Then, the approximate radius is parametrized by:

$$R = \frac{1}{k} \quad (2.50)$$

Following these steps has provided the parameterized approximates required for the least-squares adjustment of the cylinder. As such, the parametrized approximates are computed in terms of g , μ , ϑ , and k using Equations (2.46), (2.49), (2.50), and (2.53).

2.2.2 Least-Squares Adjustment of a Sphere

The least-squares adjustment fitting for a sphere is begun by creating a Jacobian matrix A . The following are the partial derivatives of the parameterized approximates with respect to the distance from the origin to the surface of the sphere:

$$A(i,0) = \frac{\partial \tilde{d}}{\partial g} = k * (g - (p \bullet \bar{n})) + 1 \quad (2.51)$$

$$A(i,1) = \frac{\partial \tilde{d}}{\partial \mu} = (-k * g - 1) * (p \bullet n'') \quad (2.52)$$

$$A(i,2) = \frac{\partial \tilde{d}}{\partial \vartheta} = (-k * g - 1) * (p \bullet n^{\vartheta}) \quad (2.53)$$

$$A(i,3) = \frac{\partial \tilde{d}}{\partial k} = \frac{1}{2}(|p|^2 - 2g(p \bullet \bar{n}) + g^2) \quad (2.54)$$

These are associated with a misclosure vector:

$$w(i) = -\frac{k}{2} * (|p|^2 - 2g(p \bullet \bar{n}) + g^2) + g - (p \bullet \bar{n}) \quad (2.55)$$

The parametric least-squares formulation is used for set of equations:

$$\partial x = (A^T * A)^{-1} * A^T * w \quad (2.56)$$

The method of least-squares adjustment for a sphere fitting produced satisfactory results in all trials. This method effectively calculates the parametrized approximates and allows least-squares adjustments to converge successfully.

The parametrization of the solid plays a major role in the process of solid fitting. There are different general approaches to this problem including Algebraic Fitting, Taubin's Fitting, and Euclidian Fitting. Each of these approaches has both limitations and strengths, which are described in the following section.

The Algebraic Fitting (AF) is based on the approximation of the Euclidian distance by the algebraic distance and is described by this equation:

$$dist_i(\bar{x}_r, Z(f)) = f(\bar{x}_r) \quad (2.57)$$

Given the algebraic distance for each point, using the least-squares method to minimize the distance between best surface fit and the data is expressed as:

$$\frac{1}{P} \sum_{p=1}^P dist(\bar{x}_r, Z(f)) \rightarrow \text{minimum} \quad (2.58)$$

Equation (2.61) can be formulated as an eigenvector problem. Constraining the eigenvectors of the parameter set $\{a_{ijk}\}$ avoids the trivial solution $\{a_{ijk}\} = \bar{0}$ and any possibilities of the multiple solution. The algebraic approach has high computational efficiency if the closed-form solution is available, which is generally the case. However, the results obtained from the algebraic method are unsatisfactory in most cases.

Therefore, the algebraic method is often used for first approximation if the situation does not necessitate a statistical approach that is more time consuming and advanced.

Taubin's Fitting (TF) uses the first order approximation of Equation (2.61) to estimate $dist_{\tau}(\bar{x}_p, Z(f))$.

$$dist_{\tau}(\bar{x}_p, Z(f)) = \frac{|f(\bar{x}_p)|}{\|\nabla f(\bar{x}_p)\|} \quad (2.59)$$

The Taubin's distance algorithm does not require lengthy iterative calculations. The first order approximation of the Taubin's distance algorithm is the exact distance. However, the approximated Taubin's distance is biased due to a lack of internal consistency. The perfect example of this phenomenon is a data point \bar{x}_p which is close to a critical point of the polynomial, i.e., $\|\nabla f(\bar{x}_p)\| \approx 0$, but $f(\bar{x}_p) \neq 0$. The distance between the point and the described polynomial becomes large due to the fact that magnitude is extremely close to zero. However, the fact that the distance is not zero is certainly a limitation, which leads to an answer which is numerically highly unstable.

Euclidian Fitting (EF) uses the Euclidian distance instead of approximated distances, which is invariant to transformation in Euclidian space. The use of the Euclidian distances allows for calculating a non-biased estimator. A closed form expression exists for primitive curves and surfaces, ellipses, planes, cylinder cones, and ellipsoids. The Euclidian distance is obtained from a point to the origin of the Cartesian coordinate system in order to calculate a closed form expression, as described in detail for cylinder and sphere in Sections 2.0 to 2.2.2.

In its general expression, the Euclidian distance is a highly complicated equation and there is no known closed form expression for solving the problem. Therefore, an

iterative optimization procedure must be carried out in order to compensate for this fact. Given the Euclidian distance $dist_E(\bar{x}_p, Z(f))$ for each point, the following steps are used to obtain the solution. First, an initial estimate for the parameters $\{a_{ijk}\}$ of the Euclidian fitting is obtained. Next, $\{a_{ijk}\}^{[s]}$, $s = 0, 1, \dots$ is updated using the algorithms presented in Sections 2.1.2 and 2.2.2, which involves minimizing the least-squares of Euclidian distances for all data points. Finally, each $\{a_{ijk}\}^{[s+1]}$ is evaluated by a M-estimator ζ on the basis of $dist_E(\bar{x}_p, Z(f))$. If $\zeta(\{a_{ijk}\}^{[s+1]}) < \zeta(\{a_{ijk}\}^{[s]})$, $\{a_{ijk}\}^{[s+1]}$ is satisfied, the adjustment is carried out. Otherwise, the adjustment is terminated and $\{a_{ijk}\}^{[s]}$ becomes the final solution.

In order to understand data distribution and data errors, it is important to appreciate the basic concepts behind laser technology. The acronym LASER means Light Amplification by Stimulated Emission of Radiation. There are two major types of laser scanners: triangulation scanners and ranging scanners. LIDAR scanners are of the ranging scanners family. The scanner used to collect the data for this study was a LIDAR scanner. LIDAR scanners often use a pulsed laser to measure the range to a point on an object's surface. The LIDAR scanner is any ranging laser scanner that has four observables, range ρ , elevation angle α , horizontal direction θ , and intensity that are relative to the scanner's internal coordinate system. The difference in time (Δt) between the pulse and the detection of the returning energy is used to calculate the distance (ρ) to the target using the following equation:

$$\rho = \frac{c * \Delta t}{2} \quad (2.60)$$

The obtained parameters are used to calculate the x, y, and z point in a Cartesian coordinate system. The triangulation and spatial resection algorithms are implemented to carry out those calculations. The distance from the scanner to the target is measured based on Equation (2.63). The triangulation is based on an active stereoscopic principle and the distance to the object is calculated based on the angular orientation of the light source and the relative orientation of the receiving unit.

Data quality and distribution are critical issues in the investigation of the calculation of the critical parameters. In this thesis, the data received from J-Tech (www.j-techdesign.com) were already compiled in a Cartesian coordinate frame; the observables were x, y, z and intensity. The object that the data were derived from was a section of cylindrical pipe; 20 mm in diameter and 200 mm in length. The point data cloud covered approximately 40% of the surface area of the pipe and consisted of approximately 29,000 points. Due to the limited nature of the LIDAR system, namely FOV (field of view) and distance from the target, the point cloud data received had a mid to heavy density and the point data became sparser as the scan approached the edges of the cylinder.

Different methods of reverse engineering of the primitives allow us to produce mathematical models. However, these models have finite accuracy and precision as well as errors inherited from data acquisition systems. One example is the LIDAR system. The inherent errors as well as errors in calculation are not necessarily controlled in the process of least-squares adjustment. In some cases, the best fit is not the actual primitive figure that is sought and its spatial orientation may be completely wrong.

3.0 ADVANCED NUMERICAL ANALYSIS OF A GOODNESS-OF-FIT

The model proposed below presents a detailed explanation of the mathematical algorithms as well as the methodology used to write a fully functional software suite. The arising model issues, including parametrization and data distribution, are presented and discussed. A cylinder and a sphere are the two geometric primitives that are used in this case study due to their frequency of occurrence in industrial settings.

The proposed model has certain design flaws. For example, the model of the cylinder is parametrized by using three spatial angles, the shortest distance between the origin and the outside of the projected cylinder as well as the radius of the cylinder. Two of the mentioned angles describe the orientation of the normal to the line passing through the origin. Angle μ describes the orientation of the normal vector in the x-y plane where the positive x axis is treated as the reference. The ϑ angle is a horizontal angle. Those two angles define a spatial orientation using a polar coordinate system. The third angle, α , i.e., the alpha angle, describes the final orientation of the cylinder. The cylinder is pivoting at the imaginary end of the normal and the alpha angle defines the cylinder orientation. The distance between the origin and the closest point on the outside shell of the cylinder as well as the radius are calculated in a very similar manner.

A detailed description of parametrization concepts and algorithms may be found in Section 2.1.1, which details the cylinder fitting. The sphere is parametrized in similar manner but without the use of the alpha angle.

A geometric least-squares fitting of spheres and cylinders was achieved by developing algorithms to calculate first approximates and by utilizing algorithms introduced by Marshall et al. (2001).

3.1 Advanced Analysis using Condition Numbers (Stability) as a Criterion

Mathematical problems and calculations in engineering are plagued with errors of all sorts. In numerical solutions of linear equations, several sources of the inaccuracy are encountered even when exact methods are used (Faddeev et al., 1963). The finite accuracy of the computer introduces truncation errors during the computational processes. In addition, rounding errors are present when we reach the computational limits of the computer on which the calculations are carried out. Mathematical models are merely a means to an end and they are only an approximation trying to model and mimic a real world phenomenon. Given that calculations are often plagued with observational errors and blunders, it is often surprising that one is able to calculate an approximated answer at all!

The models used in linear systems employ vectors and matrices. In order to make a statement about computed solutions and errors imbedded in them, it is necessary to describe the relative dependency between solutions using the condition number, norms and errors. The definition of a condition number of a matrix is derived directly from the definition of norms and errors. The use of a condition number and norms allows for an objective way of estimating the statistical significance of "Hilbert Space" or in this case the "N"-Normal Equations matrix. It also allows us to determine the relative accuracy of the solved linear system. The problem described in this thesis is a linearized least squares problem. The following definition should aid in understanding the notion of simplified linear least squares problems and error estimation techniques.

3.1.1 Linear Least Squares Problems

A typical linear least squares problem seeks to solve a linear system $A^*x = b$ where the matrix A is unsuitable for treatment by the standard methods used for square, non-singular, well-conditioned coefficient matrices.

In almost all cases involving geomatics-related data, there are many more equations than unknowns, so that A is actually a matrix with rectangular order of M rows by N columns. Generally, M is much greater than N . The right hand side may or may not be consistent, and A may not be of full rank. Such problems cannot be solved accurately using the linear approach, since the matrix is extremely overdetermined. It is possible to obtain a unique solution or many exact solutions (x) might exist. Alternatively, there might be no exact solutions, but some vectors (x) offer a better goodness-of-fit than others in the sense that they produce a smaller residual error where residual error is defined as:

$$r = b - A^*x \quad (3.1)$$

where the residual error is a measure of the error that occurs when a given approximate solution vector x is substituted into the equations of the problem being solved. For a system of linear equations, the residual errors are expressed by the vector r . When carrying out an iterative solution process, it is common practice to compute the residual error vector for each new approximate solution and to terminate the iteration successfully if the vector norm of the residual error decreases below some tolerance. It is important to realize that the residual error is not, by itself, a reliable estimate of the error in the solution of a linear system. If the residual error has small norm, it is desirable that the

solution error (between the computed x and the true solution x^*) is small. It is necessary to know the norm of the inverse of the matrix, in which case the following restriction holds:

$$\|x - x^*\| \leq \|A^{-1}\| * \|A(x - x^*)\| = \|A^{-1}\| * \|A * x - b + b - A * x^*\| = \|A^{-1}\| * \|r\| \quad (3.2)$$

Thus, when the norm of the residual error $\|r\|$ is small, there is a precise upper bound on the error in the solution. Since such matrix norms are generally not known or computable, there is a promise of continuity in the errors: as the residual is driven down, the upper limit on the approximation error is forced down (Dineen, 1998).

Once the residual error is calculated based on its vector norm, the logical operator might be put in place to determine which of the following cases are applicable to the solution. If there is a unique exact solution, each residual vector component should be equal to zero and the exact solution could be obtained. If there are multiple exact solutions, then they form a linear space, and that linear space should be described. If there are no exact solutions, then a "best" solution, which produces the smallest possible residual error, should be produced. This theory is applied in the algorithms and based on those criteria the final solution is obtained by selecting the smallest residual error with preselected threshold in order to stop iterative process as described and depicted by Euclidian Fitting (EF) algorithm described in Section 2.2.2.

A linear space (also known as linearization method) mentioned above is a collection X of "vectors", a scalar field F , (usually the real or complex field) and in the cases observed in this paper it is always the real field since we operate in spatial domain which can not be imaginary or negative i.e. the negative distances do not exist in real

world. The operations of vector addition and scalar multiplication, with the properties that X includes the zero vector; if x is in X , then so is $\lambda * x$, for any scalar λ ; and if x , y and z are in X , then so is the sum $x + y + z$.

The problem encountered in this study was solved using the least squares-adjustment method applying standardized techniques of equation linearization in geomatics problems. A square coefficient matrix can be constructed by replacing the M by N rectangular system $A * x = b$ by the square system of order N : $A^T * A * x = A^T * b$.

$$A^T * A * x = A^T * b \quad (3.3)$$

and can be further manipulated to the unweighted form of:

$$x = (A^T A)^{-1} (A^T b) \quad (3.4)$$

This linear system is known as the normal equations matrix called the N -matrix in the software created by manipulating the A -matrix called the Jacobian matrix or design matrix. If the columns of the original matrix A are independent, then $A^T * A$ has a real inverse, and the system can be solved by Gauss elimination, Cholesky decomposition or another applicable method. The method selected for this exercise was Cholesky decomposition, as it offers efficient algorithms. However, the Cholesky algorithm might not be the most robust algorithm and it fails in the imaginary domain. However, the working domain for this exercise is measurements in the real world and the spatial domain is the only domain allowable which rules out the possibility of the imaginary or negative distances. The solution x will not usually satisfy any of the original equations exactly, but it will be the estimate that minimizes the Euclidean norm (The L_2 vector norm is also known as the Euclidean vector norm or the root-mean-square vector norm)

of the residual error. The L2 vector norm is described in more details further on in this thesis in Section 3.1.10.

The normal equations may be used as a guide as to how the answer can be obtained, at least for systems with maximal rank. However, the coefficient matrix of the normal equations is sometimes ill conditioned (i.e., its condition number being the square of the condition number of A). Other methods of solving this problem are preferred, usually via the QL factorization, QR factorization or a pseudoinverse. Such methods can also handle the case where the matrix does not have full rank. However, the methods mentioned above were not selected due to their computational inefficiency and the fact that the stated problem classifies as invertible matrices which Cholesky decomposition is able to handle with ease and superior computational efficiency.

3.1.2 Condition Number

The condition number of the coefficient matrix A of a linear system is a positive number used to estimate the amount by which small errors in the right hand side b , or in A itself, can change the solution x . Small values of the condition number suggest that the algorithm will not be sensitive to errors. Large values indicate that small data or arithmetic errors may lead to enormous errors in the answer.

The condition number is defined in terms of a particular matrix norm. Many different matrix norms may be chosen, and the actual value of the condition number will

vary depending on the norm chosen. However, the general rule that large condition numbers indicate sensitivity will hold true no matter what norm is chosen.

The condition number for a matrix A is defined as

$$\text{condition}(A) = \|A\| * \|A^{-1}\| \quad (3.5)$$

If A is not invertible, the condition number is infinite.

The following are the characteristics of the condition number: it is always at least 1 and the identity matrix produces a condition number equal to 1. In addition, the condition number of any orthogonal or unitary matrix is 1.

Turing's M condition number, $M(A)$, for a matrix of order N , is defined as

$$M(A) = N * \max |A_{ii}| * \max |A_{ij}^{-1}| \quad (3.6)$$

Turing's N condition number, $N(A)$ is

$$N(A) = \frac{\text{Frob}(A) * \text{Frob}(A^{-1})}{N} \quad (3.7)$$

where $\text{Frob}(A)$ is the Frobenius matrix norm.

The Von Neumann and Goldstine P condition number is

$$P(A) = \left| \frac{\max \lambda}{\min \lambda} \right| \quad (3.8)$$

where $\max \lambda$ and $\min \lambda$ are the eigenvalues of largest and smallest magnitude, which is equivalent to using the spectral radius of A and A^{-1} . The Von Neumann and Goldstine $P(A)$ number was selected for the calculations of condition numbers. The $P(A)$ number was selected as a stability criterion due to its computational efficiency when compared to

other methods. Furthermore, the technique of obtaining the eigenvalues and eigenvectors, which are needed to calculate the condition numbers, played an important role in the selection of the method. The method of calculating the eigenvalues and eigenvectors is explained in the following chapters. Moreover, it is the least complicated method of obtaining the condition numbers without the need for highly complex and expensive statistical software packages.

There is also a condition number defined for the eigenvalue problem, which attempts to estimate the amount of error to be expected when finding the eigenvalues of a matrix A .

Given a square matrix A , the L_2 condition number $k_2(A)$ is defined as:

$$k_2 = \|A\|_2 * \|A^{-1}\|_2 \quad (3.9)$$

if the inverse of A exists. If the inverse does not exist, then we say that the condition number is infinite. Similar definitions apply for $k_{1(A)}$ and $k_{\infty}(A)$.

Condition numbers are used in error estimation for linear system problems. This study focuses on solving the linear system as described in detail in the Linear Least Squares Problems in Section 3.1.1 using the following model $A^*x = b$ with the exception that the right hand side of the equation has a small error in it. This perturbed right hand side is denoted as $b + \partial b$ and the mathematical model is rewritten as:

$$A^*(x + \partial x) = b + \partial b \quad (3.10)$$

If relative errors and the norm are used to define condition number $k(A)$ and if they are compatible with the vector norm used, it is necessary to show that:

$$\frac{\|\partial x\|}{\|x\|} \leq k(A) * \frac{\|\partial x\|}{\|b\|} \quad (3.11)$$

Thus, the condition number needs to be as small as possible, with its smallest possible value being 1 (Press, et al., 1986).

3.1.2 Eigenvalues

Eigenvalues are special values associated with a square matrix. They can be used to analyze the behavior of a matrix in multiplying any vector as well as to calculate its condition number. The formal definition of an eigenvalue of a matrix A is that it is any value λ which is a root of the characteristic equation of the matrix,

$$\det(A - \lambda * I) = 0 \quad (3.12)$$

λ is an eigenvalue of A if and only if there is a nonzero vector x , known as an eigenvector described in detail later on, with the property that

$$A * x_i = \lambda_i * x_i \quad (3.13)$$

There must also be a left eigenvector y , with the property

$$y * A = A^T * y = \lambda * y \quad (3.14)$$

The characteristic equation has exactly N roots, so a matrix has N eigenvalues. An important consideration is whether any eigenvalue is a repeated root, which determines how difficult the eigenvector computation will be. If a matrix has the maximum possible number of linearly independent eigenvectors N , then the eigenvalues and eigenvectors can be used to diagonalize the matrix. This only happens when the matrix is orthogonal. Eigenvalues of A have the following characteristics: A is singular if and only if

0 is an eigenvalue of A ; A is symmetric if all eigenvalues are real; and A is positive definite symmetric matrix if all eigenvalues are real and positive. Algorithms for computing eigenvalues include the power method and the inverse power method. The QR method or QL method are more powerful methods that can handle complex and multiple eigenvalues (Press et al., 1986).

3.1.4 Eigenvectors

A nonzero vector x is an eigenvector of the square matrix A if

$$A * x_i = \lambda_i * x_i \quad (3.15)$$

for some scalar value λ , called the associated eigenvalue.

Sometimes this eigenvector is more particularly described as a right eigenvector, so that we may also consider left eigenvectors, that is, vectors y for which it is true that

$$y * A = A^T * y = \mu * y \quad (3.16)$$

for some scalar μ .

For every eigenvalue of a matrix, there is at least one eigenvector. Every nonzero multiple of this eigenvector is also an eigenvector, but in an uninteresting way. If, and only if, an eigenvalue is a repeated root, then there may be more than one linearly independent eigenvector associated with that eigenvalue. For example, if an eigenvalue is repeated 3 times, then there will be 1, 2 or 3 linearly independent eigenvectors corresponding to that eigenvalue.

The following properties of the eigenvector have to be satisfied: if x is an eigenvector, so is $s*x$, for any nonzero scalar s . If A is singular, then it has an eigenvector associated with the eigenvalue 0, so that $A*x=0$. If x is a right eigenvector for eigenvalue λ , and y is a left eigenvector for eigenvalue μ , and λ and μ are distinct, then x and y are orthogonal, that is, $y^T * x = 0$. This property is sometimes called biorthogonality.

The following methods may be used to calculate eigenvalues in order to determine condition numbers depending on matrix properties and behaviour: Gershgorin Disks, The Cayley-Hamilton Theorem and Circulant Matrix. In addition, more advanced methods may be applied to calculate condition numbers. These are called Matrix Norms. Although Matrix Norms are more robust and descriptive of the matrix system, they also require advanced techniques and additional computational power (Press et al., 1986).

3.1.5 Gershgorin Disks

The method of Gershgorin disks provides an estimate of the size of the eigenvalues of a matrix. The accuracy of the estimate varies significantly, depending on the size of the elements of the matrix. It is most useful for matrices that are predominantly diagonally or sparse (Press et al., 1986).

Gershgorin's theorem states that the eigenvalues of any matrix A lie in the space covered by the disks $D(I)$:

$$D(I) = \left(\frac{x}{\sqrt{x - A(I, I)}} \right)^2 \leq R(I) \quad (3.17)$$

where $R(I)$ is the sum of the absolute values of the off-diagonal ($/$) elements of row I :

$$R(I) = \text{sum}(J \neq I) |A(I, J)| \quad (3.18)$$

The theorem may also be applied using columns instead of rows due to symmetry.

3.1.6 The Cayley-Hamilton Theorem

The Cayley-Hamilton Theorem guarantees that every square matrix satisfies its own characteristic equation.

For example, if A is given by:

$$A = \begin{pmatrix} 2 & 3 \\ 1 & 4 \end{pmatrix} \quad (3.19)$$

then the characteristic equation is

$$\lambda^2 - 6\lambda + 5 = 0 \quad (3.20)$$

which is not true for all values λ , only those few special values known as eigenvalues.

The Cayley-Hamilton theorem guarantees that the matrix version of the characteristic equation, with A taking the place of λ , is guaranteed to be true (Press et al., 1986):

$$A^2 - 6A + 5I = 0 \quad (3.21)$$

3.1.7 Characteristic Equation

The characteristic equation of a square matrix A is the polynomial equation:

$$\det(A - \lambda I) = 0 \quad (3.22)$$

where λ is an unknown scalar value.

The left hand side of the equation is known as the characteristic polynomial of the matrix.

If A is of order N , then there are N roots of the characteristic equation, possibly repeated, and possibly complex.

For example, if A given by Equation (3.19), then the characteristic equation is

$$\det \begin{pmatrix} 2-\lambda & 3 \\ 1 & 4-\lambda \end{pmatrix} = 0 \quad (3.23)$$

or

$$\lambda^2 - 6\lambda + 5 = 0 \quad (3.24)$$

This equation has roots $\lambda = 1$ or $\lambda = 5$.

Values of the scalar λ which satisfy the characteristic equation are known as eigenvalues of the matrix.

A property of characteristic equation of A :

- A and A^T have the same characteristic equation.

The Cayley-Hamilton Theorem guarantees that the matrix itself also satisfies the matrix version of its characteristic equation (Press et al., 1986).

3.1.8 Matrix Norm

A matrix norm is a scalar quantity, which may be thought of as a type of "magnitude" of the matrix. The norm may be used to estimate the effect of multiplying the matrix times a vector, solving a linear system, or other matrix operations. The norm is also used in the analysis of error and convergence.

A matrix norm $\| \cdot \|$ must satisfy the following four properties:

1. $\|A\| > 0$, unless $A = 0$ in which case $\|A\| = 0$;
2. $\|s * A\| = |s| * \|A\|$ for any real number s ;
3. $\|A + B\| \leq \|A\| + \|B\|$ (triangle inequality);
4. $\|A * B\| \leq \|A\| * \|B\|$ (sub-multiplicativity).

Matrix norms are most often needed when dealing with combinations of matrices and vectors. In such a case, it is important that the matrix norm and vector norm that are being used are compatible.

Any given vector norm can be used to derive a corresponding matrix norm and will be guaranteed to be compatible. This matrix norm is known as the vector-bound matrix norm. Only if the matrix norm and vector norm are compatible can one write a useful boundary condition like:

$$\|A * x\| \leq \|A\| * \|x\| \quad (3.25)$$

The following matrix norms will be described in the following sections:

1. L1 matrix norm;
2. L2 matrix norm;
3. Frobenius matrix norm;
4. L Infinity matrix norm;
5. Spectral radius;
6. EISPACK matrix norm;

3.1.9 L1 Matrix Norm

The L_1 matrix norm is a matrix norm that is vector-bound to, and hence compatible with, the L_1 vector norm. Thus, the formal definition of the norm is

$$\|A\| = \max \left(\frac{\|A * x\|}{\|x\|} \right) \quad (3.26)$$

where the vector norm used on the right hand side is the L_1 vector norm, and the maximum is taken over all nonzero vectors x . However, it is easy to show that the L_1 matrix norm has a simpler formula: $\|A\| =$ the maximum, over all matrix columns, of the sum of the absolute values of the entries in the column (Stewart, 1995).

3.1.10 L_2 Matrix Norm

The L_2 matrix norm is a matrix norm that is vector-bound to, and hence compatible with, the L_2 vector norm. Thus, the formal definition of the norm is presented in Equation (3.26).

Where the vector norm used on the right hand side is the L_2 vector norm, and the maximum is taken over all nonzero vectors x .

The L_2 matrix norm has another formulation: $\|A\| =$ the square root of the maximum absolute value of the eigenvalues of $A^T * A$.

The computation of the L_2 norm is computationally expensive, and so it is often simpler to use the easily-computed Frobenius matrix norm, which is not vector-bound to the L_2 vector norm, but is compatible with it (Stewart, 1995).

3.1.11 Frobenius Matrix Norm

The Frobenius matrix norm is a matrix norm that has the simple formula: $\|A\| =$ the square root of the sum of the squares of all the entries of the matrix.

The Frobenius matrix norm is not a vector-bound matrix norm, although it is compatible with the L_2 vector norm, and much easier to compute than the L_2 matrix norm.

The Frobenius matrix norm is sometimes called the Schur matrix norm or Euclidean matrix norm (Press et al., 1986).

3.1.12 L Infinity Matrix Norm

The L_∞ infinity matrix norm is a matrix norm that is vector-bound to, and hence compatible with, the L_∞ infinity vector norm. Thus, the formal definition of the norm is as presented in Equation (3.26).

Where the vector norm used on the right hand side is the L_∞ vector norm, and the maximum is taken over all nonzero vectors x . However, it is easy to show that the L_∞ matrix norm has a simpler formula: $\|A\| =$ the maximum, over all matrix rows, of the sum of the absolute values of the entries in the row (Stewart, 1995).

3.1.13 Spectral Radius

The spectral radius of a matrix is the magnitude of the largest eigenvalue of a matrix. The spectral radius is often easy to compute, and it is a useful measure of the "size" or "strength" of a matrix. However, the spectral radius is not a vector-bound matrix norm; it is not compatible with any vector norm.

Simply stated, there is no vector norm for which it will be true, for all vectors x , that:

$$\|A * x\| \leq \|A\| * \|x\| \quad (3.27)$$

if the matrix norm used is the spectral radius.

To see this, consider a matrix whose dominant eigenvalue $\lambda > 0$ has algebraic multiplicity strictly greater than its geometric multiplicity. Then there must be an eigenvector x so that $A * x = \lambda * x$, but there is also a generalized eigenvector y ,

orthogonal to x , with the property that $A^*y = x + \lambda^*y$. Now, if the spectral radius is a vector-bound matrix norm, then it must be the case that

$$\|A^*y\| \leq |\lambda^*| \|y\| \quad (3.28)$$

but, since x is orthogonal to y , we can show that:

$$\|A^*y\| = \|x + \lambda^*y\| > |\lambda^*| \|y\| \quad (3.29)$$

Hence, the spectral radius is not a vector-bound matrix norm.

On the other hand, the value of the spectral radius is a lower bound for the value of any vector-bound matrix norm on A , because there must be an eigenvalue λ and a vector of unit norm x with the property that

$$A^*x = \lambda^*x \quad (3.30)$$

Thus, the norm of A^*x divided by the norm of x is λ . Therefore, the matrix norm of A must be at least $|\lambda|$.

The Euclidean norm of a real symmetric matrix is equal to its spectral radius (Press et al., 1986).

3.1.14 EISPACK Matrix Norm

The EISPACK matrix norm is used in the EISPACK eigenvalue package.

The definition of the norm for an M by N matrix is:

$$\|A\| = \sum_{I=1}^M \sum_{J=1}^N |A_{ij}| \quad (3.31)$$

This is a simple exercise to verify that this quantity satisfies the requirements for a matrix norm.

This norm is easy to calculate and was used in EISPACK in order to have a standard against which to compare the size of matrix elements that were being driven to zero (Press et al., 1986).

3.1.15 Householder and Givens Transformation Method

The actual calculations used to estimate the condition number using the Householder Method are based on the QL method, which was preceded by a reduction of a symmetric matrix to tridiagonal form. The Householder Method is used to reduce the matrix to the simple form in order for the iterative process to be implemented. The algorithm is applied to the symmetric matrix to transform it into the tridiagonal form. There are two approaches to this method: first is the Givens Method which is a modified Jacobi Method which instead of reducing the whole matrix to diagonal form reduces it to a tridiagonal form. This variation to the Jacobi method allows for carrying a finite number of operations unlike Jacobi method which requires iterations (Press et al., 1986).

A Givens rotation is a linear transformation applied to two vectors, or two rows or columns of a matrix, which can be interpreted as a coordinate axis rotation. The intent of the rotation is to zero out an entry of the vector or matrix using an orthogonal transformation. A Givens rotation is similar to the elementary row operation that adds a multiple of one row to another, but because a Givens rotation employs an orthogonal similarity transformation, it offers greater stability and ease of inversion (Press et al., 1986).

A Givens rotation matrix G has the form:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & (\cos(\theta))_{ii} & 0 & 0 & (\sin(\theta))_{ij} & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & (-\sin(\theta))_{ji} & 0 & 0 & (\cos(\theta))_{jj} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.32)$$

It is possible to zero out entries of a matrix, one by one, using Givens rotations, similar to the way that Householder matrices are used, to reduce a matrix to a simpler form. The process can be used to zero out the entire lower triangle of a matrix, but further operations on the upper triangle would reintroduce zeroes in the lower triangle. Nonetheless, zeroing out the lower triangle means that Givens rotations can be used to produce the QL factorization of the matrix.

On the other hand, the Householder method is as robust and stable as the Givens Method. The Householder method is superior to the Givens method due to its efficiency. There are experiments done with the Givens method which reduce the number of operations to the same as the Householder method and even avoid the necessity of taking

square roots which makes those methods competitive. However, this approach is still in the development phase and is not yet recommended to be used over the Householder reduction.

The Householder algorithm reduces a square matrix A of the dimensions N by N to tridiagonal form by $N-2$ orthogonal transformation. Householder matrices can be used to compute the QL factorization of a matrix. A Householder matrix can be found which will eliminate all the subdiagonal entries of the first column of the original matrix. Another Householder matrix can be found which will eliminate all the subdiagonal entries of the second column, and so on. At the end of $N-1$ steps of this process, the following series is computed which is the product of a Householder matrix:

$$H = H_{N-1} * H_{N-2} * \dots * H_2 * H_1 \quad (3.33)$$

the computed matrix is an orthogonal matrix and it is important to note that:

$$H * H^T = H * H = I \quad (3.34)$$

Therefore, by multiplying both sides by its transpose, which is also its inverse, one gets

$$A = H^T * L \quad (3.35)$$

or, if Q is defined to be H^T :

$$A = Q * L \quad (3.36)$$

where L is a lower triangular. The QL algorithm is preferred over the QR due to the significantly smaller round off error. Moreover, it is important to mention that the A matrix that is being used here is always a real, symmetric, and tridiagonal matrix. Those properties are guaranteed by the Householder algorithm.

3.2 QL Algorithm - Factorization of A

The QL factorization factors a matrix A into an orthogonal matrix Q and a lower triangular matrix L , so that:

$$A = Q * L \quad (3.37)$$

The factorization can also be applied to rectangular matrices, in which case one of the factors is no longer square.

The QL factorization can be useful for solving the full range of linear systems, whether they are nonsingular, under-determined, over-determined or ill conditioned. It can be used to carry out the Gram-Schmidt orthogonalization of a set of vectors constituting the columns of A . The QL factorization is also used repeatedly in an iterative solution of eigenvalue problems. This property was exploited in the code in order to calculate the condition numbers.

The QL factorization was achieved by producing incrementally, by a series of transformations utilizing Householder matrices. It is also possible to achieve the same results using Givens rotation matrices. However, due to the issues and factors discussed in Section 3.1.15, the Householder method was selected for this particular factorization (Press et al., 1986).

3.3 Singular Value Decomposition

The singular value decomposition of a rectangular M by N matrix A is a factorization of the form:

$$A = U * S * V^T \quad (3.38)$$

where; U is an M by N matrix with orthogonal columns; S is an N by N diagonal matrix, containing the non-negative singular values of A ; V is an N by N orthogonal matrix.

The solution of $A * x = b$ for non-square A can be found by seeking that x which minimizes the $\|A * x - b\|$. That x is equal to

$$V * S^{-1} * U^T \quad (3.39)$$

where, since S may have zeroes on its diagonal, S^{-1} is constructed by replacing each nonzero diagonal element by its inverse.

For any column I no greater than the minimum of M and N , let u_i be the I -th column of U , and v_i be the I -th column of V , and s_{ii} be the I -th diagonal element of S . Then it is a fact that

$$A * v_i = s_{ii} * u_i \quad (3.40)$$

and

$$A^T * u_i = s_{ii} * v_i \quad (3.41)$$

which leads to the conclusion that

$$A * A^T * u_i = s_{ii} * s_{ii} * u_i \quad (3.42)$$

and

$$A^* A^T * v_i = s_{ii} * s_{ii} * v_i \quad (3.43)$$

In other words, U , V and S contain information about the eigenvalues and eigenvectors of $A^* A^T$. Conversely, if the eigenvalues and eigenvectors of $A^* A^T$ are known, then the squares of the singular values of A , and the left singular vectors of A (the U matrix) are also known. The singular value decomposition can be used to construct the pseudoinverse of a rectangular or singular matrix A (Press et al., 1986).

3.4. Chapter Summary

Numerous methods of calculating condition numbers by using different techniques of calculating eigenvectors and eigenvalues were discussed above in order to illustrate the complexity and diversity of using condition numbers as a stability criterion. The simple change in the method of calculating the eigenvector might produce a different answer when computing condition numbers. In addition, there is more than one approach to calculate the condition number itself. This multitude of combinations produces a large number of possibilities to calculate the condition number where each calculated condition number will produce a different answer due to the variation in the manner of calculating the condition number itself and varying the way of calculating the eigenvalues and eigenvectors.

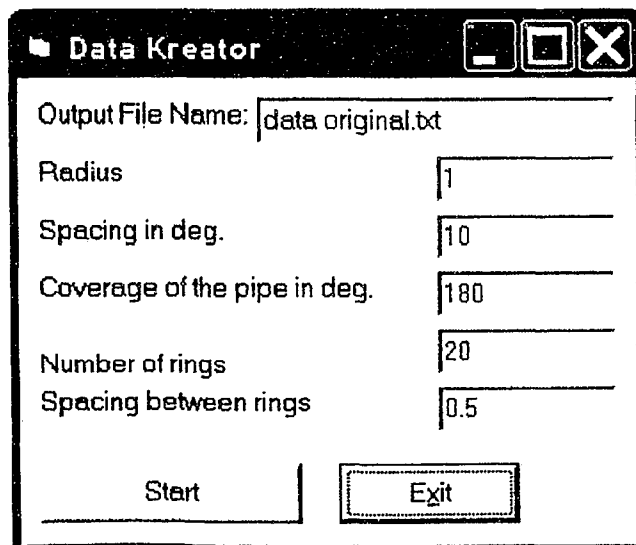
The situation is much less complicated when choosing calculations of the SVD (Singular Value Decomposition). The choice was made based mainly on the availability of the source code. The code was rewritten to the Visual Basic programming language and modified to fit into the scope of this thesis. However, the code and its intricacies are

not as well explained as condition numbers due to the lack of detailed information on the subject. This situation exists due to the problem of using “black box” approaches in programming and the concept of SVD being quite mature. Therefore, the SVD method was tested many times in the past and there is no need of proving the functionality and correctness of this method. On the other hand, a controversy still exists around the condition numbers, the best way to compute them, and their usefulness for setting stability criteria of the solutions of the matrices.

4.0 EXPERIMENTATION

The five parameters are tested to determine their critical values. The g parameter, which is the distance from the origin of the Cartesian system to the outer-shell of the solid, is tested first. The test is carried out by constructing a cylinder model. The cylinder is created by software (Data Kreator) designed to calculate the position of the points in the X-Y plane with the spacing between the points provided by the user in degrees on the circumference of the circle. Then the angle is decreased by half to simulate the LIDAR data, which is sparse on the sides and dense in the middle of the scanned object.

In order to control the coverage of the circle, an additional control box is introduced into the software. This option provides the user with the ability to control how much surface coverage there is on the cylinder, e.g., 90 degree coverage or preset of 180 degree coverage (i.e., half of the circle). The total number of points on a ring is about 1450 points if 180 degree coverage is selected.

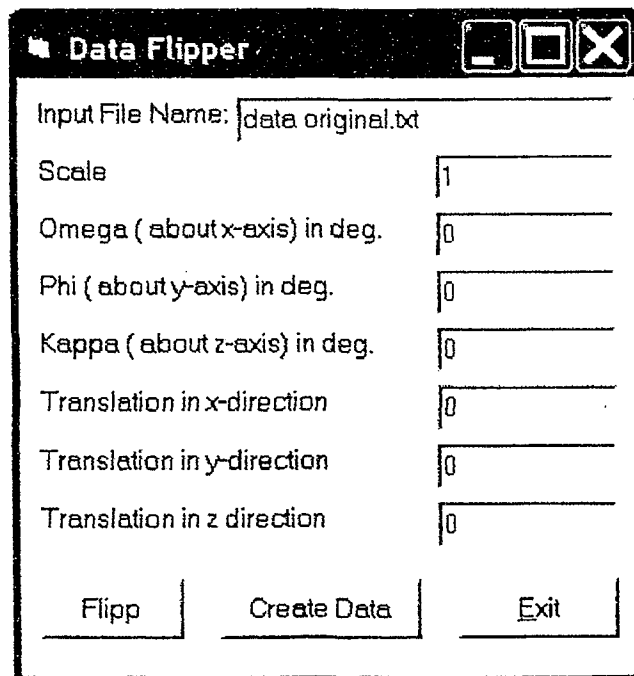
The image shows a screenshot of a software window titled "Data Kreator". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. Inside the window, there are several input fields and two buttons. The "Output File Name" field contains "data original.txt". The "Radius" field contains "1". The "Spacing in deg." field contains "10". The "Coverage of the pipe in deg." field contains "180". The "Number of rings" field contains "20". The "Spacing between rings" field contains "0.5". At the bottom of the window, there are two buttons: "Start" and "Exit".

Parameter	Value
Output File Name	data original.txt
Radius	1
Spacing in deg.	10
Coverage of the pipe in deg.	180
Number of rings	20
Spacing between rings	0.5

Figure 4.1: Preset for Data Kreator.

The user has the power to specify the radius of the circle as well as the number of rings he/she wants. The user can also define spacing between the rings. Then the ring is copied upward along the z-axis.

The second component of the software suite designed for testing is Data Flipper. Data Flipper allows seven degrees of freedom when manipulating the data. These include one scale, angle omega (ω), angle phi (ϕ), angle kappa (κ), translation in the x-direction, translation in the y-direction, and translation in the z-direction.



The screenshot shows a window titled "Data Flipper" with standard Windows window controls (minimize, maximize, close). The window contains the following fields and buttons:

Parameter	Value
Input File Name:	data original.txt
Scale	1
Omega (about x-axis) in deg.	0
Phi (about y-axis) in deg.	0
Kappa (about z-axis) in deg.	0
Translation in x-direction	0
Translation in y-direction	0
Translation in z direction	0

At the bottom of the window, there are three buttons: "Flipp", "Create Data", and "Exit".

Figure 4.2: Preset for Data Flipper.

The scale is the first option on the form, and the preset is scale equal to one. There are three rotations: one each about x-axis, y-axis and z-axis, respectively. Additionally, three translations in x, y and z directions, respectively, may be introduced. This set up allows for a fully controlled environment and data manipulation in 3D.

4.1 Testing the g parameter

The first parameter to be tested is g , assumed to be in centimeters, which is the shortest distance between the outer shell of the “extended” cylinder and the origin of the coordinate system. The “extended” cylinder can be defined as the cylinder projected to infinity in both axial directions. Data are created using the software Data Kreator with the following five preset parameters:

Radius = 1

Spacing = 10 degrees

Coverage of the pipe = 180 degrees

Number of rings = 20

Spacing between rings = 0.5.

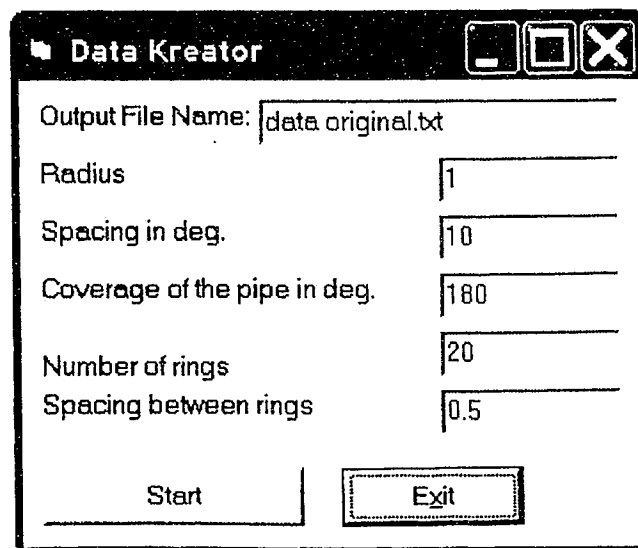


Figure 4.3: G-test-1, Data Kreator.

The cylinder that is wrapped around the z-axis is translated down along the z-axis to ensure that the center of the cylinder coincides with the origin of the coordinate system. The translation along the z-axis by -5 units is achieved by using the software Data Flipper. The remaining six parameters of the Data Flipper are treated as control values and were not changed during this test.

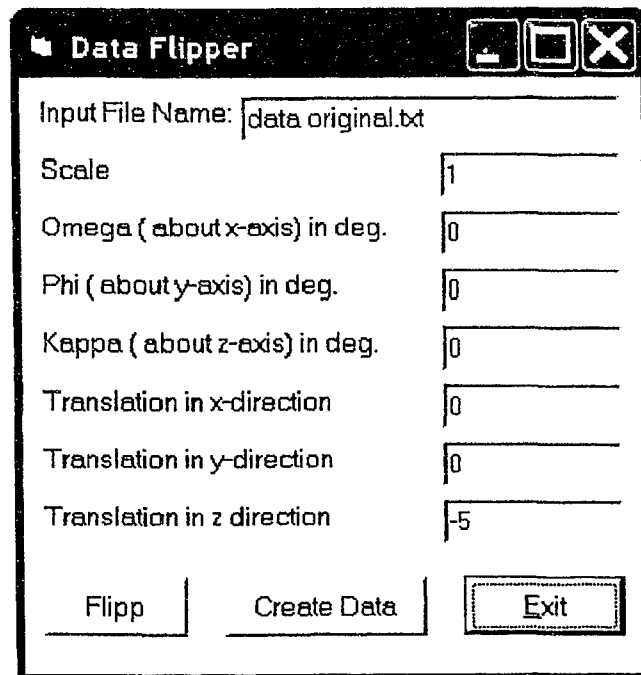


Figure 4.4: G-test-1 translation along z-axis by -5 units.

The data cloud used in this thesis can be found in Appendix A, (File name "g-test-1"). The data are translated down the z-axis by -5 units to align the centroid of the cylinder with the origin of the x, y, and z coordinate system. In order to ensure no further transformation, the following seven parameters are set to:

Scale = 1

Omega = 0 degrees

$\Phi = 0$ degrees

$\kappa = 0$ degrees

Translation in x-direction = 0

Translation in y-direction = 0

Translation in z-direction = -5.

The software “Point Cloud Data 3-D Adjuster and Analyser” is first used to depict the generated point cloud data from the file “g-test-1.txt”. The created data are displayed below:

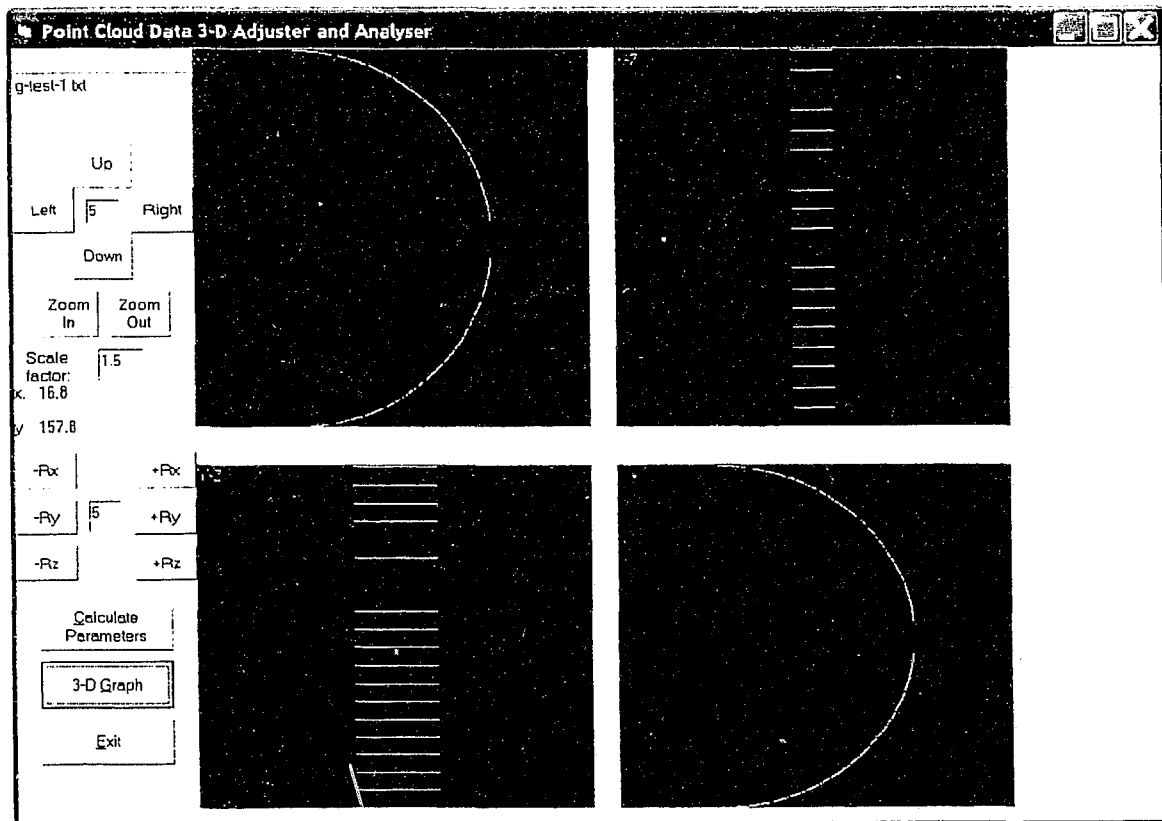


Figure 4.5: G-test-1 visualisation of data.

The next step is to adjust the data and calculate the geometric parameters of the cylinder. Furthermore, an analytical and numerical analysis is applied to establish the stability of the estimated parameters. This thesis concentrates on the development of the advanced stability criterion routines and testing the stability and thresholds of the calculations.

Two methods of calculating the condition numbers are compared to ensure the validity of calculated eigenvalues and condition numbers derived from them. The correlation-coefficient matrix is used to establish the strength of the relationship between the geometric parameters.

The following table illustrates the summary of values obtained from the analysis:

Expected Values of critical parameters:						
g	Angle_u_polar	Angle_v_polar	alpha * radto deg	1 / k		
-1	0	90	0.00E+00	1		
Converged answers:						
g	Angle_u_polar	Angle_v_polar	alpha * radto deg	1 / k		
-1	226.8545559	90	1.11E-15	1		
Square root of the diagonal terms ((diag i,i)^0.5):						
2.41E-02	6.29358E+14	8.36E-03	5.91E-02	2.22E-02		
Unsorted rearranged Eigenvalues for a real symmetric matrix using TRED2						
Eigenvalue	1	=	0			
Eigenvalue	2	=	14630.089			
Eigenvalue	3	=	14310.632			
Eigenvalue	4	=	1661.6976			
Eigenvalue	5	=	2111.9079			
Unsorted Eigenvalues for a real symmetric matrix using SVD						
Eigenvalue	1	=	14630.089			
Eigenvalue	2	=	14310.632			
Eigenvalue	3	=	1661.6976			
Eigenvalue	4	=	2111.9079			
Eigenvalue	5	=	1.90E-13			
CN# from TRED	CN# from SVD	Correlation-Coefficient Matrix				
1.46E+18	1	1	-7.13E-05	8.02E-02	1.52E-03	8.44E-02
1	1.022323083	-7.13E-05	1	-6.83E-02	0.9900026	5.90E-04
1.022323083	8.804302916	8.02E-02	-6.83E-02	1	-6.91E-02	4.04E-03
8.804302916	6.927427651	1.52E-03	0.9900026	-6.91E-02	1	9.85E-03
6.927427651	7.71E+16	8.44E-02	5.90E-04	4.04E-03	9.85E-03	1

Table 1: G-test-1 Summary Table

The pre-defined value of g is identical to the estimated value as well as the value for $1/k$, which is the radius of the cylinder. The marginal difference between the expected and converged answer in the $alpha$, (α), angle. This can be explained by the fact that a finite number of digits is used. In addition, truncation and round off error may influence this particular quantity. The remaining parameters, which are the most important ones, match the expected values. The μ angle has an extremely large calculation error of approximately 227 degrees, which could be explained by the very

poorly defined angle μ in the x-y plane. The μ angle is an angle in the x-y plane. In this test, the center line of the cylinder coincides with the z-axis. This setup creates the problem of a numerically unstable angle, since the x and y components of this angle are calculated based on residuals and the lengths of x and y components are zero. On the other hand, the ϑ angle is well defined. Its numerical value of 90 degrees corresponds precisely with the converged answer. In addition, the critical parameters have corresponding standard deviation values for each calculated critical parameter of g , μ , ϑ , α , and k . Thus, the only value with an extremely high standard deviation is the μ angle due to extremely poor geometry for this particular data set. The high value of the condition number suggests that the solution is unstable. This is true, since the cylinder is wrapped around the z-axis and the μ angle is highly unstable.

In order to further investigate the stability of the software, especially the g parameter, the data are moved down by 5 units as in the last instance and then moved away from the origin of the coordinate system by 100 units along the x-axis.

Data Flipper

Input File Name: data original.txt

Scale: 1

Omega (about x-axis) in deg.: 0

Phi (about y-axis) in deg.: 0

Kappa (about z-axis) in deg.: 0

Translation in x-direction: 100

Translation in y-direction: 0

Translation in z direction: -5

Flipp Create Data Exit

Figure 4.6: G-test-2, Data Flipper.

This approach enables one to predict the value of the g parameter. The output file is renamed to g-test-2.txt (see Appendix A).

Expected Values of critical parameters:						
<i>g</i>	<i>Angle_u_polar</i>	<i>Angle_v_polar</i>	<i>alpha * radtod</i>	<i>1 / k</i>		
99	0	90	0	1		
Converged answers:						
<i>g</i>	<i>Angle_u_polar</i>	<i>Angle_v_polar</i>	<i>alpha * radtod</i>	<i>1 / k</i>		
99	7.81E-15	90	1.16E-15	1		
Square root of the diagonal terms ((diag i.i)^0.5):						
0.191809238	1.71E-03	8.30E-03	8.39E-03	3.92E-02		
Unsorted Eigenvalues for a real symmetric matrix using TRED2						
Eigenvalue	1	=	13662.1913			
Eigenvalue	2	=	18050137.8			
Eigenvalue	3	=	26.8921286			
Eigenvalue	4	=	828.92372			
Eigenvalue	5	=	740880.86			
Unsorted Eigenvalues for a real symmetric matrix using SVD						
Eigenvalue	1	=	740880.86			
Eigenvalue	2	=	18050137.8			
Eigenvalue	3	=	26.8921286			
Eigenvalue	4	=	828.92372			
Eigenvalue	5	=	13662.1913			
CN# from TRED	CN# from SVD	Correlation-Coefficient Matrix				
1321.174428	24.36307743	1	1.55E-14	0.876991	-1.58E-14	0.465048
1	1	1.55E-14	1	1.80E-14	-0.990113	-1.30E-15
671205.2447	671205.2447	0.87699063	1.80E-14	1	-1.81E-14	1.17E-03
21775.39056	21775.39056	-1.58E-14	-0.99011284	-1.81E-14	1	9.10E-17
24.36307743	1321.174428	0.465048395	-1.30E-15	1.17E-03	9.10E-17	1

Table 2: G-test-2 Summary Table

The pre-defined value of g is identical to the estimated value as well as the value for $1/k$, which is the radius of the cylinder. The marginal difference between the expected and the converged answer in the α angle and μ angle occurs because a finite number of digits is used. As in the previous test, truncation and round off error may influence this particular quantity. The remaining parameters, i.e., the most important ones, match the expected values. The μ angle is an angle in the x-y plane. In this test, the center line of the cylinder coincides with the x-axis due to translation of 100

cm in the x-direction. On the other hand, the ϑ angle is well defined and its numerical value of 90 degrees corresponds precisely with the estimated answer. In addition, the critical parameters have corresponding standard deviation values for each calculated critical parameter of g , μ , ϑ , α , and k . It is important to emphasize a high correlation of 0.876991 (where 1 is the highest possible correlation, 0 is the lowest) between the g parameter and angle ϑ . The g parameter is the shortest distance between the origin and the outer shell of the cylinder. Angle ϑ is the zenith angle. There is also a notably high correlation of 0.465045 between the g parameter and k parameter (where k is the inverse of the radius). Moreover, there is a trend of the μ angle being inversely correlated to the ϑ angle by - 0.990113 (where -1 is highest inversely correlated value and 0 is the lowest). The value of the condition number is considerably smaller, since the cylinder was moved away from the origin. This translation allowed the g parameter and other free parameters to be well defined, with the exception of the μ angle. The improvement in the geometry of the problem yields a smaller condition number. Thus, it is clear that the condition number can be used as a measure of stability of equations including the geometry of a cylinder.

The final test for g compliance is test-3, where the data are moved away from the origin by 1 km. This test indicates that the software is able to handle the difference in scale of 1:100000. However, the software is unable to maintain the stability of the algorithms at great distances. The cylinder is created in the same manner as in two previous tests, with the difference in the translations. The cylinder is moved by -5 units in the z-direction as before and translated in the y-direction by 100,000 units.

Parameter	Value
Input File Name:	data original.txt
Scale	1
Omega (about x-axis) in deg.	0
Phi (about y-axis) in deg.	0
Kappa (about z-axis) in deg.	0
Translation in x-direction	0
Translation in y-direction	100000
Translation in z direction	-5

Buttons: Flipp, Create Data, Exit

Figure 4.7: G-test-3, Data Flipper.

The values for translation were selected based on easily predictable solutions to the parameters. The table below depicts the output of this test:

Expected Values of critical parameters:						
<i>g</i>	<i>Angle_u_polar</i>	<i>Angle_v_p</i>	<i>alpha * rad</i>	<i>1/k</i>		
99999	90	90	0	1		
Converged answers:						
<i>g</i>	<i>Angle_u_polar</i>	<i>Angle_v_p</i>	<i>alpha * rad</i>	<i>1/k</i>		
99999.05	89.94270424	90	-1.89E-07	0.9999994		
Square root of the μ^2 :						
0.381923828	3.80E-06	8.39E-03	8.30E-03	3.92E-02		
Unsorted Eigenvalues for a real symmetric matrix using TRED2						
Eigenvalue	1 =		14509.383			
Eigenvalue	2 =		6.8515938			
Eigenvalue	3 =		658.73656			
Eigenvalue	4 =		3562663.1			
Eigenvalue	5 =		1.746E+13			
Unsorted Eigenvalues for a real symmetric matrix using SVD						
Eigenvalue	1 =		1.746E+13			
Eigenvalue	2 =		3562663.1			
Eigenvalue	3 =		6.8515944			
Eigenvalue	4 =		658.74182			
Eigenvalue	5 =		14509.381			
CN# from TRED CN# from SVD Correlation-Coefficient Matrix						
1203129883	1	1	-1.35E-02	0.9927265	9.93E-05	0.102669
2.54783E+12	4899894.339	-1.35E-02	1	6.07E-05	-0.989345	-0.13239
26500233387	2.54783E+12	0.992726	6.07E-05	1	-2.01E-05	1.18E-06
4899894.339	26500021867	9.93E-05	-0.989345	-2.01E-05	1	1.15E-03
1	1203130069	0.102669	-0.132386	1.18E-06	1.15E-03	1

Table 3: G-test-3 Summary Table

The pre-defined value of g has an error of 5 mm in a distance of 100 km to the estimated value. The value for $1/k$, which is the radius of the cylinder, has a negligibly small error. This test demonstrates that the software has an ability to correctly calculate the distance and the radius of the cylinder with 5 mm accuracy from 1 km. This test assumes that the data are error free. The marginal difference between the expected and converged answer in the α angle and the μ angle may be explained, as before,

by the fact that a finite number of digits is used. In addition, truncation and round off error may influence this particular quantity. The remaining parameters, i.e., the most important ones, match the expected values. The μ angle has a value of 90 degrees in the x-y plane due to the translation in the y-direction of 100,000 cm. As in all tests, the ϑ angle is very well defined and its numerical value of 90 degrees corresponds precisely to the converged answer. In addition, the critical parameters have corresponding standard deviation values for each calculated critical parameter of g , μ , ϑ , α , and k . The standard deviation tables are suggesting that the angles are very strongly defined with the increase of the distance. However, the g parameter and the radius are decreasing in precision. On the other hand, g and k become de-correlated. The large increase in distance increases the value of condition number. The increase in the condition number is not as dramatical as it is when the cylinder is wrapped around the z-axis. This suggests that the distortion of the geometry of the cylinder is not as significant as when the cylinder coincides with the origin or one of the axes.

It is important to emphasize a high correlation of 0.9927265 between the g parameter (i.e., the shortest distance between the origin and the outer shell of the cylinder) and angle μ (i.e., the X-Y plane angle). There is also a high inverse correlation of -0.989345 between the μ parameter and the α parameter. The μ parameter and α parameter have a large negative correlation due to the normal pointing in the z-direction. This creates a singularity situation and the μ angle is not solved properly.

In summary, the g parameter influences the statistical analysis rather strongly. When increased, the g parameter allows the angle to be strongly defined, but at the same

time it minimizes its own precision. Overall, the increase of the g parameter results in a positive effect on the algorithms. As the angle definitions are strengthened, the g parameter loses its precision, but only by 1 ppm. Moreover, the radius accuracy is not influenced. By increasing the g parameter, the correlation between g and $1/k$ is minimized. Understanding the role of the g parameter in numerical analysis is critical, since this parameter significantly influences test values. The condition numbers are affected in a not very significant way by increase in g parameter. The G-test-1 test has the highest condition number due to poor geometry. The condition number is significantly improved as the cylinder is moved away from the origin. Finally, the condition number is increased as the cylinder is translated by great distance. Therefore, the condition number suggests a stable solution.

4.2 Testing the Influence of the Coverage of the Pipe

The following tests are completed to determine how the point cloud coverage of the pipe would influence the five parameters. This is of particular importance, since this reduced coverage is a commonly encountered problem with LIDAR data. In order to keep the standard deviations and stability constant, the tests will be performed using data with the center of gravity at the origin of the coordinate system.

The first test is completed with the pipe coverage of 150 degrees which is 43% coverage. Data are created using Data Kreator:

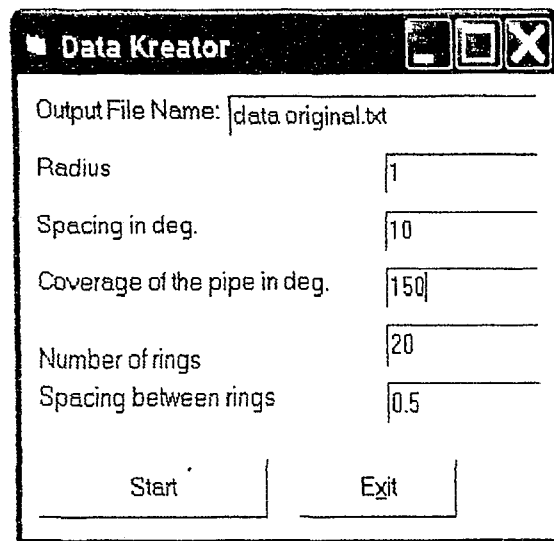


Figure 4.8: Coverage-150 degrees, Data Kreator.

As in previous tests, the data are created using the radius of 1 unit, 10-degree spacing, and 20 rings with the spacing between the rings of 0.5 units. The coverage of the pipe is reduced to 150 degrees in order to determine how this will affect the standard deviations of the estimated parameters and the stability criteria of the algorithms.

No further changes are applied. Thus, the data are centered around the z-axis and the data's center of gravity is located at the center of the coordinate system.

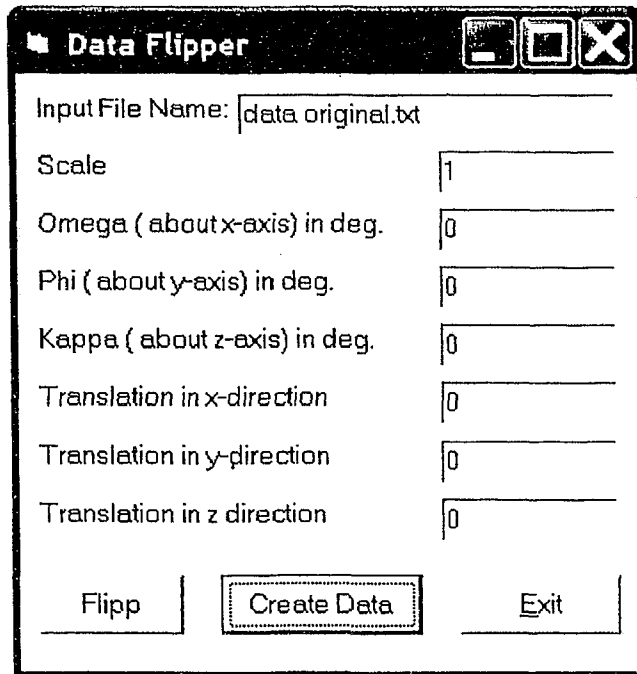


Figure 4.9: Coverage-150 degrees, Data Flipper

The depicted data are the result of this simulation. The 150 degree coverage of the pipe in the x-y window is quite noticeable. There is also a recognizable narrow pattern in the x-z projection. This pattern represents one of the sides of the cylinder being reduced when compared to the y-z projection:

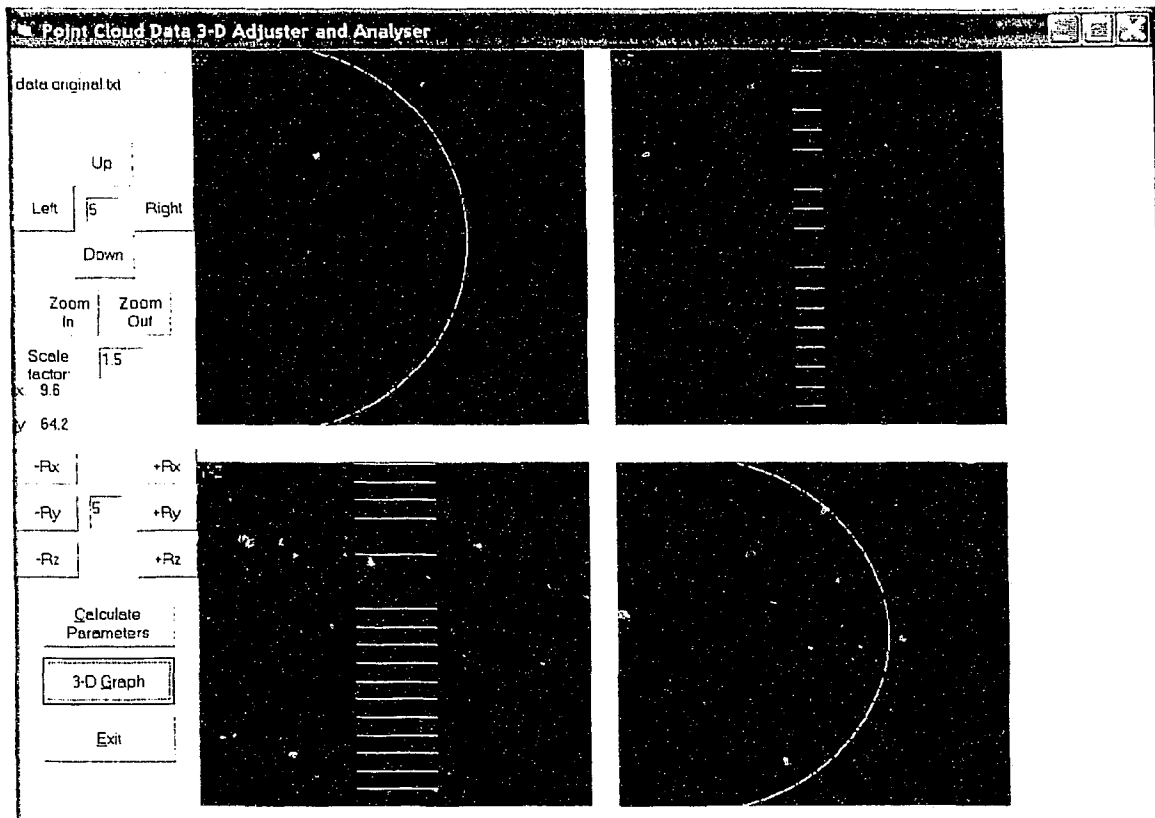


Figure 4.10: Coverage-150 degrees, Data Visualisation

The depicted data have been processed in the software and the following chart was produced based on the summary of the numerical analysis:

Converged answers:						
<i>g</i>	<i>Angle_u_polar</i>	<i>Angle_v_pd</i>	<i>alpha * rad</i>	<i>1 / k</i>		
-1	40.70778983	90	180	1		
Square root of the diagonal terms ((diag i,j)*0.5):						
5.07E-02	1.27722E+14	6.46E-02	2.29E-02	6.32E-02		
Unsorted Eigenvalues for a real symmetric matrix using TRED2						
Eigenvalue	1	=	0			
Eigenvalue	2	=	469.7096			
Eigenvalue	3	=	243.7539			
Eigenvalue	4	=	36432.86			
Eigenvalue	5	=	54846.48			
Unsorted Eigenvalues for a real symmetric matrix using SVD						
Eigenvalue	1	=	54846.48			
Eigenvalue	2	=	36432.86			
Eigenvalue	3	=	469.7096			
Eigenvalue	4	=	243.7539			
Eigenvalue	5	=	3.80E-13			
CN# from TRED CN# from SVD Correlation-Coefficient Matrix						
5.48E+18	1	1	2.23E-02	-9.04E-02	-1.32E-02	-0.27529
116.7667836	1.505412393	2.23E-02	1	0.9909461	0.9661954	-0.3153
225.0075962	116.7667836	-9.04E-02	0.990946	1	0.9595428	-0.30728
1.505412393	225.0075962	-1.32E-02	0.966195	0.9595428	1	-0.18809
1	1.44E+17	-0.275289	-0.3153	-0.307279	-0.188092	1

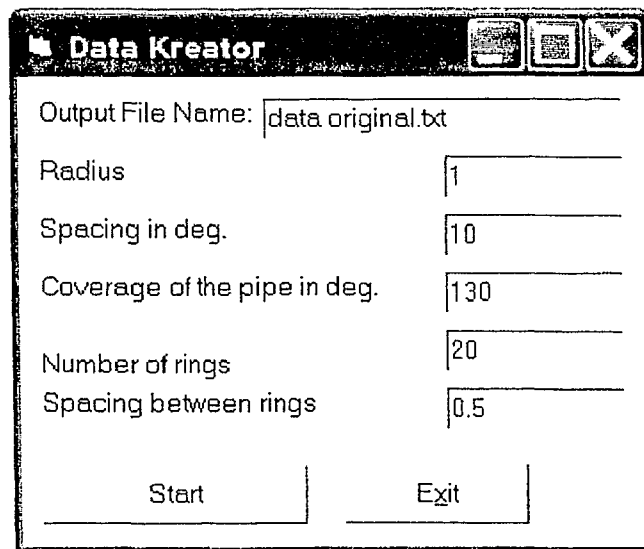
Table 4: Coverage-150 degrees, Summary table

The value of *g* is computed correctly without any errors. The value for $1/k$, which is the radius of the cylinder, has no errors as well. The difference between the expected and the estimated answer in the *alpha* (α) angle and the μ angle can be explained, as before, by the fact that the *alpha* (α) angle is unable to distinguish which part of the pipe is the beginning and which is the end of the pipe. Moreover, the μ angle is calculated based on small errors in the residuals, as explained previously.

As in all instances, the ϑ angle is very well defined and its numerical value of 90 degrees corresponds precisely to the expected answer. In addition, the critical parameters

have corresponding standard deviation values for each calculated critical parameter of g , μ , ϑ , α , and k . The standard deviation tables indicate that the angles are strongly defined with the exception of the μ angle. In contrast to previous tests, this is a new trend of angles being highly correlated. The condition number is not influenced by changing coverage from 180 degrees to 150.

The next test is completed to determine the effect of decreased coverage of the pipe on the outcome of the numerical analysis. The following test is completed in 130-degree coverage with the same parameters as in the previous test.



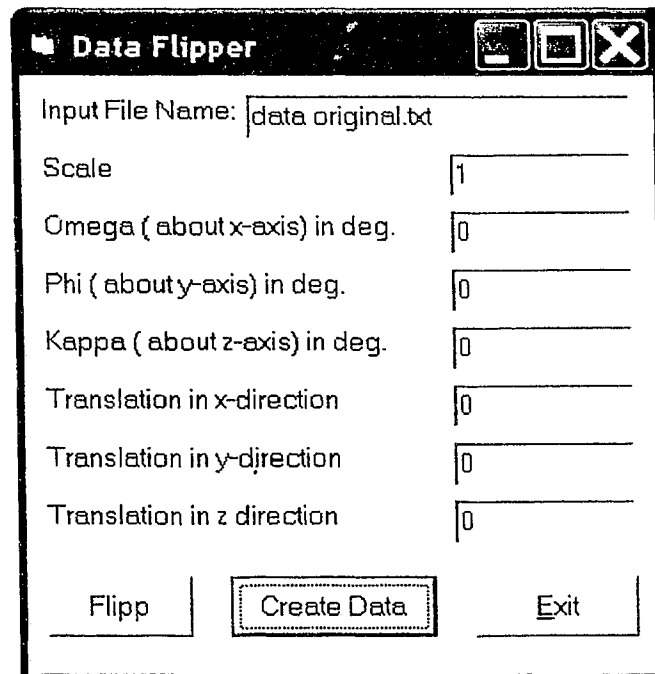
The screenshot shows a window titled "Data Kreator" with standard Windows window controls (minimize, maximize, close). The window contains several input fields and two buttons at the bottom.

Parameter	Value
Output File Name:	data original.txt
Radius	1
Spacing in deg.	10
Coverage of the pipe in deg.	130
Number of rings	20
Spacing between rings	0.5

At the bottom of the window, there are two buttons: "Start" and "Exit".

Figure 4.11: Coverage-130 degrees, Data Kreator

In order to better control the environment, Data Flipper is used with the scale set to one while the other parameters are zero.



Data Flipper

Input File Name: data original.bt

Scale 1

Omega (about x-axis) in deg. 0

Phi (about y-axis) in deg. 0

Kappa (about z-axis) in deg. 0

Translation in x-direction 0

Translation in y-direction 0

Translation in z direction 0

Flipp Create Data Exit

Figure 4.12: Coverage-130 degrees, Data Flipper.

Data are depicted in the figure below. The change in the coverage of the pipe is noticeable in the x-y window.

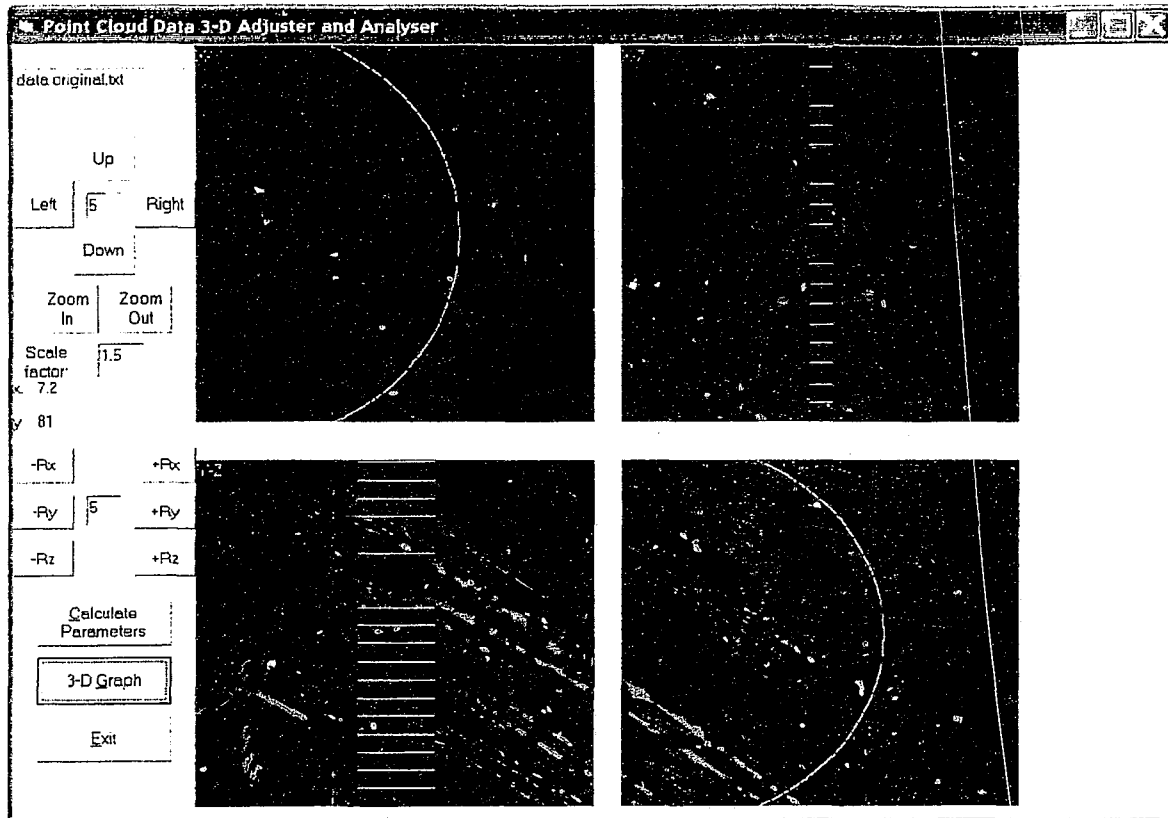


Figure 4.13: Coverage-130 degrees, Data Visualization

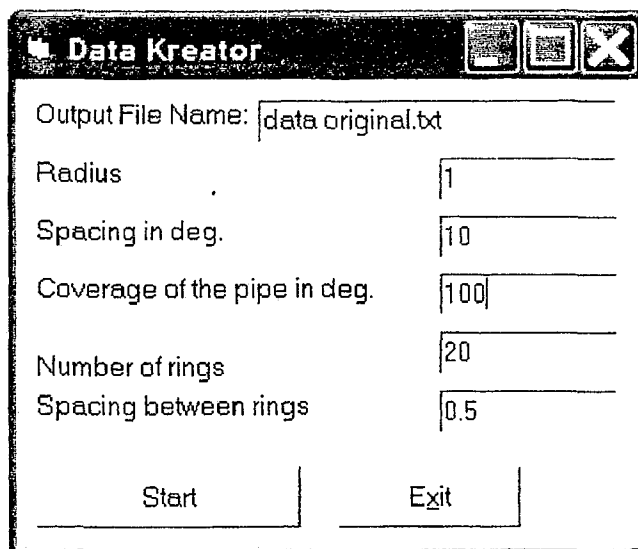
The Point Cloud Data 3-D Adjuster and Analyser is run. The output files are combined into one summary chart, which is depicted below. The input and output files can be found in Appendix A, in the folder entitled “130 deg coverage”.

Converged answers:						
g	Angle_u_polar	Angle_v_pol	alpha * radto	1/k		
-1	-124.9604728	90	180	1		
Square root of the diagonal terms ((diag I.I)^0.5):						
5.90E-02	3.16E-05	1.10E-02	8.83E-03	4.80E-02		
Unsorted Eigenvalues for a real symmetric matrix using TRED2						
Eigenvalue	1	=	-2.33E-26			
Eigenvalue	2	=	280.4094893			
Eigenvalue	3	=	423.3083807			
Eigenvalue	4	=	26640.54901			
Eigenvalue	5	=	53021.17753			
Unsorted Eigenvalues for a real symmetric matrix using SVD						
Eigenvalue	1	=	53021.17753			
Eigenvalue	2	=	26640.54901			
Eigenvalue	3	=	280.4094893			
Eigenvalue	4	=	423.3083807			
Eigenvalue	5	=	7.62E-13			
CN# from TRED CN# from SVD Correlation-Coefficient Matrix						
-2.27E+30	1	1	-1.04E-20	-0.828559	5.43E-03	-6.63E-03
189.084819	1.990243426	-1.04E-20	1	-1.17E-19	1.81E-19	-6.03E-20
125.254259	189.084819	-0.8285591	-1.17E-19	1	-0.2733882	0.21790036
1.990243426	125.254259	5.43E-03	1.81E-19	-0.273388	1	-0.8192815
1	6.96E+16	-6.63E-03	-6.03E-20	0.2179004	-0.8192815	1

Table 5: Coverage-130 degrees, Data Summary

As in previous examples, the parameters are calculated correctly with the exception of the angle in the x-y plane. It is important to emphasize that this is a singular example. When the ϑ -angle is 90 degrees and the cylinder is wrapped around the z axis, the μ -angle is calculated based on calculation errors. The remaining parameters are well-defined based on the standard deviations. The values are recovered with the standard deviation in millimeters for linear parameters and hundredth of a degree for angular parameters. The correlation-coefficient matrix suggests strong inversely correlated g and μ angle parameters. The condition number increased in value which suggests that the reduction in coverage have impacted the solution.

The data are further reduced down to 100-degree coverage to further investigate the influence of the coverage on the outcome of the statistical analysis.



Parameter	Value
Output File Name	data original.txt
Radius	1
Spacing in deg.	10
Coverage of the pipe in deg.	100
Number of rings	20
Spacing between rings	0.5

Start Exit

Figure 4.14: Coverage-100 degrees, Data Kreator.

There is no rotation or translation introduced to the data in order to keep them consistent with previous tests.

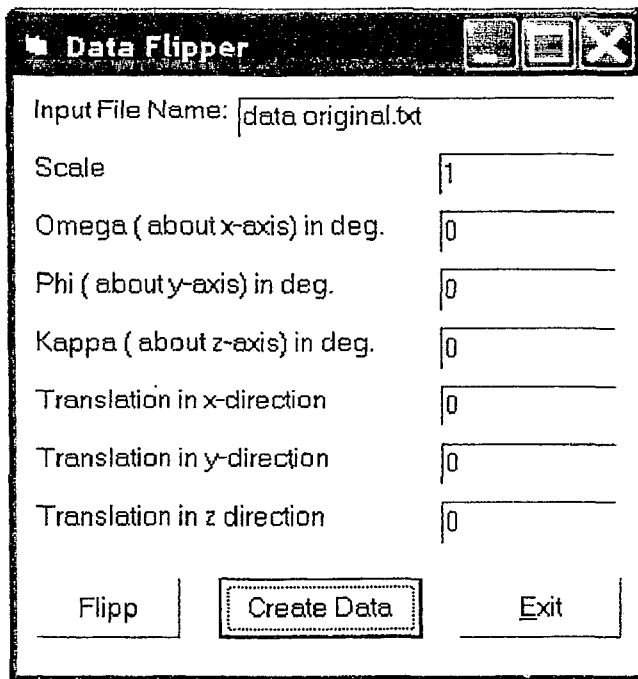


Figure 4.15: Coverage-100 degrees, Data Flipper.

The data are displayed in three projections for easy comprehension and data visualization. When observing the data, it is noticeable that the coverage is decreased to 100 degrees in the x-y window as well in the x-z window. The data in the x-z window are a projection of the pipe from the side view. The y-z projection allows for visualization of the frontal side of the pipe.

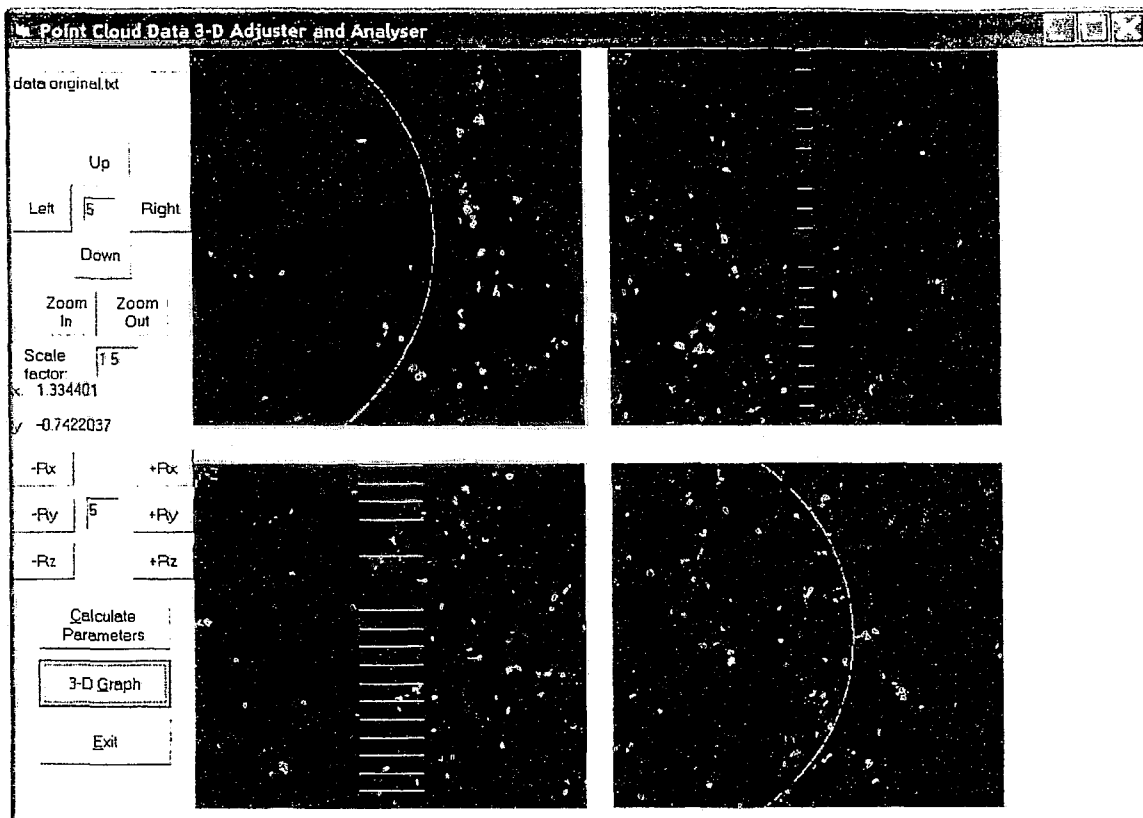


Figure 4.16: Coverage-100 degrees, Data Visualization.

The following compiled chart presents the numerical summary for the data set described above:

<i>Converged answers:</i>						
<i>g</i>	<i>Angle_u_polar</i>	<i>Angle_v_p</i>	<i>alpha * radts</i>	<i>1/k</i>		
-1	330.9218981	90	180	1		
<i>Square root of the diagonal terms ((diag I,i)^0.5):</i>						
0.437186101	2.49533E+13	1.50E-02	1.42E-02	0.206844		
<i>Unsorted Eigenvalues for a real symmetric matrix using TRED2</i>						
<i>Eigenvalue</i>	1	=	-1.37E-26			
<i>Eigenvalue</i>	2	=	5.4495954			
<i>Eigenvalue</i>	3	=	1451.8416			
<i>Eigenvalue</i>	4	=	7874.76			
<i>Eigenvalue</i>	5	=	30884.099			
<i>Unsorted Eigenvalues for a real symmetric matrix using SVD</i>						
<i>Eigenvalue</i>	1	=	30884.099			
<i>Eigenvalue</i>	2	=	5.4495954			
<i>Eigenvalue</i>	3	=	1451.8416			
<i>Eigenvalue</i>	4	=	7874.76			
<i>Eigenvalue</i>	5	=	1.74E-14			
<i>CN# from TRED CN# from SVD Correlation-Coefficient Matrix</i>						
-2.25E+30	1	1	-0.4562544	0.154464	-0.39538	0.991054
5667.227917	5667.227917	-0.45625	1	-0.59769	-0.21085	-0.48628
21.27236157	21.27236157	0.154464	-0.5976903	1	0.140306	0.261401
3.921909921	3.921909921	-0.39538	-0.2108521	0.140306	1	-0.39474
1	1.78E+18	0.991054	-0.4862831	0.261401	-0.39474	1

Table 6: Coverage-100 degrees, Data Summary.

The summary chart illustrates a strong trend of decreasing precision of the *g* parameter. There is also a similar trend in the definition of the radius of the cylinder. Moreover, the correlation-coefficient matrix demonstrates that all parameters are becoming positively correlated or inversely correlated. The condition numbers show highly unstable equations, which is a continuation of the trend from the previous test. Thus, the prediction of a highly degradable precision of the calculations is demonstrated. Further testing will demonstrate this trend more dramatically.

The next step is to reduce the coverage of the pipe even further and to determine whether this will influence the five parameters as well as the stability criterion. In order to maintain the standard deviations and the stability constant, the tests will be completed using data with the center of gravity at the origin of the coordinate system. The test is carried out with the pipe coverage of 70 degrees. Once again, the data are created using “Data Kreator”.

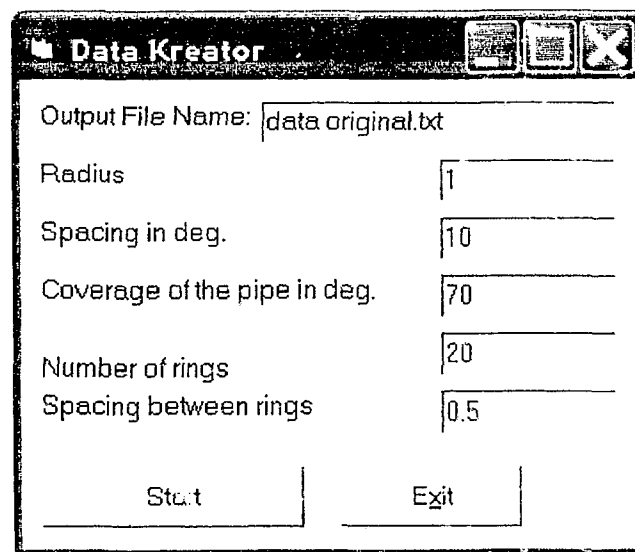


Figure 4.17: Coverage-70 degrees, Data Kreator.

As in previous examples, the data are created using the radius of 1 unit and 10-degree spacing. There are 20 rings with the spacing between the rings of 0.5 unit. The coverage of the pipe is reduced to 70 degrees in order to determine how it will affect standard deviations and the algorithms' stability. No further changes are applied and the data are centered around the z-axis. The data's center of gravity is located at the center of the coordinate system.

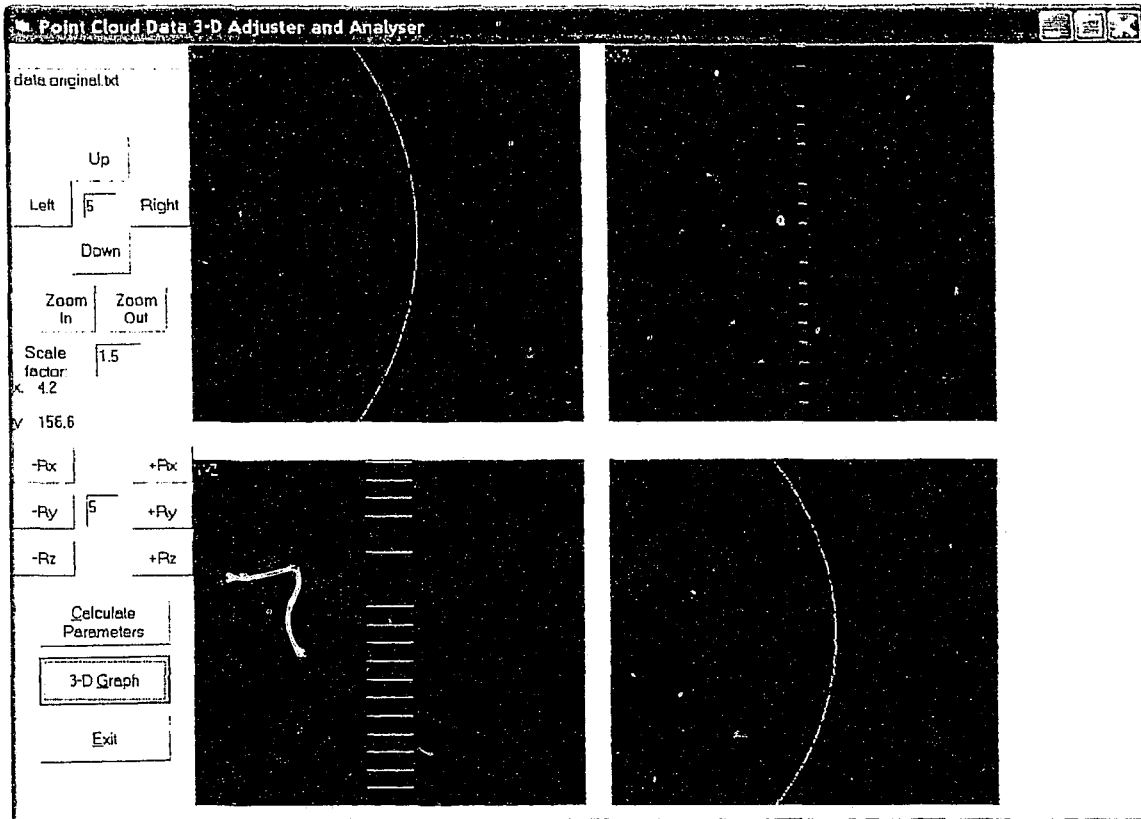


Figure 4.18: Coverage-70 degrees, Data Visualization.

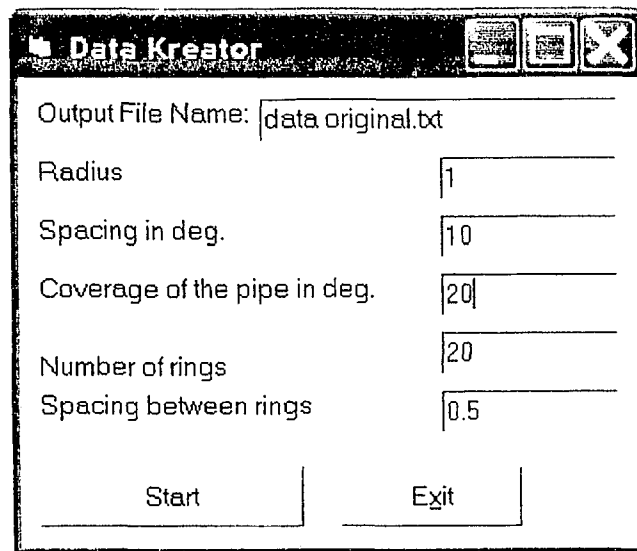
The chart below represents the statistical summary of the calculated data.

Converged answers:						
<i>g</i>	<i>Angle_u_polar</i>	<i>Angle_v_pola</i>	<i>alpha * radiu</i>	<i>1 / k</i>		
-1	1.72E-09	90	180	1		
Square root of the diagonal terms ((diag I.)^0.5):						
0.983584541	3.16E-05	1.05E-02	1.44E-02	0.4750978		
Unsorted Eigenvalues for a real symmetric matrix using TRED2						
Eigenvalue	1 =		-8.69E-17			
Eigenvalue	2 =		4840.76439			
Eigenvalue	3 =		0.83858263			
Eigenvalue	4 =		1343.50747			
Eigenvalue	5 =		41938.0166			
Unsorted Eigenvalues for a real symmetric matrix using SVD						
Eigenvalue	1 =		41938.0166			
Eigenvalue	2 =		1343.50747			
Eigenvalue	3 =		0.83858263			
Eigenvalue	4 =		4840.76439			
Eigenvalue	5 =		1.15E-14			
CN# from TRED CN# from SVD Correlation-Coefficient Matrix						
-4.82E+20	1	1	-6.74E-22	-8.83E-02	1.61E-10	0.998198
8.663511204	31.2153207	-6.74E-22	1	-9.72E-19	-8.38E-20	-6.81E-22
50010.59535	50010.59535	-8.83E-02	-9.72E-19	1	-4.17E-11	-3.63E-02
31.2153207	8.663511204	1.61E-10	-8.38E-20	-4.17E-11	1	1.62E-10
1	3.65E+18	0.99819804	-6.81E-22	-3.63E-02	1.62E-10	1

Table 7: Coverage-70 degrees, Data Summary

There is clearly visible the degradation of the parameters in terms of the precision. There is almost now a degradation of 100% of a *g* parameter, and 47% of radius. However the correlation-coefficient matrix shows that the parameters are not correlated. The exception is the *g* parameter and radius. Moreover, the stability criterion suggests that the equation is more stable in this test than in previous one. The stability is almost as good as in the 150 degree coverage test. This suggests that stability criteria can not be used alone. It is obvious that stability criteria become stable since the error is so great that even a large change in errors will not have much influence on the poorly defined parameters.

The last test presented is the 20-degree coverage test. This test demonstrates that data without errors can be calculated even with extremely low coverage. The user should be aware that this software would calculate the parameters even though they may not be correct or well defined. The numerical summary is designed for the user to verify the answer. The 20-degree coverage test was the lowest coverage that the software was able to complete. This software is unable to process data with less than 20-degree coverage.



The screenshot shows a window titled "Data Kreator" with a standard Windows-style title bar (minimize, maximize, close buttons). Inside the window, there is a text input field for "Output File Name:" containing the text "data.original.txt". Below this, there are five rows of labels and input fields:

Parameter	Value
Radius	1
Spacing in deg.	10
Coverage of the pipe in deg.	20
Number of rings	20
Spacing between rings	0.5

At the bottom of the window, there are two buttons: "Start" and "Exit".

Figure 4.19: Coverage-20 degrees, Data Kreator.

The data are depicted below. There is a clearly visible arch in x-y window. However, the data in the x-z window resemble a line instead of a rectangle.

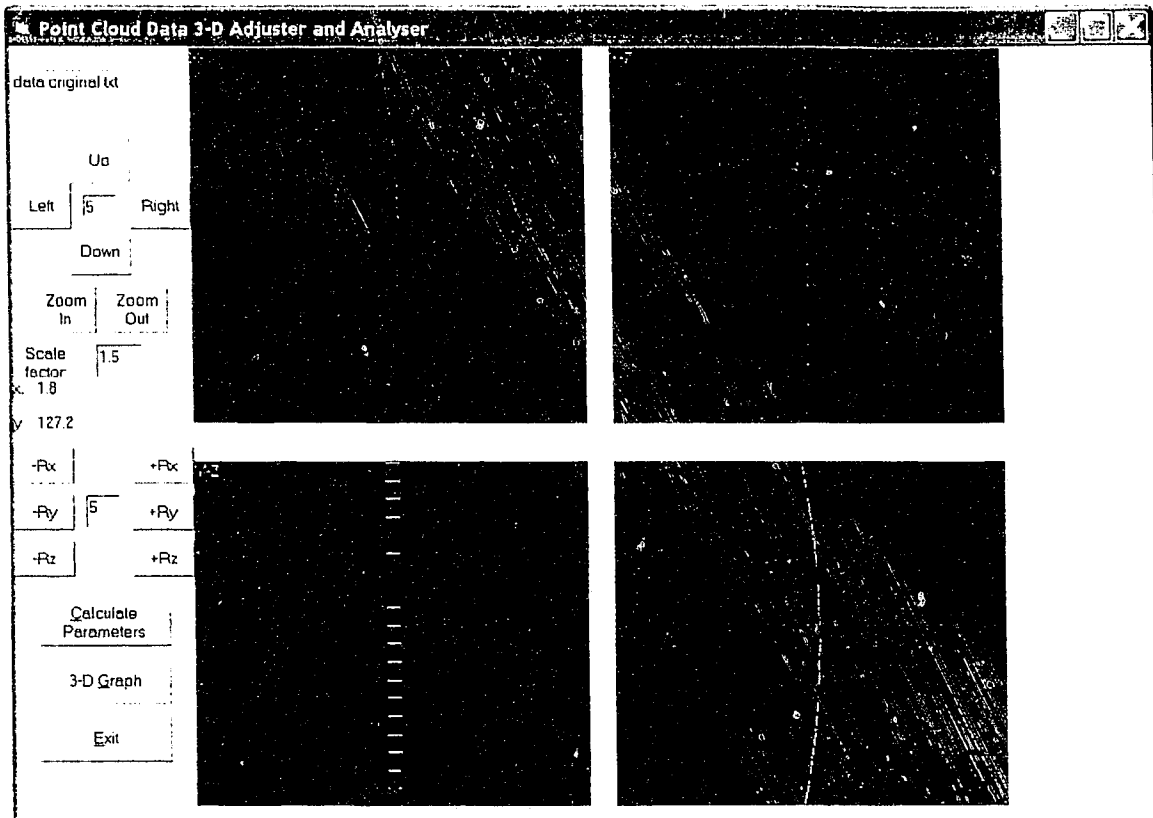


Figure 4.20: Coverage-20 degrees, Data Visualization.

The summary is a good example of how the results may look when there is no random error introduced to the data. The parameters are recovered very well. However, the estimates are showing an extremely poor definition of those parameters.

Converged answers:						
<i>g</i>	Angle_u_polar	Angle_v_p	alpha * radto	1 / k		
-1	-398.5359506	90	9.66E-14	1		
Square root of the θ^2 :						
14.30038906	1.37E+15	0.957415	4.1026742	8.797347		
Unsorted Eigenvalues for a real symmetric matrix using TRED2						
Eigenvalue	1	=	-1.60E-29			
Eigenvalue	2	=	5.10E-02			
Eigenvalue	3	=	9.0545105			
Eigenvalue	4	=	51.552019			
Eigenvalue	5	=	1431.8757			
Unsorted Eigenvalues for a real symmetric matrix using SVD						
Eigenvalue	1	=	1431.8757			
Eigenvalue	2	=	51.552019			
Eigenvalue	3	=	5.10E-02			
Eigenvalue	4	=	9.0545105			
Eigenvalue	5	=	3.41E-15			
CN# from TRED CN# from SVD Correlation-Coefficient Matrix						
-8.93E+31	1	1	-0.9577253	-0.86016	-0.98044	0.992153
28084.26171	27.77535555	-0.95773	1	0.706229	0.994393	-0.98587
158.139489	28084.26171	-0.86016	0.7062286	1	0.775387	-0.80034
27.77535555	158.139489	-0.98044	0.9943927	0.775387	1	-0.9966
1	4.20E+17	0.992153	-0.9858712	-0.80034	-0.9966	1

Table 8: Coverage-20 degrees, Data Summary

As mentioned above, the statistics are very degraded. The statistical deviation of the *g* parameter is calculated to be +/-14.3 units when *g* = -1 and the radius error is calculated to be at +/-8.79 units when radius = 1 unit. The condition numbers are suggesting that even a small change will greatly influence the stability of the algorithm. This phenomenon occurs because this is a point where the program numerically overflows if lower coverage is introduced. There is no stability of the algorithm at this point and the numerical summary demonstrates it very clearly.

4.3 Summary of Results

It should be noted that the software presented in this study, Point Cloud Data 3D Adjuster and Analyser, is comparable with software Cyclone by Cyra (Hong, 2003). However, testing has revealed that Point Cloud Data 3D Adjuster and Analyser is able to calculate a more accurate approximation of the radius. Table 9 depicts a comparison of pipe diameter measurements for the two software packages.

Pipe ID	Materials	Actual Diameter	3D Adjuster	Errors	Cyclone by Cyra	Errors
1	Brass	38.02	33.81	4.21	29.99	8.03
2	PVC	38.05	21.74	16.31	21.04	17.01
3	Stainless Steel	31.69	24.60	7.09	23.63	8.06
4	Galvanized Iron	80.9	74.64	6.26	77.28	3.62
5	Steel	63.45	54.35	9.10	55.56	7.89
6	Ceramic	111.4	112.21	-0.81	109.23	2.17
7	Aluminum	113.25	113.09	0.16	110.77	2.48

Table 9: Results of Diameter Measurements by Two Different Software Packages (Unit: mm)

5.0 CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

The importance of using a full set of numerical analyses in order to establish algorithm stability was demonstrated in this thesis. Various numerical methods were implemented to derive the goodness-of-fit of the cylinder. The findings indicate that it is possible not only to find the goodness-of-fit of the results, but also to foresee the behaviour of the algorithms if the condition numbers were calculated. The condition numbers combined with the use of standard deviations and correlation-coefficient matrices in a controlled environment allowed to analyze the solution and establish the behaviour of the solution as changes were introduced into the point data cloud.

The test of the g parameter reveals that as this parameter increases, angles μ and ϑ increase in precision. This is particularly noticeable with the μ angle, which is extremely poorly defined when it is wrapped around the z -axis and its centre of gravity corresponds to the origin of the coordinate system. This finding is not surprising given that angles are better defined over great distances. This is also clearly seen in the correlation-coefficient matrices. The correlations between the g parameter and the μ angle and the ϑ angle are almost 1 as the distance is increased to 100000 units.

The parametrization methods used to derive the critical parameters in the algorithm are taking full advantage of this property. The *alpha* (α) angle is not influenced by this property since it is defined locally and it is not dependent on the distance from the origin. The correlation-coefficient matrix clearly shows almost a zero correlation between the g parameter and the *alpha* (α) angle.

Moreover, the radius of the cylinder is not influenced by increasing the distance from the origin. This allows the radius to be well defined wherever the cylinder is positioned. In addition, the increase in the distance allows the radius to be less correlated with the g parameter. The only degradation in the quality of the estimates is seen in the g parameter alone. However, this degradation is minimal if compared to the distance. For example, it was observed that at a distance of 100,000 units, the degradation of precision is 0.3819 units, which corresponds to a loss of precision of 3.8 centimetres in 1 km.

The test of the coverage of the pipe revealed that decreasing the coverage does not influence the μ and ϑ angles greatly. Therefore, the μ and ϑ angles are not a function of the coverage. However, the test strongly suggests that the α -alpha angle is strongly influenced by the changes in coverage of the pipe. The most susceptible parameters are the g parameter and the radius. They both decrease in precision as the coverage is decreased. Moreover, the condition numbers drastically increase in value as the threshold of 130 degrees is exceeded. This is a sign that less than 130-degree coverage will destabilize the precision of calculations. The 100-degree coverage is an example of this trend. The 100-degree coverage test shows a rather large decrease in the precision of the g parameter and the radius. Those errors are amplified even more in the 70-degree and 20-degree coverage configurations.

5.2 Recommendations

One of the recommendations for data collection is to maintain the scan coverage between 130 degrees and 360 degrees, since this coverage produces the most reliable algorithms. Moreover, it is recommended that a distance of at least 10 times the radius is maintained when pipes are scanned. The distance should not exceed 10,000 times the radius. This is necessary to ensure that angles are well defined and that the g parameter has not been corrupted. In addition, it is recommended that the effects of operational errors on the solution are analyzed. Operational errors occur during data collection and they may distort the data and affect the solution. It is also desirable to study data filtering techniques in order to create algorithms that are able to detect and reject blenders and outliers.

Furthermore, intensity values should be taken into consideration to improve the quality of the solution. Studying the influence of incident angles and range may produce additional useful information in calculating spatial orientation. Also, the use of different materials as targets will help to better understand the point cloud data coverage of the pipe.

REFERENCES

- Anderson, E., Crippin, P. W., Davidson, L., Rachich, F. J., & Zorzitto, F. A., 1988. *Algebra and Geometry*. Toronto: Holt, Rinehart, & Winston of Canada.
- Byne, G. F., Crapper, P. F., & Mayo, K. K., 1980. *Monitoring land-cover change by principal component analysis of multitemporal landsat data*. *Remote Sensing of Environment*, 10, 175-184.
- Dineen, S., 1998. *Multivariate calculus and geometry*. New York, NY: Springer-Verlag.
- Faddeev, D.K., & Faddeeva, V.N., 1963. *Computational Methods of Linear Algebra*. New York, NY: Freeman.
- Fausett, L. V., 1999. *Applied numerical analysis using MatLab*. Upper Saddle River, NJ: Prentice Hall.
- Fung, T., & LeDrew, E., 1987. *Application of principal component analysis to change detection*. *Photogrammetric Engineering and Remote Sensing*, 53 (12), 1649-1658.
- Hong, J., 2003. *3D modeling of industrial piping systems using digital photogrammetry and laser scanning*. Toronto: Ryerson University.
- Jensen, J.R., 2004. *Introductory Digital Image Processing: A Remote Sensing Perspective (3rd ed.)*, Prentice Hall.
- Kwon, S., Liapi, K. A., Sreenivasan, S. V., & McLaughlin, J. T., 2002. *Human-assisted object fitting to sparse range point clouds for rapid workspace modeling in construction automation*. BFRl Publications Online, URL: <http://fire.nist.gov/bfrlpubs/build02/PDF/b02107.pdf>
- Lukacs, G., Marshall, A. D., & Martin, R. R., 1997. *Geometric least-squares fitting of spheres, cylinders, cones and tori*. *Copernicus*, 1068, 1-20.
- Marshall, D., Lukacs, G., & Martin, R., 2001. *Robust segmentation of primitives from range data in the presence of geometric degeneracy*. *IEEE* 23 (3), 304-314.
- Press, W. H., Flannery, B. P., Teukolsky, S. A., & Vetterling, W. T., 1986. *Numerical recipes: The art of scientific computing*. New York, NY: Cambridge University Press.
- Stewart, J., 1995. *Calculus (3rd ed.)*. Pacific Grove, CA: Brooks/Cole Publishing.

Watson, G. A., 2000. *Least squares fitting of parametric surfaces to measured data*. ANZAM, 42, C68-C95.

Werghi, N., Fisher, R. B., Robertson, C., & Ashbrook, A. P., 2000. *Faithful recovering of quadric surfaces from 3D range data by global fitting*. International Journal of Shape Modeling, 6 (1), 65-78.

Werghi, N., Fisher, R. B., Robertson, C., & Ashbrook, A. P., 2000. *Shape reconstruction incorporating multiple non-linear geometric constraints*. Constraints Journal, 7 (2), 117-149.

www.j-techdesign.com

APPENDIX

Software and directory folders are included on the attached CD.

Input and output test files are included on the attached CD.