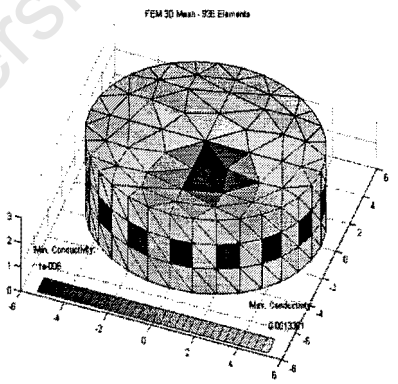
UNIVERSITY OF CAPE TOWN

SIGNAL AND IMAGE PROCESSING FOR ELECTRICAL RESISTANCE TOMOGRAPHY

PREPARED BY: Thoneshan Naidoo, masters student in the Department of Electrical Engineering at the University of Cape Town

PREPARED FOR: The Department of Electrical Engineering at the University of Cape Town



31 December 2002

Thesis submitted in fulfilment of the requirements for the degree of Master of Science (MSc) in Electrical Engineering.

II. TERMS OF REFERENCE

Dr. A. J. Wilkinson proposed this thesis. The project follows on from an undergraduate thesis that was completed by the author (Naidoo, 2001). The masters topic was put forward to the author in January 2002. Dr. Wilkinson is a senior lecturer at the University of Cape Town who is involved with the development of the hardware and software for electrical resistance tomography. The intention of this thesis is to expand the field of electrical resistance tomography at the University of Cape Town. During informal meetings with Dr. Wilkinson the following requirements for this project were laid out:

To continue from the undergraduate project (Naidoo, 2001) and complete the reconstruction algorithm.

- To extend the algorithm into three dimensions.
- To implement the above algorithm in a high level programming language.
- To make the algorithm fast.
- To document the knowledge acquired from the research.
- To document the work completed by the author.
- To hand in the thesis documentation before July 2004.

I. ACKNOWLEDGEMENTS

The author greatly appreciates the people who have contributed to the input of this thesis.

Thanks go the following people:

Dr. A. J. Wilkinson. His contribution and involvement has been invaluable. His involvement covered the following areas:

i. Introducing the author to the field of electrical resistance tomography;
ii. Providing information on the topic in the form of conference papers, books and conversations;
iii. Supervising and providing guidance on the direction of this thesis;
iv. Searching and securing funding for this students studies.

The author is extremely grateful to have Dr. Wilkinson as his supervisor.

E. W. Randall who jointly developed the two-dimensional hardware system at the University of Cape Town thus allowing real voltage measurements to be captured and fed into the software developed by author.

III. ABSTRACT

Electrical Resistance Tomography (ERT) is in essence an imaging technique. In ERT current is injected into and removed from a vessel via paired electrodes. The resulting voltage measurements are captured between the remaining electrode pairs. The principle behind ERT is to map these boundary voltages into a conductivity distribution that represents the domain of the vessel.

The author has coded a versatile reconstruction algorithm based on the Newton-Raphson algorithm. The knowledge gained by implementing the algorithm is documented in this thesis. The literature covers the basic aspects of two-dimensional and three-dimensional ERT. It is hoped that this thesis will create a greater interest in ERT at the University of Cape Town (UCT) and also act as a building block for further developments.

The thesis starts by presenting the basic concepts of ERT such as the underlying equations, the various boundary measurement strategies and a global perspective of ERT. The nature of this thesis is on software reconstruction and in so doing information on the incorporation of the Finite Element Method in ERT is provided. The thesis goes on to provide information about the reconstruction algorithms, which incorporate regularization.

A novel aspect of this thesis involves the calibration and pre-processing of boundary voltages. These concepts were conceptualised and developed during formal communications with Dr. Wilkinson (2002) and Randal (2002). The calibration schemes try to eliminate the potential errors that can arise in the captured data thus allowing for a clearer image to be reconstructed.

This thesis further develops the idea of parallelizing the Newton-Raphson algorithm to increase the speed of the algorithm. Various schemes on how this parallelization is achievable are put forward.

Based on the results, conclusions can be drawn that the reconstructed algorithms are correctly working. The author is optimistic that the developed software will become more robust and accurate with the recommendations that are put forward.

IV. TABLE OF CONTENTS

V. LIST OF ILLUSTRATIONS

VI. LIST OF TABLES

VII. GLOSSARY

σ	Conductivity
∇	Divergence operator
Φ	Potential
e	Number of Finite Elements in Domain
E	Electric Field (V/m)
EIT	Electrical Impedance Tomography
ERT	Electrical Resistance Tomography
FEM	Finite Element Method
J	Current Density (A/m^2)
L	Number of Electrodes
MPI	Message Passing Interface
n	Number of Nodes in Mesh
p	Number of Current Injection Patterns
PC	Personal Computer
PVM	Parallel Virtual Machine
RPI	Rensselaer Polytechnic Institute
S	Smoothness Matrix or Regularization Matrix
UCT	University of Cape Town
UMIST	University of Manchester Institute of Science and Technology
V	Voltage
V_{cal}	Calculated Voltages
V_{mea}	Measured Boundary Voltages

VIII. DECLARATION

The author declares that the software developed for this thesis operates to the exact specification as is referred to in the literature.

Furthermore the author declares that the literature presented in this thesis is adequately and properly referenced.

Signed by candidate

T. Naidoo

1. INTRODUCTION

The following subsections introduce the reader to the concept of electrical resistance tomography.

1.1. DEFINITION OF ELECTRICAL RESISTANCE TOMOGRAPHY

Electrical Resistance Tomography (ERT) is in essence an imaging technique. Imaging the contents of non-transparent objects can provide vital information that can be used to control or change internal conditions. Electrical Resistance Tomography is one such imaging technique that can be used to gain an internal visual perspective of objects such as pipes, thoracic cavity, brain, etc. ERT is based on mapping the conductivity distribution that occurs within the object.

This conductivity distribution is set by evenly placing electrodes around the object to be imaged and then injecting and removing an electric current via these electrodes. The remaining electrodes measure the voltage difference between pairs of electrodes. These voltages are used with the aid of an algorithm to image the conductivity distribution within the cylinder and indirectly the contents of the object.

1.2. AIMS AND OBJECTIVES OF THE THESIS

In 2001 an undergraduate thesis was completed by the author (Naidoo, 2001), which investigated the forward solution of two-dimensional ERT. This thesis is a continuation of the undergraduate thesis.

The aims of this thesis are to firstly conduct a study of the reconstruction algorithms involved with two and three-dimensional electrical resistance tomography. Secondly, after researching the subject, the author is required to use the knowledge gained to develop signal-processing tools for image reconstruction. More specifically, a reconstruction algorithm is to be implemented that will integrate with the software developed at undergraduate level and also interact with the UCT Current-pulse ERT system.

1.3. HISTORY OF ELECTRICAL RESISTANCE TOMOGRAPHY

Resistance imaging can be traced back to the 1920s (Dickin et al, 1996b). Geophysicists in this era inserted an array of metallic electrodes into the ground and passed a current through them. By measuring the voltages on the unused electrodes, a map of the sub-surface strata could be compiled. This allowed for a more precise delineation of oil-bearing rock and was used by companies such as *Schlumberger* (Dickin et al, 1996b).

In 1978 the first papers on impedance imaging were introduced by Henderson and Webster (Brown, 2001; Breckon, 1990). This method was seen as a low cost and portable alternative to the x-ray system that was favourable at that time (Dickin et al, 1996b). Over the years a number of contributions were made

by various scientists and engineers to improve the system. Various measurement strategies were proposed and numerous reconstruction algorithms were developed. With the advancement in technical computing electrical tomography was no longer seen as a two-dimensional system. In the 90's interest shifted to three-dimensional systems.

The field of Tomography has grown with numerous papers published on both the hardware and software. Its popularity has resulted in numerous conferences and special journal publications. With further advancements in computing power it can be expected that electrical tomography will become faster and more accurate creating greater interest in the field.

1.4. ELECTRICAL RESISTANCE TOMOGRAPHY AT THE UNIVERSITY OF CAPE TOWN

Electrical Resistance Tomography is a relatively new field of study in the Electrical Engineering department of UCT. One of the systems, namely the UCT Current-pulse ERT system was used by the author to capture the voltage data that is fed into the authors reconstruction software. This novel two-dimensional ERT hardware system uses a bi-directional current pulse, which reduces the need for DSP demodulation techniques implying that high-speed measurements are attainable (Cilliers et al, 2001; Randall et al, 2001).

Further development on the hardware has been undertaken in the Electrical Engineering department at the University of Cape Town (Durrett, 2002; Hooper, 2001). The most recent work completed by Durrett (2002) shows potential for the system to collect frames at a rate of 1000 frames per second and higher.

At the end of 2001, the data captured from the UCT Current-pulse ERT system was reconstructed with software developed externally. With the progress of this thesis a two-dimensional algorithm based on the Newton-Raphson method was developed by the author and is currently used as a means to reconstruct images for the UCT Current-pulse ERT two-dimensional system. The author has proceeded further to develop a three-dimensional algorithm, which is also based on the Newton-Raphson method. It should be noted that the UCT Current-pulse ERT system has not as yet been extended for a three-dimensional ERT system.

1.5. SOFTWARE PROJECT DEVELOPED

As part of the thesis study the author was asked to develop software that implements the reconstruction algorithms for two and three-dimensional ERT. The software was required to complement the software developed for the author's undergraduate thesis. The two algorithms developed were aptly named Tomo_Imager_2D and Tomo_Imager_3D. The above-mentioned software was programmed in *Matlab,* Version 6, Release 12. *Matlab* is a high level mathematical programming language created by *MathWorks,* and is ideally suited for the nature of this thesis.

1.6. SCOPE AND LIMITATIONS OF THESIS

This thesis is limited to electrical resistance tomography and does not include closely related fields such as electrical capacitance tomography and other forms of tomography. The research covers the investigation of the software concepts and touches briefly on the hardware concepts.

The UCT Current-pulse ERT system is developed for two-dimensional tomography but has not been extended to a three-dimensional system. The

author has developed the software for three-dimensional ERT but had to then adapt the software since he could not attain any true 3D measurements. The author's adapted software for 3D incorporated the use of a single layer of electrodes.

The images presented in this thesis were reconstructed on a Celeron X PC with 500MHz clock speed and 128 megabytes of ram. This proved a limitation for 3D imaging. The 3D elements posed a large burden on the computers resources and meshes with greater than 4000 elements caused the program to halt due to insufficient memory.

1.7. PLAN OF DEVELOPMENT

Firstly the basic concepts underlying electrical resistance tomography will be discussed in Chapter Two. Chapter two goes on to derive the equations needed for electrical resistance tomography. This is followed by Mesh Discretisation in Chapter Three. Chapter Four explains the concepts of the Finite Element Method and its use in tomography. The next chapter, Chapter Five discusses the reconstruction algorithms. Chapter Six describes the novel concepts developed at UCT regarding the Calibration and Pre-processing of Data. Next the concept of parallelizing the Newton-Raphson algorithm is presented in Chapter Seven. The theory presented up to this point is put into a practical context and elaborated on in Chapter Eight. This is followed by the results of the algorithm. Lastly conclusions will be drawn and recommendations provided.

2. BASIS OF ELECTRICAL RESISTANCE TOMOGRAPHY

ERT requires the integration of hardware and software. Thus this chapter will provide a general global perspective of ERT.

2.1. GLOBAL VIEW OF ELECTRICAL RESISTANCE TOMOGRAPHY

The following diagram aptly depicts ERT in a modular form.

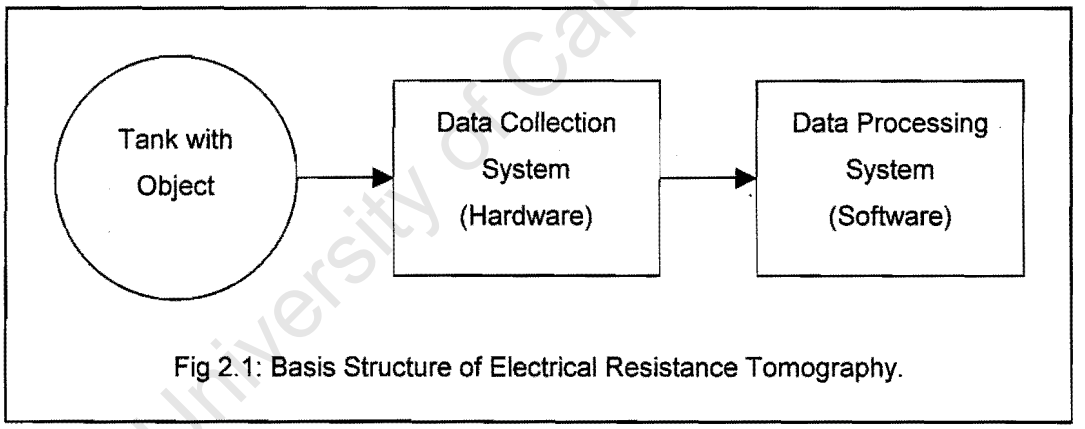


Fig 2.1: Basis Structure of Electrical Resistance Tomography.

Figure 2.1 illustrates the idea that tomography can be viewed into three separate aspects, namely (Ying, 2000):

i. Tank with Object and sensing system,
ii. Data Collection System, and
iii. Data Processing System.

In Figure 2.1, a computer controls both the Data Collection System and Data Processing System. The nature of this thesis is software but it is not limited to the Data Processing System but extends partially to the other two aspects.

2.1.1. Data Collection System

The Data Collection System can be further illustrated by means of the diagram below.

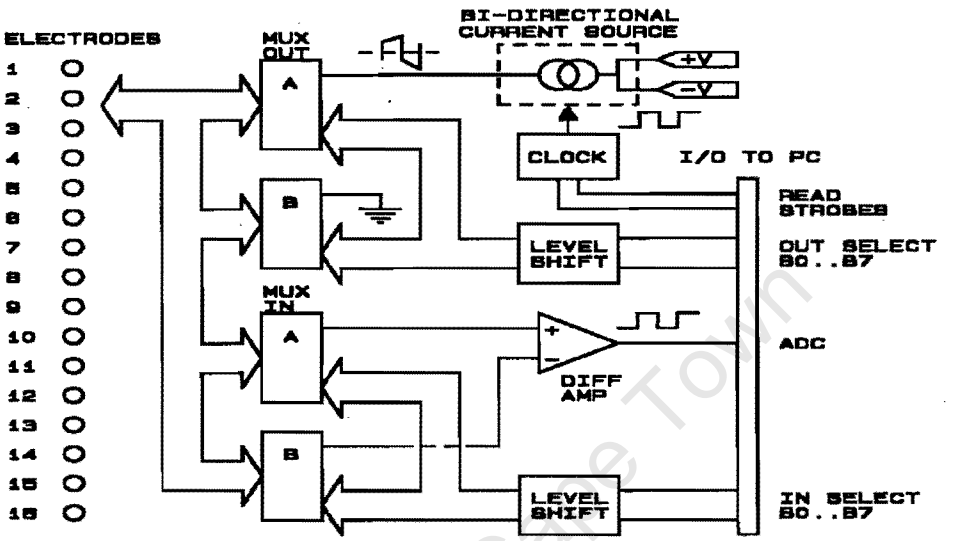


Fig 2.2: Data Acquisition Hardware

Source: Randall et al, 2001.

Figure 2.2 implies that the data collection system works in the following way: A current is sent and received through the multiplexor to the necessary electrodes. The multiplexor is then used to capture the voltage difference on the remaining appropriate electrodes. These signals are buffered and read by the Analogue to Digital Converter (ADC). The data is now ready to be processed. It should be noted that the above diagram is specific to the UCT Current-pulse system.

For further information on the data acquisition system that is depicted above the reader is directed to the following documentation (Cilliers, 2001; Randall, 2001; Hooper, 2001). In 2002 further work was completed on the hardware to increase the speed of the system to 400 frames per second (Durrett, 2002b).

The present speed for data collection is in the order of 1000 frames per second (Randall, 2002).

2.1.2. Data Processing System

The Data Processing System for ERT can be further split up into three distinct areas, namely:

i. Meshing of a domain

ii. Forward solution, and

iii. Inverse solution.

It is imperative that the domain to be imaged is modelled as accurately as possible since this model is used to solve for the unknown conductivities in the domain (Cheney et al, 1990; Hua et al, 1990; Kolehmainen et al, 1997a; Molinari et al, 2001b; Polydorides, 2000; Vauhkonen et al, 2000). (By domain it is implied to the tank and its geometry).

The relationship between the forward solution and the inverse solution is best described by the following figure.

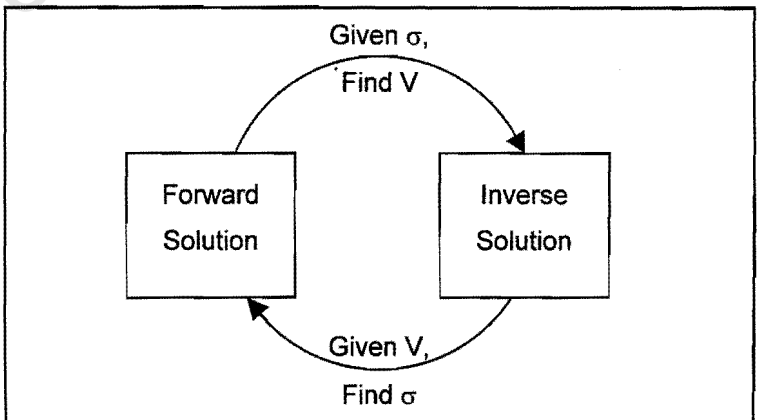


Fig 2.3: Cyclic Process of Data Processing System

Figure 2.3 depicts the cyclic nature of ERT. In the forward solution one has a conductivity distribution within the domain and it is required to calculate the voltages that form at the boundary. The inverse problem is concerned with using these boundary voltages to find a unique conductivity distribution in the domain (Belward, 1997; Grootveld et al, 1997; Hua et al, 1990; Pinheiro et al, 1997; Vauhkonen, 1997). The following flow chart is used to encapsulate the essence of the imaging technique.

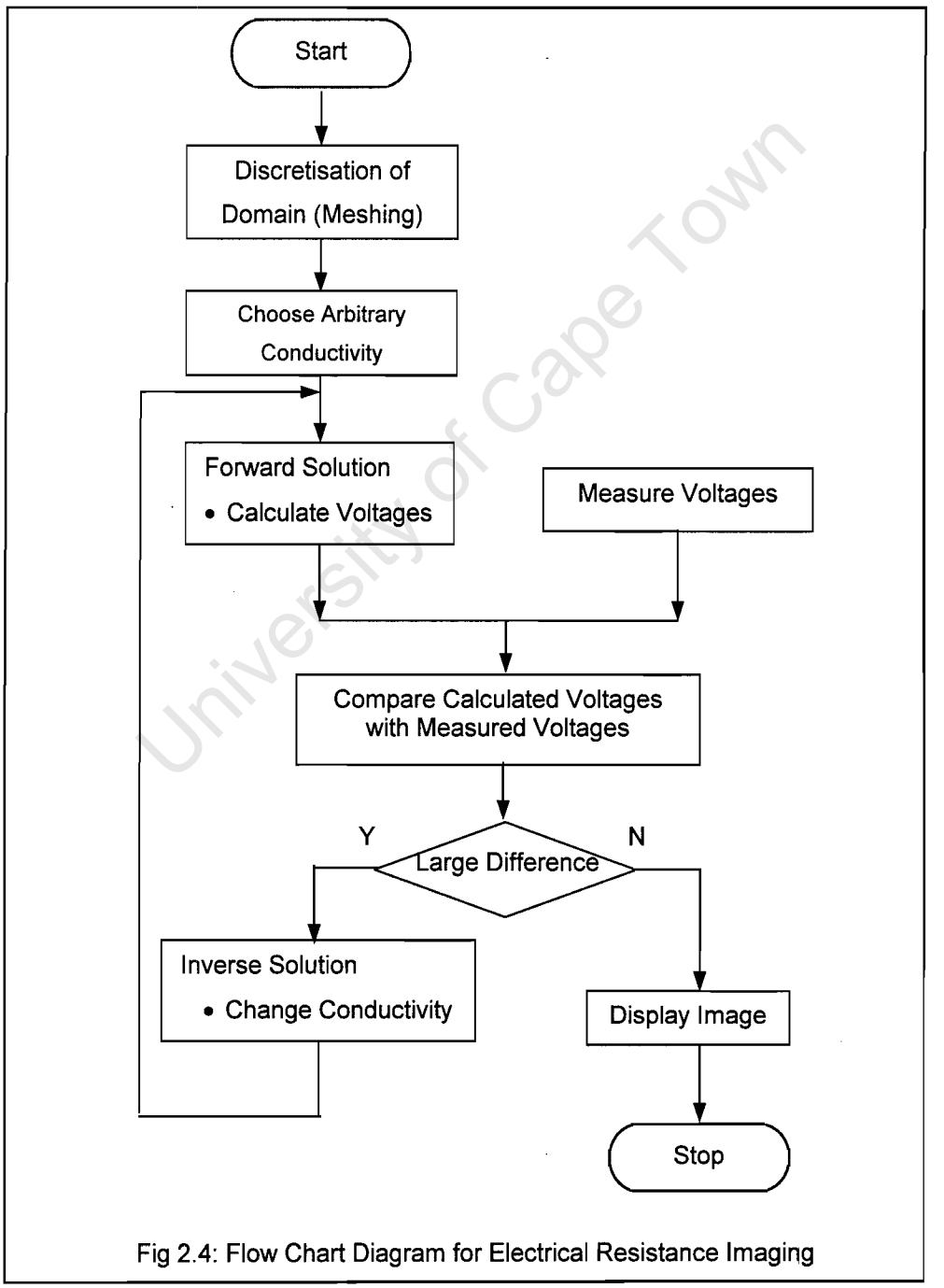


Fig 2.4: Flow Chart Diagram for Electrical Resistance Imaging

Figure 2.4 is adapted from a flow chart in Naidoo (2001). It shows the imaging of the object with the means of an iterative reconstruction algorithm. The software concepts outlined in this chapter will be discussed in greater detail in the forthcoming chapters.

2.2. CHARACTERISTIC EQUATIONS OF ELECTRICAL RESISTANCE TOMOGRAPHY

The governing equations of electrical resistance tomography can be derived from Maxwell's equations. The following equations are based on references Belward, 1997; Cheney et al, 1999; Dickin et al, 1996a; Hua et al, 1990; Kolehmainen, 2001; Murai, 1985; Polydorides, 2000; Tattersfield, 1999; Vauhkonen, 1997; Yorkey, 1990.

In a conductive medium the electric field E created by charges on the electrodes applies a force on the free electrons in the medium. This force enables the electrons to propagate and hence a current to be established. Thus under quasi-static conditions or in a linear isotropic medium the resulting current density can be expressed as:

$$J = \sigma \cdot E \tag{2.1}$$

where J is the current density (A/m^2).

σ is the conductivity of the medium (S/m).

E is the electric field (V/m).

If there are no electrical current sources or sinks within the domain then the following principle is established:

$$\nabla \cdot J = 0 \tag{2.2}$$

where ∇ is the divergence operator.

Proof of the following well-known equation relating the electric field to the potential Φ can be found in Tattersfield (1999).

$$E = -\nabla \Phi \tag{2.3}$$

Combining the three equations produces Poisson's equation:

$$\nabla \cdot (\sigma \nabla \Phi) = 0 \tag{2.4}$$

For a more mathematical orientated derivation of equation 2.4 the reader is referred to Cheney et al (1999). Equation 2.4 in collaboration with the following boundary condition:

$$\sigma \, \partial\Phi/\partial n = J \tag{2.5}$$

describes the potential distribution within and on an inhomogeneous domain. If the domain is homogenous then Poisson's equation can be reduced to Laplace's equation:

$$\nabla^2 \Phi = 0 \tag{2.6}$$

Solution of the above two equations enables the image to be reconstructed.

2.3. CURRENT INJECTION AND VOLTAGE MEASUREMENT STRATEGY

The manner in which current is injected and removed from the domain directly impacts on the quantity of independent information extractable and indirectly on the quality of the image reconstructed. The following sections describe the options that can be implemented in ERT.

2.3.1. Current Source versus Voltage Source

In the 80's the issue on whether to inject current and measure voltage or inject voltage and measure current became an issue of much discussion. The trend from there on was to inject current and measure voltages (Hua et al, 1990). The rationale behind this transformation was provided by Brown et al (1985). The logic behind such a choice is twofold in nature. Firstly, voltmeters have a large input impedance whereas ammeters have a small input impedance thus the contact impedance that exists between the electrode and the domain has a negligible effect on voltage measurements. Secondly, the output impedance of a current source is significantly larger than that of a voltage source (Cilliers et al, 2001; Dickin et al, 1996b; Randall et al; 2001, Vauhkonen, 1997).

2.3.2. Single Current Source versus Multiple Current Sources

Typically ERT has been implemented with a single current source. This current is then multiplexed to the relevant electrodes. Research at the Rensselaer Polytechnic Institute (RPI) shows that to improve the resolution of the image by increasing the number of electrodes, current needs to be

simultaneously applied to all the electrodes (Cheney et al, 1999; Gisser et al, 1988). RPI's ACT system thus employs as many current generators as electrodes. This allows for the application of trigonometric current patterns. For further reading on trigonometric and optimal current patterns the reader is referred to Cheney et al (1999); Gisser et al (1988); Gisser et al (1990); Kolehamainen (2001); Vauhkonen (1997), and the references cited within. However, the advantage of greater resolution in multiple current sources is negated by the requirement of more complex circuitry.

2.3.3. Alternating Current versus Direct Current

Traditionally alternating sinusoidal current was regarded as the standard method for ERT imaging. Typically the ac current is injected at a high frequency of 50kHz (Dickin et al, 1996b). Direct current injection was ignored due to change in resistance at the electrodes due to electrochemical reactions that transpire. Recently papers have been produced (Cilliers et al, 2001; Randall et al, 2001; van Weereld et al, 2001), which demonstrate the ERT technique with the use of a bi-directional pulsed current. The 'dc' bi-directional pulse technique is advantageous over the conventional ac method in the following ways (Boone et al, 1995; Cilliers et al, 2001; Randall et al, 2001; van Weereld et al, 2001):

i. Bi-directional current eliminates any long-term polarisation effects at the electrodes.

ii. Compensates for dc offset voltages that exist between electrodes.

iii. The absence of phase shifts greatly simplifies the measurement and circuitry.

iv. The simplicity of the circuitry allows for the measurement speed to be significantly quicker than conventional ac techniques, which indirectly allows for real-time processing.

2.3.4. Two-Electrode Method versus Four-Electrode Method

In the two-electrode method current is injected into and removed from paired electrodes and the voltages are measured from all electrodes. For such an injection method the number of measurements that can be extracted is:

$$L(L-1) \tag{2.7}$$

where L = number of electrodes.

However due to the reciprocity theorem which states that reversing the voltage measurement and current injection electrodes would give an identical value of resistivity (Webster, 1990) implies that the number of independent voltages attainable is:

$$\frac{L(L-1)}{2} \tag{2.8}$$

In the four-electrode method current is injected into and removed from a pair of electrodes and voltages are measured from the remaining electrodes. For such an injection method the number of measurements that can be extracted is:

$$L(L-3) \tag{2.9}$$

Similar to the two-electrode method, the number of independent measurements that can be extracted due to the reciprocity theorem (Webster, 1990) is limited to:

$$\frac{L(L-3)}{2} \tag{2.10}$$

Thus the four-electrode technique can extract fewer independent measurements than the two-electrode method. However, the four-electrode method is advantageous over the two-electrode method in that the error due to contact impedance is minimised since no measurement is taken on the excited electrodes (Barber, 1989; Kolehmainen, 2001; Vauhkonen, 1997).

From this point on the author will concentrate on the four-electrode technique, as this is the current method employed at UCT. There are two widely used schemes that fall into the four-electrode field, namely (Dickin et al, 1996b; Kolehmainen, 2001; Vauhkonen, 1997):

i. Adjacent Strategy (Neighbouring)

ii. Opposite Strategy (Polar)

These two schemes will be discussed below.

The Adjacent Method

The adjacent scheme is the more widely used of the two schemes described here. The strategy is best described with aid of the figure below.

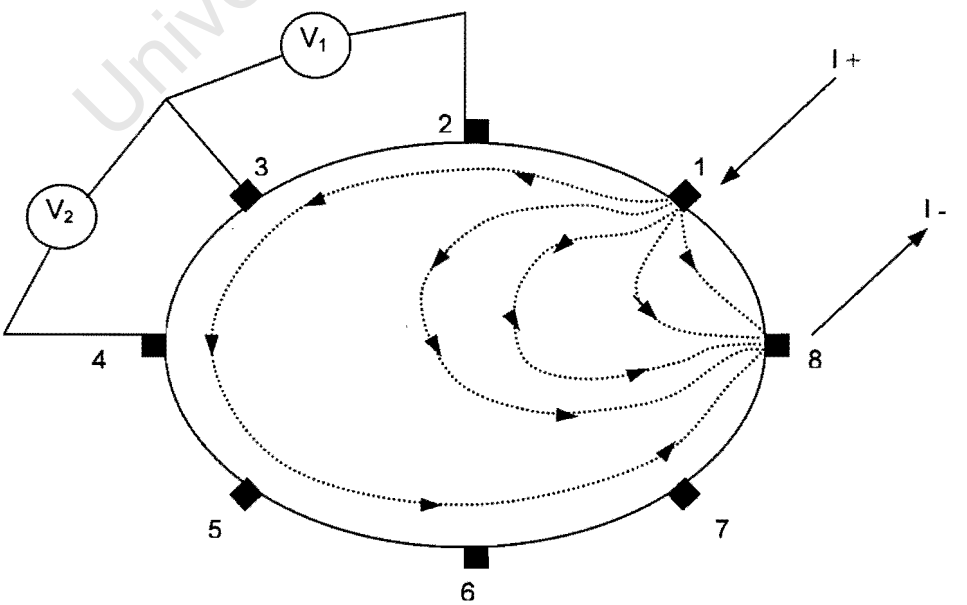


Fig 2.5: Adjacent Four-Electrode Current Injection Strategy

Figure 2.5 depicts the technique of the adjacent strategy. Verbally this can be described as follows. Firstly current is injected at electrode 1 and removed via an adjacent electrode (electrode 8). The first differential voltage V_1 is measured at the next passive pair of free electrodes (electrodes 2 and 3). V_2 is then measured from electrodes 3 and 4. When the final voltage measurement V_5 (electrodes 6 and 7) for this current injection is captured the current is injected into and removed from the next passive pair of electrodes (electrodes 1 and 2). Following this strategy there are 20 independent measurements for the above instance as calculated from Equation 2.10.

As can also be inferred from Figure 2.5, the current density projected as dashed lines is non-uniform. This is caused by most of the current travelling along or close to the boundary concentrated between the injection pair (Avis et al, 1994; Dickin et al, 1996b; Hua et al, 1993; Kolehmainen, 2001; Vauhkonen, 1997). This implies that the adjacent strategy is less sensitive to resistivity changes in the centre than at the boundaries. Thus this strategy is superior if the region of interest to be imaged is located at the boundary but inferior if the region of interest to be imaged is located at the centre.

The Opposite Method

The opposite scheme is the less widely used of the two schemes described here. The strategy is best described with aid of the figure on the next page.

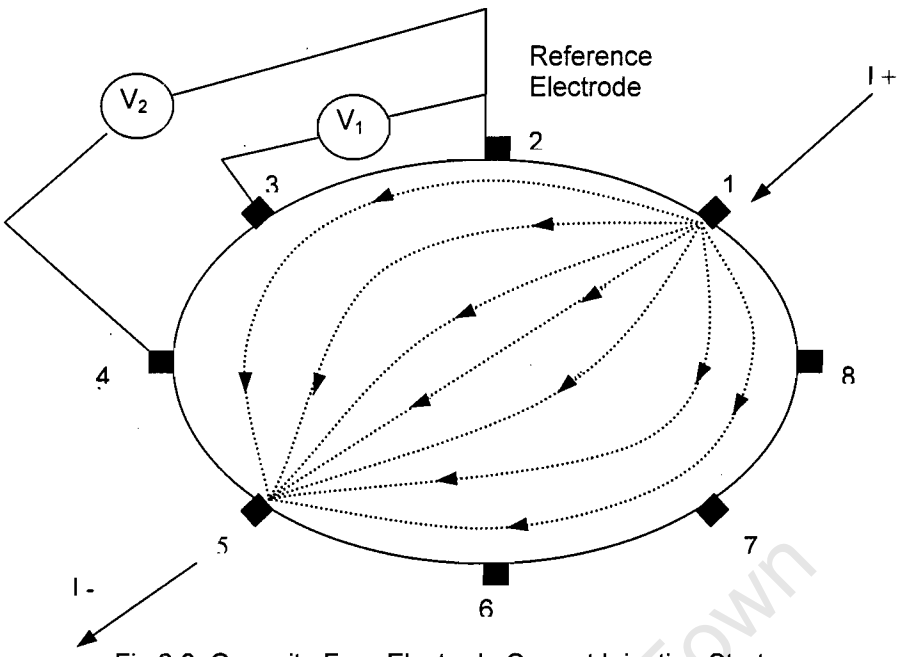


Fig 2.6: Opposite Four-Electrode Current Injection Strategy

Figure 2.6 depicts the technique of the opposite strategy. Verbally this can be described as follows. Firstly current is injected at electrode 1 and is then removed via a diagonal electrode (electrode 5). A reference electrode is usually stipulated from which voltages are measured. In the above circumstance the adjacent electrode (electrode 2) is the reference electrode. The first differential voltage V_1 is measured between the reference electrode (electrode 2) and the next passive electrode (electrode 3). V_2 is then measured from electrodes 2 and 4. When the final voltage measurement V_5 (electrodes 2 and 8) for this current injection is captured the current is injected into and removed from the next pair of polar electrodes (electrodes 2 and 6). The reference electrode will then become electrode 3. Following this strategy there is 20 independent measurements (which is the same as the adjacent configuration) for the above instance as calculated from equation 2.10.

As can also be inferred from Figure 2.5, the current density projected as dashed lines is uniform (Avis et al, 1994; Kolehmainen, 2001; Vauhkonen, 1997). This is due to the current being intentionally injected into and removed from polar electrodes. This implies that the opposite strategy is more sensitive

to resistivity changes in the centre than at the boundaries. Thus this strategy is superior if the region of interest to be imaged is located at the centre but inferior if the region of interest to be imaged is located at the boundary.

A note in passing: The question may be asked if there is a compromise between the above two methods. The answer is yes. The derived method is called the cross method (Hua et al, 1993). In general there are numerous other models of injection and measurement. One such method involves magnetically inducing current at electrodes with the use of coils (Gencer et al, 1992; Gencer et al, 1994; Gencer et al, 1996; Healey et al, 1992; Ruan et al, 1996). To read on such alternate models the reader is referred to Cheney et al (1999); Gisser et al (1988); Gisser et al (1990); Kolehamainen (2001); Vauhkonen (1997), and the references cited within.

3. DOMAIN DISCRETISATION FOR 2D AND 3D ERT

Domain discretisation forms an integral part of the Finite Element Method. The object of interest is required to be accurately modelled and subdivided into smaller sub-domains called finite elements. In the sections to follow some of the fundamentals behind two-dimensional and three-dimensional discretisation will be elaborated on. The reader is informed that work in this chapter is adapted from the author's undergraduate thesis (Naidoo, 2001).

3.1. TYPES OF FINITE ELEMENTS

For different dimensions there are various shapes (finite elements) that can be used to represent a domain. The boundaries of finite elements are demarcated with nodes (Cook, 1995; Pepper et al, 1992; Rao, 1999). The following subsections illustrate the types of finite element shapes that can be used for the three dimensions.

3.1.1. One-Dimensional Finite Elements

One-dimensional finite elements can be used to model forces such as pressure, heat or current flow through a resistor. Finite elements in one-dimensional space are limited to lines. An example is depicted below.

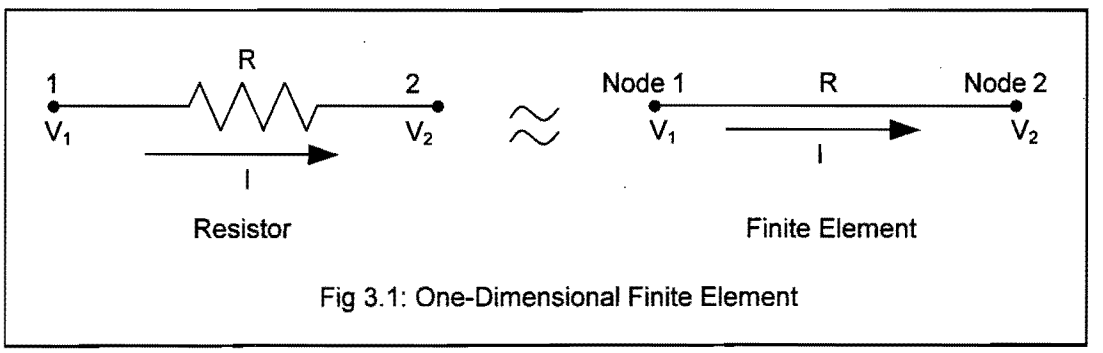


Fig 3.1: One-Dimensional Finite Element

Figure 3.1 illustrates a simple resistor 'R' modelled as a line, which represents a finite element in one-dimensional space.

3.1.2. Two-Dimensional Finite Elements

Two-dimensional finite-elements are used to represent bending and stress forces. It is also practical to use two-dimensional finite elements to model domains in ERT. Finite elements in two-dimensional space are limited to areas. Examples of such areas are depicted below.

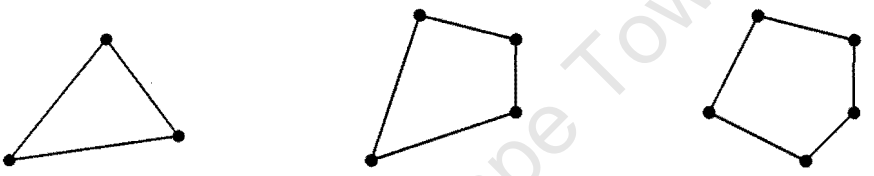


Fig 3.2: Two-Dimensional Finite Elements

Figure 3.2 shows three specific shapes that can be used as finite elements. The essential difference between each shape in Figure 3.2 is visible in the number of nodes that they contain. Elements with straight boundaries as depicted in Figure 3.2 are known as linear elements whilst elements with curved boundaries are known as higher order elements (Rao, 1999).

3.1.3. Three-Dimensional Finite Elements

Three-dimensional finite elements are used to represent forces that travel in three dimensions. It is the correct practical method to model electrical resistance tomography which is inherently a three dimensional problem. Finite elements in three dimensions are limited to volumes. Examples of such volumes are depicted in the figure on the next page.

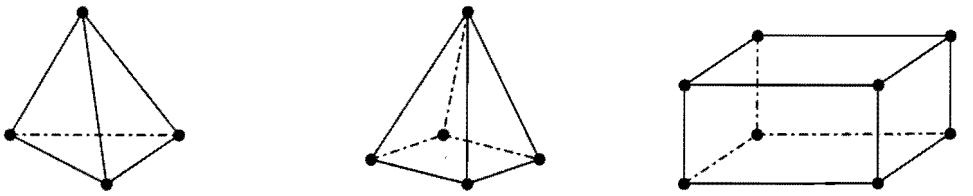


Fig 3.3: Three-Dimensional Finite Elements

Figure 3.3 shows three specific shapes that can be used as finite elements. The essential difference between each shape in Figure 3.2 is visible in the number of nodes that they contain. Elements with straight boundaries as depicted in Figure 3.3 are known as linear elements whilst elements with curved boundaries are known as higher order elements (Rao, 1999).

3.2. Requirements of Finite Elements to Form a Mesh

The previous subsection illustrated the types of elements that can be used for discretisation of a domain in various dimensions. The requirements of the domain elements will be outlined in the following subsections.

3.2.1. Conformity of Meshes

Conformity means compliancy. For a mesh to be conformal it must comply with the following conditions (George, 1991):

i. The sum of the elements must represent the domain i.e. ∑elements = domain.

ii. The elements of mesh must have a non-empty interior.

iii. The intersection of any 2 elements brings about one of the following:

   a. Reduction to an empty set.
   b. Reduction to a point.
   c. Reduction to an edge.
   d. Reduction to a face.

An example of a conformal and non-conformal mesh is illustrated below.

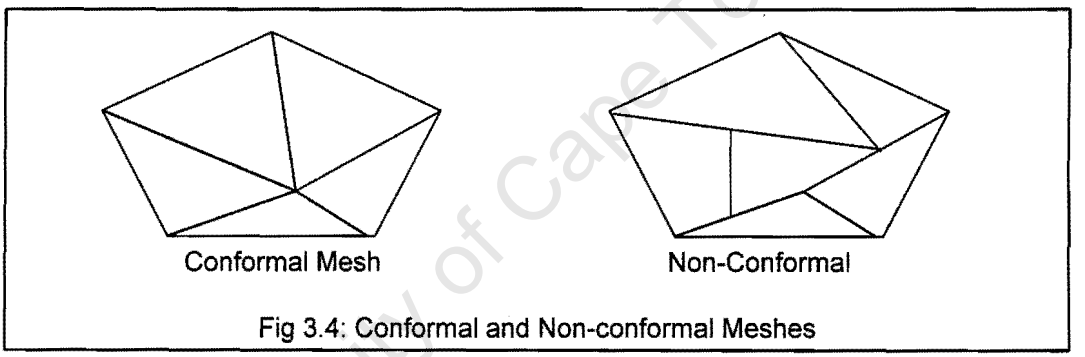


Fig 3.4: Conformal and Non-conformal Meshes

Source: George, 1991.

Attention of the reader is drawn to the conformal mesh in Figure 3.4. This is the type of mesh that the author will use for his software program.

3.2.2. Geometric Properties

The geometric properties that elements must comply to are (Burnett, 1987; Cook et al, 2002; George, 1991; Hua et al, 1990; Logan, 2002; Molinari et al 2001b; Pepper et al, 1992; Rao, 1999; Zienkiewicz et al, 2000):

i. Variations of size between adjacent elements must be gradual if not constant to prevent step changes in voltage.

ii. Regions with large changes in the field variable (i.e. temperature, voltage, heat, etc.) must contain a higher density of elements to accurately model the changes.

iii. For triangular elements the use of obtuse angles must be avoided. Obtuse triangles bias the field variable in a particular direction.

iv. Anisotropic properties of the field variable must be accounted for. For example, the sides of triangular elements can run in the direction of the largest field variable change.

v. The partitioning of elements in a domain must be consistent i.e. a node lying on the face of more than one element is a node in all the corresponding elements.

vi. Discretisation of a curved or irregular domain must be closely approximated by the boundary elements to improve accuracy of solution.

3.2.3. Physical Properties

The physical properties that elements must comply to are (George, 1991; Molinari et al, 2001b; Pepper et al 1992):

i. Each element must be denoted by a set of nodes located at its vertices. For the instance of higher order elements the elements must include nodes at points between vertices as is depicted in Figure 3.3.

ii. All nodes within the domain must be numbered globally and locally according to the element that they represent. This will allow for

assemblage of the stiffness matrix and hence computation of the finite element method.

iii. Nodes must be specified by a coordinate system. This will allow for the calculation of element areas required in the finite element method.

3.3. MESH REFINEMENT AND OPTIMISATION

A mesh can be refined to improve:

i. Accuracy and error of the forward solution.
ii. Time for convergence of the solution.
iii. Memory storage space required for node numbering.

Methods in which mesh optimisation can be implemented are elaborated on in the proceeding subsections.

3.3.1. Number of Elements

The number of elements in a domain has a determining factor on the spatial resolution and accuracy of the image (Hua et al, 1990; Molinari, 2001b). It is recommended that regions with large changes in the field variable have a higher concentration of elements (Cook, 1995; George, 1991; Pepper et al, 1992; Rao, 1999). Increasing the number of elements in these regions improves the accuracy of the image (Logan, 2002; Molinari et al, 2001b; Rao, 1999). However this is only true up to a certain point as illustrated in the following figure found on the next page.

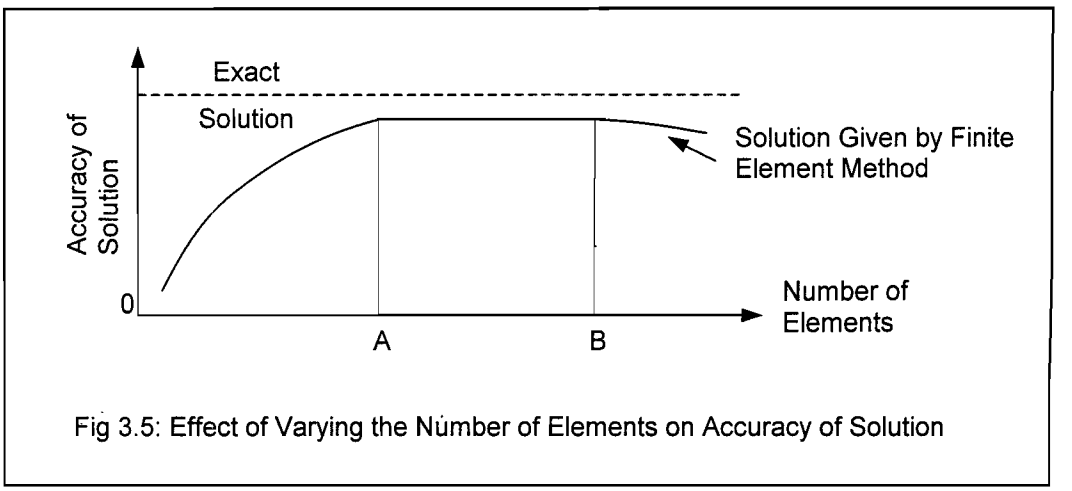


Fig 3.5: Effect of Varying the Number of Elements on Accuracy of Solution

From Figure 3.5 it can be deduced that there comes a point where increasing the number of elements has no further improvement on the accuracy of the solution (Molinari, 2001b; Rao, 1999). This point is depicted as point A on the graph. It is also possible that the solution will become less accurate when a large number of elements are used as depicted by point B on the graph (Hua et al, 1990; Rao, 1999). The divergence at point B can be attributed to the increase in numerical rounding errors due to increase in the number of elements.

A realistic and more significant pitfall of increasing the number of elements is that the computation time of the finite element method will increase exponentially (Molinari et al, 2001b; Pepper et al, 1992). This is illustrated below.

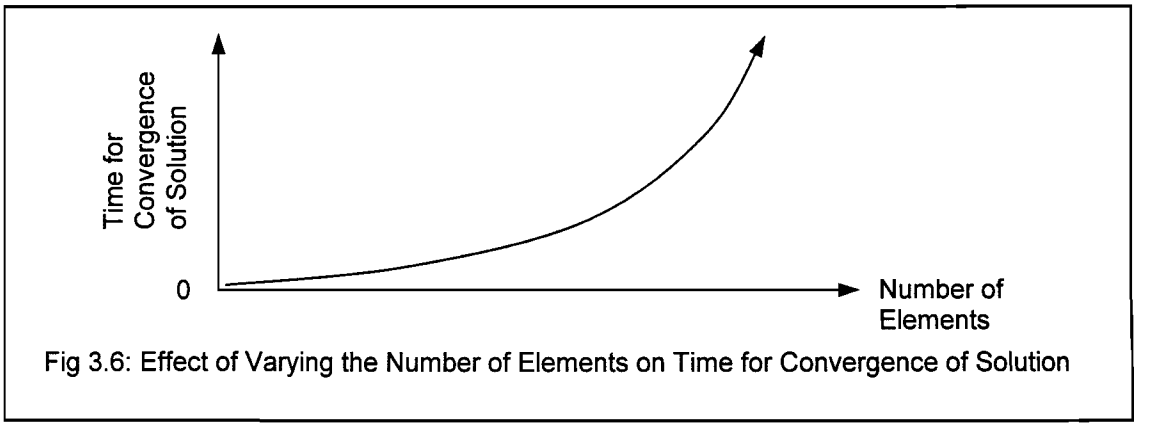


Fig 3.6: Effect of Varying the Number of Elements on Time for Convergence of Solution

Since increasing the number of elements positively improves the accuracy of solution and negatively escalates the time for convergence of the solution a trade off between the two needs to be made.

3.3.2. Size of Elements

The solution of the finite element method converges to the true solution as the element size decreases to zero, i.e. size of elements 'squared' is inversely proportional to accuracy (Cook, 1995; Hua et al, 1990; Molinari et al, 2001b; Rao, 1999).

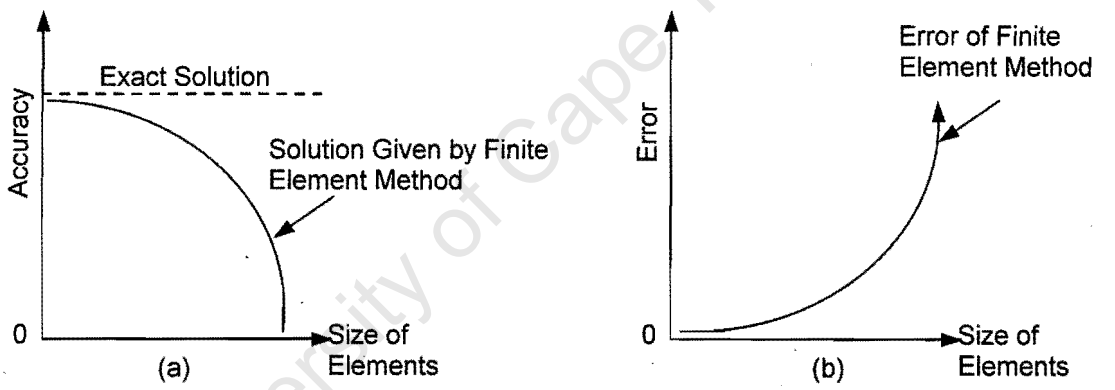


Fig 3.7: Effects of Varying the Size of Elements

Figure 3.7a illustrates this concept and conversely Figure 3.7b shows that the error decreases as the size of an element decreases. The pitfall to decreasing the size of the elements is that more elements are needed to discretise the domain. As mentioned in the previous section this implies that the time for convergence of the solution is increased exponentially.

When comparing the size of elements to one another, an aspect ratio is used. An aspect ratio is the ratio of the largest dimension of the element to the smallest (Burnett, 1987; Logan, 2002; Molinari, 2001b; Rao, 1999). The

aspect ratio of elements to be used depends on the field variable that is to be imaged. For static ERT imaging which has a low variation change, elements with an aspect ratio of unity such as equilateral triangles are preferred and provide more accurate results than alternate triangles (George, 1991; Hua et al, 1990; Logan, 2002; Molinari et al, 2001b; Pepper et al, 1992; Rao, 1999). The use of unit aspect ratio elements ensures that preference is not given to any particular direction or variable (Burnett, 1987; Molinari et al, 2001b).

3.3.3. Bandwidth

Bandwidth relates to the maximum difference in node numbers within an element. Thus the bandwidth concept is directly linked to the global numbering of nodes in a mesh (Cook, 1995; Cook et al, 2002; Dickin et al, 1996a; Logan, 2002; Pepper et al, 1992; Rao, 1999; Zienkiewicz et al, 2000). The node numbering scheme impacts directly on the memory space used for storing nodes, which indirectly impacts on the convergence time of the algorithm.

The formulae for bandwidth can be written in terms of degrees of freedom and maximum difference of nodes (Logan, 2002; Rao, 1999):

$$B = \{D + 1\} * f \tag{3.1}$$

where B = Band Width,

D = Maximum difference between node numbers in an element,

f = Number of Degrees of Freedom.

The above equation indicates that in order to reduce the bandwidth of a mesh, 'D' must be optimised (Cook, 1995; Dickin et al, 1996a; Pepper et al, 1992; Rao, 1999). This implies that the node-numbering scheme must be

optimised. The author has encountered numerous node-numbering schemes. A common arrangement that constantly emerged in the authors' literature survey was that the numbering of nodes is conducted in a counter clockwise direction (Kwon et al, 2000; Pepper et al, 1992; Rao, 1999). Two popular node-numbering schemes, which can minimise bandwidth, are:

i. Tessellation

ii. Quadtree

The reader is referred to George (1991), Rao (1999) for further information on the above methods. Bandwidth optimisation can also be implemented when the finite element matrix is assembled. For more information on this, the reader is referred to the following reference Dickin et al (1996a).

3.3.4. Higher Order Elements

Higher order elements are elements that have extra nodes. Examples of higher order elements are illustrated below.

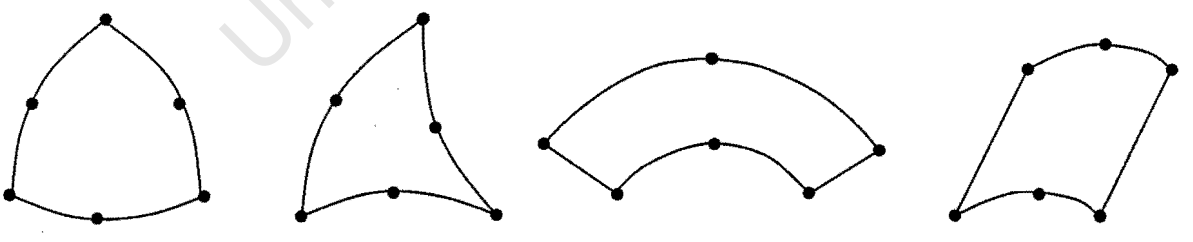


Fig 3.8: Two-Dimensional Higher Order Elements

Higher order elements are curved in nature and can be used to represent curved boundaries (Rao, 1999). Making use of higher order elements to accurately discretise the boundary reduces the error of the solution

(Kolehmainen et al, 1997a; Molinari et al, 2001b; Rao, 1999). However, higher order elements require more nodes on their boundaries than linear elements. This implies that more computation time as well as memory storage is required for higher order elements. Thus a trade off between accuracy versus speed and memory has to be decided on.

For further detailed information about meshes regarding requirements and optimisations the reader is encouraged to look up the following three references (Molinari et al, 2001b; Thompson et al 1999).

4. FINITE ELEMENT MODEL FOR 2D AND 3D ERT

The finite element method seeks a simplified solution to a complicated problem (Dickin et al, 1996a; George, 1991; Rao, 1999; Silvester et al, 1983; Pepper et al, 1992). The technique involves the simplification of a problem resulting in an approximation to the exact solution. The accuracy of the solution depends largely on the factors of the mesh elements discussed in chapter three.

The finite element method can be differentiated into four distinct phases namely:

i. Discretisation of the domain.

ii. Choosing an interpolation function for elements.

iii. Assembling the Element Stiffness Matrix and.

iv. Solving the equation for the unknown field variable.

Phase i. (Discretisation of the domain) has been discussed in Chapter 3. This chapter will deal with points ii. to iv. which are the underlying concepts of the finite element method.

4.1. CHOICE OF FINITE ELEMENT METHOD

The partial differential equations shown in Chapter 2 are required to be solved. These partial differential equations can be solved by either analytical or numerical techniques. However analytical techniques are more tedious to solve than numerical methods (Kleinermann et al, 2001). Numerical techniques that can be used are the finite element method, the finite difference method or the boundary element method. The author has noticed from his literature survey

that the finite element method is the most widely used technique. The finite element method is preferred to alternate techniques because (Breckon, 1990; Cook et al, 2002; Dickin et al, 1996a; Tarvainen et al, 2001):

i. It can model complex geometric elements and irregular boundaries.

ii. Size of elements within the domain can be altered.

iii. Varying the size of elements enables the field variables to be calculated with greater accuracy.

iv. As the domain complexity increases the finite element method converges to a solution faster than alternate techniques.

Due to the vast literature of ERT based on the finite element method, the author with approval from his supervisor (Wilkinson, 20001) has decided to implement the same technique. For further information on analytical solutions the reader is referred to Kleinermann et al (1997), Kleinermann et al (2000), and Kleinermann et al (2001). Solution by the boundary element method can be found in Tarvainen et al (2001).

4.2. CONCEPT OF FINITE ELEMENT METHOD

As mentioned before the finite element method takes a large problem and divides it into a set of simpler problems. The finite element method provides a piece-wise polynomial approximation which is illustrated below (Breckon, 1990; Cook, 1995; Dickin et al, 1996a; Zienkiewicz et al, 2000).

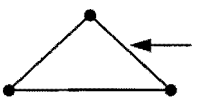


Fig 4.1: Graphical Illustration of Finite Element Method

By piece-wise polynomial it is implied that over the element the field variable is interpolated from the field variable at the nodes (Cook, 1995). In respect to ERT this can be interpreted as: the voltages at the nodes are used to approximate the voltage over the element.

The finite element method concept can be easily explained by means of the following example where the circumference of the circle is sought.

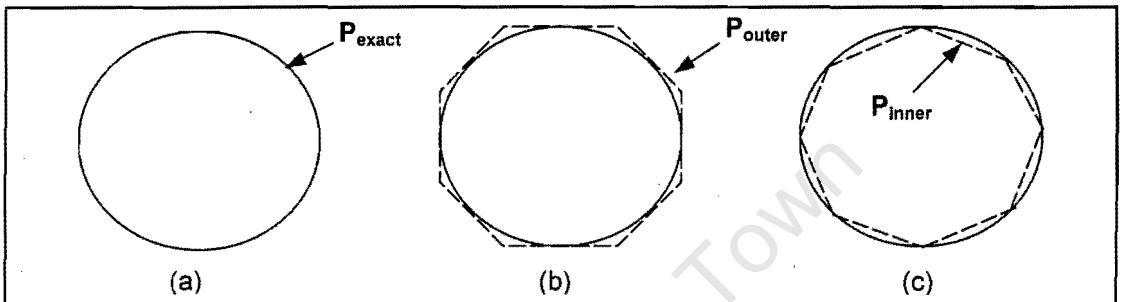


Fig 4.2: Approximating the Circumference of a Circle

In Figure 4.2 the exact solution of the perimeter of the circle is denoted as P_{exact}, which is worldly renown as $2 \bullet \pi \bullet r$, where r is the radius. In order to approximate the perimeter the use of linear polygons as illustrated in Figures 4.2b and 4.2c are made. P_{outer} is the perimeter of the polygon outside of the circle and similarly P_{inner} is the inner polygon perimeter. These perimeters are close approximations of the true solution. As the number of sides of the polygon increases the accuracy of the perimeter increases (Rao, 1999). Thus

$$\lim_{s \to \infty} P_{outer} = \lim_{s \to \infty} P_{inner} = P_{exact} \qquad (4.1)$$

where s is the number of sides of the polygon.

Drawing on a comparison with the finite element method, each side of the polygon can represent an element. Some of the principles discussed in Chapter 3 can now be realised with this example, namely:

i. As the number of sides (elements) of the polygon increases, the number of computations increases.

ii. As the number of computations increase the time to reach the solution increases.

iii. As the number of sides (elements) of the polygon increases, the accuracy of the solution converges to the exact solution.

4.3. INTERPOLATION FUNCTIONS

As previously pointed out in chapter three there are various element shapes that can be used to discretise a domain. The discretisation process converts the continuous problem into a problem with a finite number of unknowns. Solution of these unknown values is approximated by a function within each element (Cook et al, 2002; Hua et al, 1990; Logan, 2002; Rao, 1999; Zienkiewicz et al, 2000). These functions are referred to as:

i. Interpolation Models or,

ii. Approximating Functions or,

iii. Interpolation Functions (most commonly used).

Interpolation functions are defined in terms of the values of the field variable at the nodes of the element (Hua et al, 1990). The author has noticed from his research that polynomial based functions are frequently used for the choice of interpolation functions for the following reasons (Dickin et al, 1996a; Hua et al, 1990; Rao, 1999):

i. Integration and differentiation are easily completed.

ii. The accuracy of the solution can be improved by increasing the degree of the polynomial. This is illustrated below.

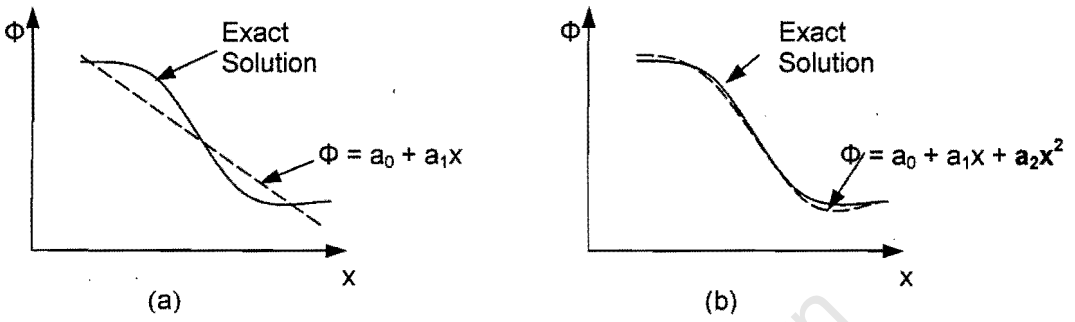


Fig 4.3: Linear and Polynomial Quadratic Interpolation Functions

Figure 4.3 illustrates the idea that as the degree of the interpolation function (Φ) increases, the trial solution approaches the exact solution. For this thesis linear polynomials in two-dimensional and three-dimensional space will be discussed. Further the elements under consideration will be triangles and tetrahedrons.

4.3.1. Two-Dimensional Linear Interpolation Functions for Triangles

The author would like to acknowledge that most of the work in this section is extracted from the following references Cook (1995), Cook (2002), Dickin et al (1996a), Hua et al (1990), Kwon et al (2000), Logan (2002), Pepper et al (1992), Rao (1999), Silvester et al (1983), Zienkiewicz (2000).

The two-dimensional triangular element is represented by the following linear polynomial:

$$V(x,y) = \alpha_1 + \alpha_2 x + \alpha_3 y \tag{4.2}$$

where V is the voltage in the element,

x and y represent coordinates in the domain,

α_n are constant polynomial coefficients.

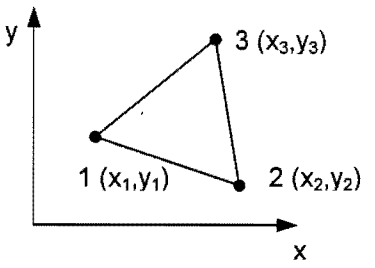


Fig 4.4: Two-Dimensional Linear Triangular Element

Figure 4.4 depicts a two-dimensional triangle with three nodes labelled 1, 2, and 3. Attention is drawn to the fact that ordering is completed in a counter clockwise direction for conformity.

At node 1 ($x = x_1$, $y = y_1$):

$$V = V_1 = \alpha_1 + \alpha_2 x_1 + \alpha_3 y_1 \tag{4.3}$$

And similarly at node 2 ($x = x_2$, $y = y_2$):

$$V = V_2 = \alpha_1 + \alpha_2 x_2 + \alpha_3 y_2 \tag{4.4}$$

And similarly at node 3 ($x = x_3$, $y = y_3$):

$$V = V_3 = \alpha_1 + \alpha_2 x_3 + \alpha_3 y_3 \tag{4.5}$$

By reduction and manipulation of the above equations the following equations can be attained:

$$\alpha_1 = \frac{(a_1V_1 + a_2V_2 + a_3V_3)}{2 \bullet \text{Area}} \tag{4.6}$$

$$\alpha_2 = \frac{(b_1V_1 + b_2V_2 + b_3V_3)}{2 \bullet \text{Area}} \tag{4.7}$$

$$\alpha_3 = \frac{(c_1V_1 + c_2V_2 + c_3V_3)}{2 \bullet \text{Area}} \tag{4.8}$$

where

$$\text{Area} = \tfrac{1}{2} \bullet \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} \tag{4.10}$$

$$a_1 = \begin{vmatrix} x_2 & y_2 \\ x_3 & y_3 \end{vmatrix} \qquad a_2 = \begin{vmatrix} x_3 & y_3 \\ x_1 & y_1 \end{vmatrix} \qquad a_3 = \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} \tag{4.11}$$

$$b_1 = -\begin{vmatrix} 1 & y_2 \\ 1 & y_3 \end{vmatrix} \qquad b_2 = -\begin{vmatrix} 1 & y_3 \\ 1 & y_1 \end{vmatrix} \qquad b_3 = -\begin{vmatrix} 1 & y_1 \\ 1 & y_2 \end{vmatrix} \tag{4.12}$$

$$c_1 = -\begin{vmatrix} x_2 & 1 \\ x_3 & 1 \end{vmatrix} \qquad c_2 = -\begin{vmatrix} x_3 & 1 \\ x_1 & 1 \end{vmatrix} \qquad c_3 = -\begin{vmatrix} x_1 & 1 \\ x_2 & 1 \end{vmatrix} \tag{4.13}$$

Thus

$$V(x,y) = V_1H_1(x,y) + V_2H_2(x,y) + V_3H_3(x,y) \tag{4.14}$$

where H is the dimensionless element shape function expressed as

$$H_n(x,y) = \frac{(a_n + b_n x + c_n y)}{2 \bullet \text{Area}} \qquad \text{for } n = 1, 2, 3. \tag{4.15}$$

Note that H_n has the value 1 at the n^{th} node and 0 elsewhere.

4.3.2. Three-Dimensional Linear Interpolation Functions for Tetrahedral Elements

The author would like to acknowledge that the work in this section is extracted from the following references Cook (1995), Cook et al (2002), Kwon et al (2000), Logan (2002), Pepper et al (1992), Rao (1999), Silvester et al (1983), Zienkiewicz et al (2000).

The three-dimensional tetrahedral element is represented by the following linear polynomial:

$$V(x,y,z) = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z \tag{4.16}$$

where V is the voltage in the element,

x, y and z represent coordinates in the domain,

α_n are constant polynomial coefficients.

A model for a tetrahedral is depicted in the following figure located on the next page.

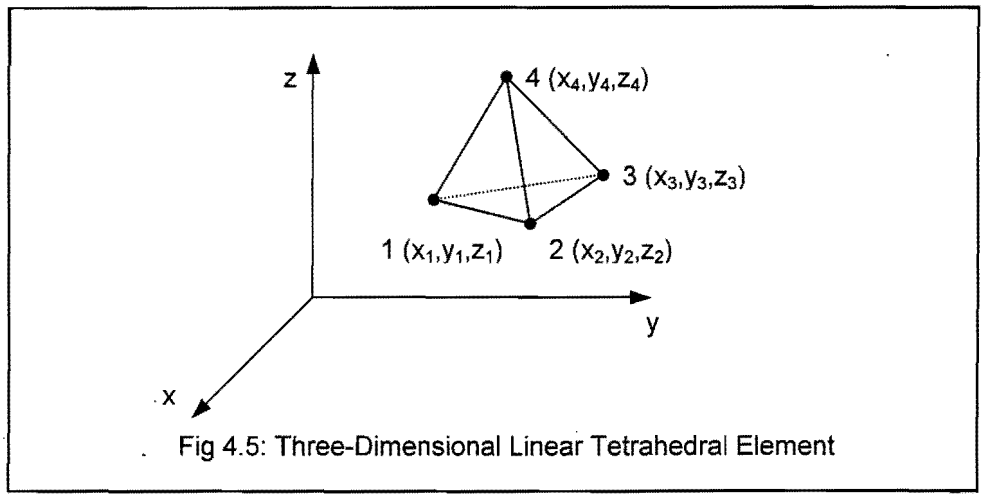


Fig 4.5: Three-Dimensional Linear Tetrahedral Element

Figure 4.5 depicts a three-dimensional tetrahedron with four nodes labelled 1, 2, 3 and 4. Attention is drawn to the fact that node numbering is completed in a counter clockwise direction starting at the base for conformity.

At node 1 ($x = x_1$, $y = y_1$, $z = z_1$):

$$V = V_1 = \alpha_1 + \alpha_2 x_1 + \alpha_3 y_1 + \alpha_4 z_1 \tag{4.17}$$

And similarly at node 2 ($x = x_2$, $y = y_2$, $z = z_2$):

$$V = V_2 = \alpha_1 + \alpha_2 x_2 + \alpha_3 y_2 + \alpha_4 z_2 \tag{4.18}$$

And similarly at node 3 ($x = x_3$, $y = y_3$, $z = z_3$):

$$V = V_3 = \alpha_1 + \alpha_2 x_3 + \alpha_3 y_3 + \alpha_4 z_3 \tag{4.19}$$

And similarly at node 4 ($x = x_4$, $y = y_4$, $z = z_4$):

$$V = V_4 = \alpha_1 + \alpha_2 x_4 + \alpha_3 y_4 + \alpha_4 z_4 \tag{4.20}$$

By reduction and manipulation of the above equations the following equations can be attained:

$$\alpha_1 = \frac{(a_1V_1 + a_2V_2 + a_3V_3 + a_4V_4)}{6 \bullet \text{Volume}} \tag{4.21}$$

$$\alpha_2 = \frac{(b_1V_1 + b_2V_2 + b_3V_3 + b_4V_4)}{6 \bullet \text{Volume}} \tag{4.22}$$

$$\alpha_3 = \frac{(c_1V_1 + c_2V_2 + c_3V_3 + c_4V_4)}{6 \bullet \text{Volume}} \tag{4.23}$$

$$\alpha_4 = \frac{(d_1V_1 + d_2V_2 + d_3V_3 + d_4V_4)}{6 \bullet \text{Volume}} \tag{4.24}$$

where

$$\text{Volume} = \begin{vmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{vmatrix} \tag{4.25}$$

$$a_1 = \begin{vmatrix} x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \end{vmatrix} \quad a_2 = \begin{vmatrix} x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \\ x_1 & y_1 & z_1 \end{vmatrix} \quad a_3 = \ldots \quad a_4 = \ldots \tag{4.26}$$

$$b_1 = - \begin{vmatrix} 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \\ 1 & y_4 & z_4 \end{vmatrix} \quad b_2 = - \begin{vmatrix} 1 & y_3 & z_3 \\ 1 & y_4 & z_4 \\ 1 & y_1 & z_1 \end{vmatrix} \quad b_3 = \ldots \quad b_4 = \ldots \tag{4.27}$$

$$c_1 = -\begin{vmatrix} x_2 & 1 & z_2 \\ x_3 & 1 & z_3 \\ x_4 & 1 & z_4 \end{vmatrix} \quad c_2 = -\begin{vmatrix} x_3 & 1 & z_3 \\ x_4 & 1 & z_4 \\ x_1 & 1 & z_1 \end{vmatrix} \quad c_3 = \ldots \quad c_4 = \ldots \tag{4.28}$$

$$d_1 = -\begin{vmatrix} x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \\ x_4 & y_4 & 1 \end{vmatrix} \quad d_2 = -\begin{vmatrix} x_3 & y_3 & 1 \\ x_4 & y_4 & 1 \\ x_1 & y_1 & 1 \end{vmatrix} \quad d_3 = \ldots \quad d_4 = \ldots \tag{4.29}$$

Thus

$$V(x,y,z) = V_1H_1(x,y,z) + V_2H_2(x,y,z) + V_3H_3(x,y,z) + V_4H_4(x,y,z) \tag{4.30}$$

where H is the dimensionless element shape function expressed as:

$$H_n(x,y) = \frac{(a_n + b_nx + c_ny + d_ny)}{6 \bullet \text{Volume}} \quad \text{for } n = 1, 2, 3, 4. \tag{4.31}$$

Note that H_n has the value 1 at the n^{th} node and 0 elsewhere.

Now that the interpolation functions are specified, the stiffness matrix can be assembled.

4.4. ASSEMBLING THE LINEAR MATRIX EQUATIONS

Once the interpolation functions have been chosen for the elements, the information now has to be stored in a matrix. The author would like to acknowledge that the work in this section is extracted from the following references Cook (1995), Cook et al (2002), Kwon et al (2000), Logan (2002), Pepper et al (1992), Rao (1999), Silvester et al (1983), Zienkiewicz et al (2000).

The finite element method tries to set up a linear matrix equation over the whole domain in the form

$$Y \bullet V = I \tag{4.32}$$

where Y is the Global Master/Stiffness Matrix with size n x n.

V is the Field Variable or Voltage Matrix to be solved of size n x p.

I is the Load or Current Matrix of size n x p.

n is the number of nodes within the mesh.

p is the number of current injection patterns.

Matrix Y is assembled by all element matrices 'y' over the domain.

4.4.1. Assembling the 2D Element Matrix 'y'

From the interpolation function given for the triangular element in section 4.3, the 3 x 3 element matrix y_{ij} is given by

$$y_{iij} = \sigma \bullet \iint_A \left(\frac{\partial H_i}{\partial x}\frac{\partial H_j}{\partial x} + \frac{\partial H_j}{\partial y}\frac{\partial H_j}{\partial y} \right) dA \quad \text{for i,j = 1,2,3.} \tag{4.33}$$

where A and dA denote the surface integral of the element.

σ is the conductivity that is constant within the element.

Substitution of Equation 4.31 into Equation 4.33 combined with a bit of manipulation results in the following element matrix representation for the linear triangular element shown in Figure 4.4:

$$y = \frac{\sigma}{4 \bullet \text{Area}} \begin{bmatrix} b_1^2 + c_1^2 & b_1b_2 + c_1c_2 & b_1b_3 + c_1c_3 \\ b_1b_2 + c_1c_2 & b_2^2 + c_2^2 & b_2b_3 + c_2c_3 \\ b_1b_3 + c_1c_3 & b_2b_3 + c_2c_3 & b_3^2 + c_3^2 \end{bmatrix} \tag{4.34}$$

For the derivation of Equation 4.34 from Equation 4.33 the reader is referred to the Index C. The attention of the reader is drawn to the important fact that the above element matrix y is symmetric. This matrix must be compiled for each and every element within the domain. All these matrices are then assembled into a global master matrix usually denoted by a capital Y (Dickin et al, 1996a; Hua et al, 1990).

4.4.2. Assembling the 3D Element Matrix 'y'

From the interpolation function given for the tetrahedral element in Chapter 4.3, the 4 x 4 element matrix y_{ij} is given by

$$y_{ij} = \sigma \bullet \iiint_V \left(\frac{\partial H_i}{\partial x}\frac{\partial H_j}{\partial x}\frac{\partial H_k}{\partial x} + \frac{\partial H_i}{\partial y}\frac{\partial H_j}{\partial y}\frac{\partial H_k}{\partial y} + \frac{\partial H_i}{\partial z}\frac{\partial H_j}{\partial z}\frac{\partial H_k}{\partial z} \right) dV \qquad (4.35)$$

for i,j,k = 1,2,3,4.

where V and dV denote the volume integral of the element.

σ is the conductivity that is constant within the element.

Substitution of Equation 4.15 into Equation 4.35 combined with a bit of manipulation results in the following element matrix representation for the linear triangular element shown in Figure 4.5:

$$y = \frac{\sigma}{6 \bullet \text{Volume}} \begin{bmatrix} (b_1^2 + c_1^2 + d_1^2) & (b_1b_2 + c_1c_2 + d_1d_2) & (b_1b_3 + c_1c_3 + d_1d_3) & (b_1b_4 + c_1c_4 + d_1d_4) \\ (b_1b_2 + c_1c_2 + d_1d_2) & (b_2^2 + c_2^2 + d_2^2) & (b_2b_3 + c_2c_3 + d_2d_3) & (b_2b_4 + c_2c_4 + d_2d_4) \\ (b_1b_3 + c_1c_3 + d_1d_3) & (b_2b_3 + c_2c_3 + d_2d_3) & (b_3^2 + c_3^2 + d_3^2) & (b_3b_4 + c_3c_4 + d_3d_4) \\ (b_1b_4 + c_1c_4 + d_1d_4) & (b_2b_4 + c_2c_4 + d_2d_4) & (b_3b_4 + c_3c_4 + d_3d_4) & (b_4^2 + c_4^2 + d_4^2) \end{bmatrix}$$

(4.36)

To gauge how Equation 4.36 can be derived from Equation 4.35 the reader is referred to Index C. The attention of the reader is drawn to important fact that the above element matrix y is symmetric. This matrix must be compiled for each and every element within the domain. All these matrices are then assembled into a global master matrix usually denoted by a capital Y (Dickin et al, 1996a; Hua et al, 1990).

4.4.3. Assembling the Global Master Matrix 'Y' for 2D and 3D

The principles behind assembling the global master matrix for 2D and 3D are essentially the same, thus this subsection will cover the slightly more complex assemblage for three-dimensional tetrahedral elements. An efficient way of describing the assemblage of the global master matrix is by presenting the pseudo-code and flow chart diagrams. The following figure aids in explaining the pseudo-code that follows.

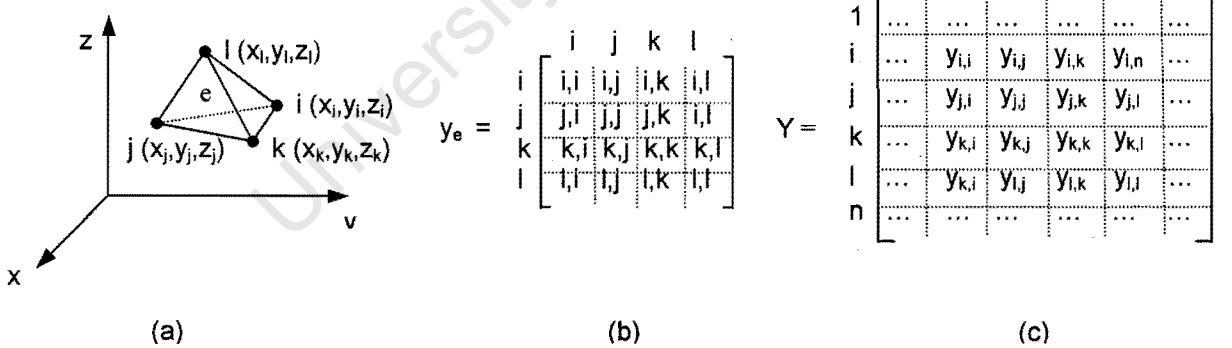


Fig 4.6: Assembling the Global Master Matrix for 3D Tetrahedral Element

for e = 1 to num_elements

- Assemble the element matrix for element 'e' with method in Chapter 4.4.2 as shown in Figure 4.6b.
- Element 'e' contains the nodes i, j, k and l as global node numbers as depicted in Figure 4.6a.

- Find the corresponding i, j, k, l rows and columns in the global master matrix.
- Add the entries of the i, j, k, l rows and columns in the element matrix y_e to the current i, j, k, l entries in the global master matrix Y.

end for loop

A flow chart of the above pseudo-code is depicted below.

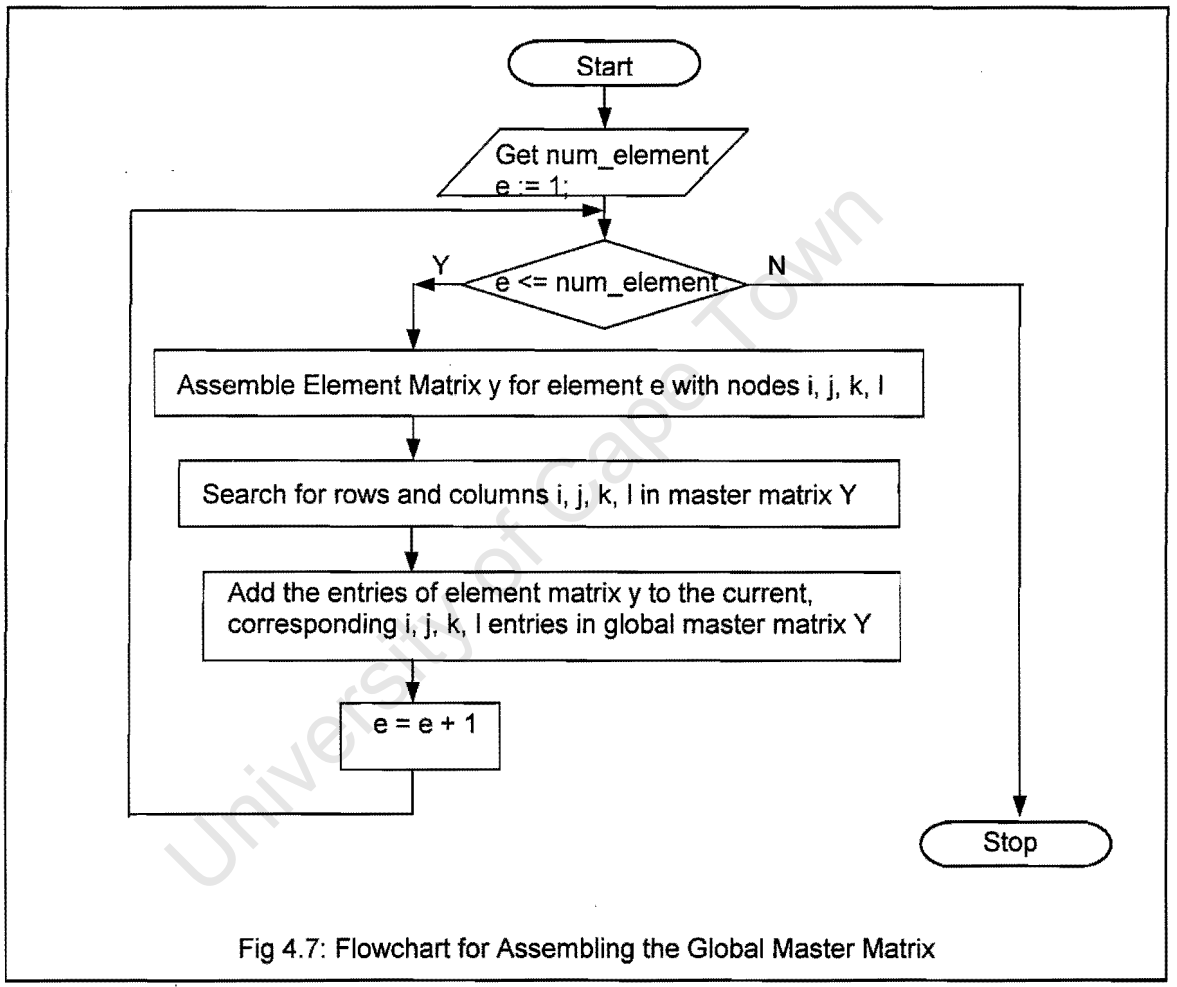


Fig 4.7: Flowchart for Assembling the Global Master Matrix

Figure 4.7 shows the importance of a global node numbering system that is easily accessible for compiling the Y matrix.

4.4.4. Assembling the Load/Current Matrix 'I'

The assembly of the load matrix 'I' requires less computation than the assembly of the global master matrix Y. The size of the load matrix 'I' is 'n x p', where 'n' is the number of nodes in the mesh and 'p' is the number of current projection patterns. The rows represent the nodes of the mesh and the columns represent the current injection patterns. All boundary input is fed into the 'I' matrix at the appropriate node. For example if there is a current flow on the boundary at a particular node, the value of the current is inserted into the 'I' matrix at that respective node. Zeros are inserted at the remaining row entries in the load matrix for that current injection pattern.

4.5. REFERENCING THE LINEAR MATRIX EQUATIONS

In order to solve the linear equations the global master matrix and the load matrix must be referenced. The field variable matrix 'V' does not have to be referenced. By referencing it is implied that a ground voltage point needs to be created from which other voltages can be referred to. The following subsections explain the process of referencing the global master matrix and load matrix.

4.5.1. Referencing the Global Master Matrix 'Y*'

After assembling the Y matrix it is of the rank (n-1). The global master matrix Y is singular implying that its inverse does not exit thus the linear set of equations cannot be solved (Rao, 1999). The global master matrix must be modified and the rank changed to n. This is achieved by referencing one of the nodes (Dickin et al, 1996a; Hua et al, 1990), which usually is the central node of the domain.

For clarity, let 'm' be the centrally located node in the domain. The referencing is reflected in the global master matrix by setting the m^{th} row and column in the Y matrix to 1. All corresponding rows and columns of the m^{th} entry in the matrix are set to zero (Dickin et al, 1996a; Hua et al, 1990). This concept is illustrated below.

$$Y^* = \begin{array}{c} \\ 1 \\ 2\ldots \\ m.. \\ n\text{-}1 \\ n \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3\ldots & m\ldots & n\text{-}1 & n \end{array} \\ \begin{bmatrix} & & & 0 & & \\ & & & 0 & & \\ 0 & 0 & 0 & 1 & 0 & 0 \\ & & & 0 & & \\ & & & 0 & & \end{bmatrix} \end{array} \qquad (4.37)$$

The global master matrix Y is now referred to as the modified global master matrix and is denoted as Y^*. The remaining entries of the master matrix are unaltered. Important characteristics of the modified global master matrix Y^* are (Dickin et al, 1996a; Hua et al, 1990):

i. It is symmetric i.e. $Y_{i,j} = Y_{j,i}$.

ii. It is greatly sparse i.e. a large amount of its entries are zero.

iii. It has a rank of n.

iv. It is positive definite i.e.

$$x^T \bullet Y \bullet x = \begin{cases} \geq 0 \text{ for } x \neq 0 \\ = 0 \text{ for } x = 0 \end{cases} \qquad (4.38)$$

where x is an 'n' sized vector,

T denotes transpose.

4.5.2. Referencing the Load Matrix 'I*'

Since a modification was made on the global master matrix Y^*, a similar modification on the load matrix I must be made. Thus all entries in the m^{th} row of the load matrix 'I' is set to 0, where 'm' is the same middle node as defined earlier.

4.6. SOLVING THE FINITE ELEMENT EQUATION

The linear set of equations established can be solved in three ways. Firstly it can be solved inversely i.e. since

$$Y \bullet V = I$$

then

$$V = Y^{-1} \bullet I \qquad (4.39)$$

However if Y is a large matrix then the computation difficulty escalates resulting in a computer halt (Pepper et al, 1992). Thus this method is not recommended. Alternatively it can be solved iteratively or directly (Borse, 1997; Dickin et al, 1996a; Hua et al, 1990; Pepper et al, 1992; Rao, 1999). The following subsections elaborate on these two methods.

4.6.1. Iterative Methods

Iterative methods involve guessing an initial answer, which is then successively updated until a better approximation is attained (Dickin et al, 1996a; Hua et al, 1990). Some of the iterative methods that can be used are:

i. Gauss-Seidel.

ii. Successive Overrelaxation (SOR).

iii. Conjugate Gradient Method.

Details of these methods will not be presented here but the reader is pointed to the following references Atkinson (1985), Chapra et al (1988), Gambolati (1980), Hageman et al (1981). The reader is also made aware of the following references Molinari et al (2001a), Molinari et al (2002), Wang (2001), which present papers for solution of electrical tomography based on a conjugate gradient method.

Table 4.1: List of Advantages and Disadvantages of Iterative Numerical Methods

Advantages	**Disadvantages**
Faster than direct methods for non-sparse matrices.	Can become numerically unstable.
Minimisation techniques can be applied decreasing computation time.	Cannot use factorisation methods for sparse matrices.
Efficient for large matrices.	Is not efficient for sparse matrices.
Insensitive to rounding off and truncation errors thus results are accurate.	Although simple large amount of iterations may be need for accurate solution.
Simple and uniform method.	

Table 4.1 compares the advantages and disadvantages of the iterative numerical method that were extracted from the following sources: Dickin et al (1996a), Hua et al (1990), Rao (1990). Clearly the iterative method is advantageous if the 'Y^*' matrix is large and non-sparse which tends to be the case in ERT.

4.6.2. Direct Methods

Direct methods yield the exact solution in a finite number of algebraic operations (Dickin et al, 1996a; Hua et al, 1990). This is true provided that there is no rounding-off errors or errors of other means. Some of the iterative methods that can be used are:

i. LU Decomposition.

ii. Cholesky Factorisation.

The mathematical details behind the above two methods will not be presented here. The reader is encouraged to read the following references for a mathematical perspective Atkinson (1985), Borse (1997).

Table 4.2: List of Advantages and Disadvantages of Direct Numerical Methods

Advantages	Disadvantages
Numerically stable.	Inefficient for large matrices.
Factorisation methods can be used.	Slow computation time for non-sparse matrices.
Efficient for sparse matrices.	Minimisation techniques cannot be applied to decrease computation time.
Small finite number of iterations to solve equation.	Due to rounding off and truncation, results can be error prone or useless.
Predefined built-in functions in Matlab.	

Table 4.2 compares the advantages and disadvantages of the direct numerical method that were extracted from the following sources: Borse (1997), Dickin et al (1996a), Hua et al (1990), Rao (1990). Clearly the direct method is advantageous if the 'Y^*' matrix is highly sparse and if a minimal amount of updates are required (Dickin et al, 1996a; Hua et al, 1990).

Please note for brevity and clarity the 'Y^*' matrix will be referred to as 'Y' matrix. Further, the author would like to acknowledge the following references from which the equations in the following subsections were extracted: Atkinson (1985), Borse (1997), Dickin et al (1996a), Hua et al (1990), Pepper et al (1992).

i. LU Decomposition

For LU decomposition the Y matrix is split as a product of two matrices i.e.

$$Y = L \bullet U \tag{4.40}$$

where L is the lower triangle.

U is the upper triangle.

Substituting the above into the linear equation produces:

$$(L \bullet U) \bullet V = I \tag{4.41}$$

Rearranging yields

$$L \bullet (U \bullet V) = I \tag{4.42}$$

Making use of the substitution $x = U \bullet V$ replaces the above equation with:

$$L \bullet x = I \tag{4.43}$$

Thus solving for x yields:

$$x = L^{-1} \bullet I \tag{4.44}$$

Finally the value of x is substituted into the following equation to solve for the required filed variable V:

$$V = U^{-1} \bullet x \tag{4.45}$$

The LU decomposition is advantageous in that the solution of the triangular sets is trivial (Dickin et al, 1996a).

ii. Cholesky Factorisation

The Cholesky factorisation is very similar to the LU decomposition. It is well suited for symmetric positive definite matrices such as Y. Initially the Y matrix is split as a product of two matrices i.e.

$$Y = L \bullet L^T \tag{4.46}$$

where L is the lower triangle with positive diagonal elements.

L^T is the transpose of L.

The following is analogous to LU decomposition and is shown for completeness. Substituting the above into the linear equation produces:

$$(L \bullet L^T) \bullet V = I \tag{4.47}$$

Rearranging yields

$$L \bullet (L^T \bullet V) = I \tag{4.48}$$

Making use of the substitution $x = L^T \bullet V$ replaces the above equation with:

$$L \bullet x = I \tag{4.49}$$

This is commonly known as the forward substitution. Thus solving for x yields:

$$x = L^{-1} \bullet I \tag{4.50}$$

Finally the value of x is substituted into the following equation to solve for the required filed variable V:

$$V = (L^T)^{-1} \bullet x \tag{4.51}$$

The Cholesky factorisation is advantageous in that it (Dickin et al, 1996a):

i. Preserves the symmetry.

ii. Only one matrix has to be stored namely L.

iii. Requires less computation than LU decomposition.

The author is favourable to the use of direct methods over iterative methods. The primary reasoning behind this choice is for the predefined built-in functions that are present in many programming languages such as *Matlab*.

5. IMAGE RECONSTRUCTION FOR 2D AND 3D ERT

In this chapter the techniques required for reconstruction will be discussed.

5.1. TYPES OF RECONSTRUCTION ALGORITHMS

As the interest in electrical tomography grew over the years there were concerted efforts to develop more robust and faster algorithms. Some of techniques that have been successfully implemented in ERT are:

i. Compensation (Sensitivity) Theorem

ii. Double Constraint Method (Variational)

iii. Kalman Filter

iv. Linear Back Projection

v. Neural Networks

vi. Newton-Raphson

vii. Perturbation Method

viii. Statistical Methods

Each method has its own characteristic advantages. Some of these features are:

i. Faster algorithm.

ii. More accurate solution.

iii. Ability to image dynamic movements in tank.

iv. Robust algorithm that does not diverge.

v. Memory storage requirement.

The most popularly used technique is the Newton-Raphson algorithm due to its robustness and accuracy. Back projection algorithms due to their use in other tomography systems were initially considered the method of choice. Over the years this algorithm proved less reliable for ERT and is now all but phased out. Recently neural networks have gained more focus due to its speed and are now widely researched. After consultation with the authors supervisor Dr. Wilkinson (2002) an outcome was reached to implement the Newton-Raphson algorithm. The following sections explain the principles behind this algorithm.

5.2. FUNDAMENTALS OF NEWTON-RAPHSON ALGORITHM

The Newton-Raphson Algorithm is an iterative algorithm developed for non-linear problems (Hua et al, 1990). Its aim is to minimise an objective function. At this stage we should define an objective function or more specifically an error function that can be minimised. In ERT the error is considered as the difference between the real conductivity and the calculated conductivity. However the real conductivity is difficult (if not impossible) to measure at various points in the tank.

A more indirect approach is to take the error as the difference between the measured boundary voltages and computed boundary voltages (Hua et al, 1990). Yorkey (1986) defined the error function as the equally weighted mean square difference between the measured and calculated boundary voltage. The following information is adapted from Hua et al (1990), and Yorkey et al, (1987).

$$\varepsilon(\sigma) = 0.5 \bullet [V_{cal}(\sigma) - V_{mea}]^T \bullet [V_{cal}(\sigma) - V_{mea}] \quad (5.1)$$

where ε is the Objective Error Function.

T is the Matrix Transpose.

V_{cal} is the Calculated Boundary Voltage.

V_{mea} is the Measured Boundary Voltage.

Equation 5.1 is minimised and equated to zero to solve for σ.

$$\varepsilon'(\sigma) = [V_{cal}'(\sigma)]^T \bullet [V_{cal}(\sigma) - V_{mea}] = 0 \quad (5.2)$$

where $V_{cal}'(\sigma)$ is the Jacobian Matrix.

For reasons of clarity and brevity the Jacobian Matrix $V_{cal}'(\sigma)$ will be represented as J. Computation of the Jacobian Matrix is explained in the next subsection. Expanding Equation 5.2 by taking the Taylor series expansion at the point σ^l results in:

$$\varepsilon'(\sigma^{i+1}) = \varepsilon'(\sigma^i) + \varepsilon''(\sigma^i) \bullet \Delta\sigma^l = 0 \quad (5.3)$$

where

$$\sigma^{i+1} = \sigma^i + \Delta\sigma^i \quad (5.4)$$

In Equation 5.3 ε'' is known as the Hessian Matrix and is expressed as

$$\varepsilon'' = \{ J^T \bullet J \} + \{ [J']^T \bullet (I \otimes [V_{cal} - V_{mea}]) \} \quad (5.5)$$

where $\otimes$ is the Kronecker Matrix Product.

In Equation 5.5 J′ is insignificantly small compared to $J^T \bullet J$, therefore Equation 5.5 can be modified to the following:

$$\varepsilon'' = J^T \bullet J \tag{5.6}$$

Substituting Equations 5.2 and 5.6 into Equation 5.3 yields

$$\varepsilon'(\sigma^{i+1}) = \{ J^T \bullet [V_{cal}(\sigma) - V_{mea}] \} + \{ J^T \bullet J \bullet \Delta\sigma^i \} \tag{5.7}$$

Rearranging Equation 5.7 for the changing conductivity results in:

$$\Delta\sigma^i = \frac{\varepsilon'(\sigma^{i+1}) - (J^T \bullet [V_{cal}(\sigma) - V_{mea}])}{J^T \bullet J} \tag{5.8}$$

Substituting Equation 5.3 into Equation 5.8 produces the final form of the required equation namely:

$$\Delta\sigma^i = \frac{- \{ J^T \bullet [V_{cal}(\sigma) - V_{mea}] \}}{J^T \bullet J} \tag{5.9}$$

The following flowchart on the next page integrates the Newton-Raphson algorithm into ERT.

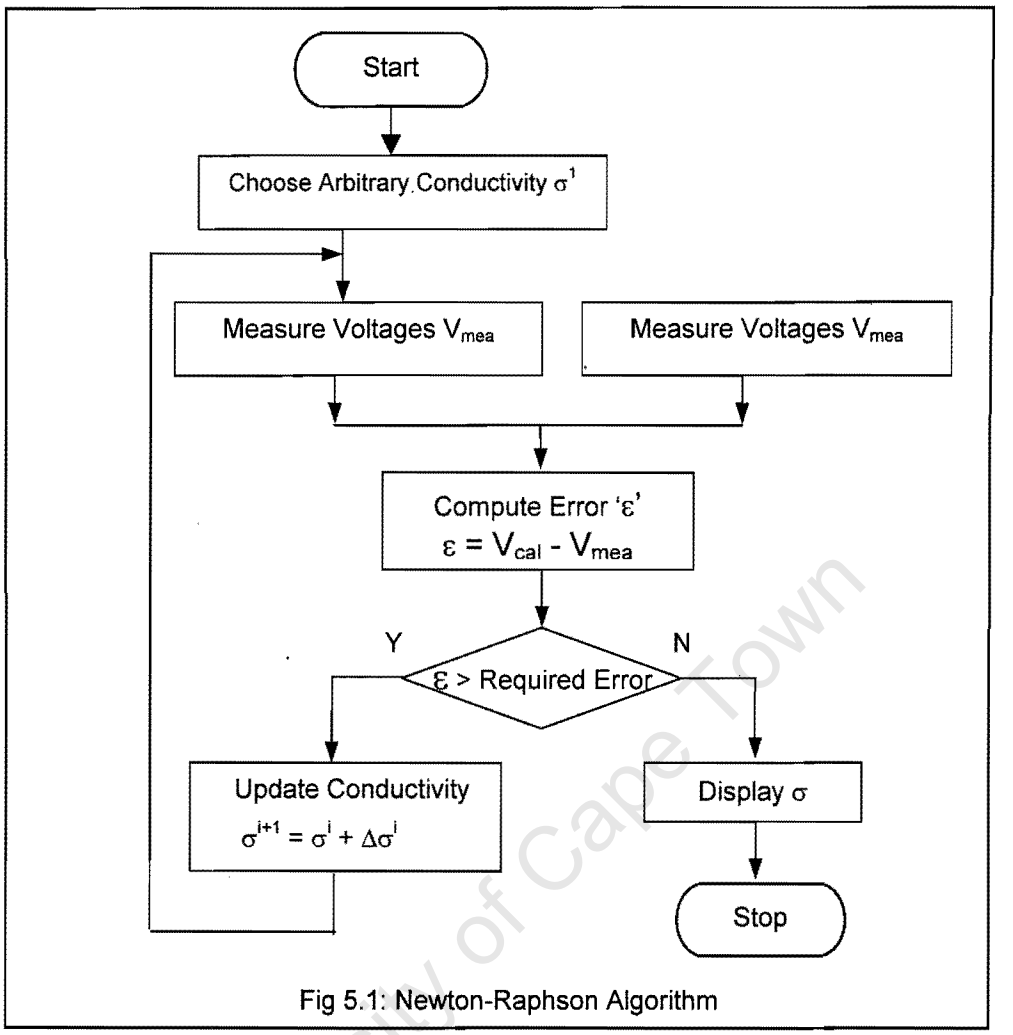


Fig 5.1: Newton-Raphson Algorithm

Figure 5.1 depicts how the Newton Raphson algorithm fits into ERT.

5.3. COMPUTATION OF THE JACOBIAN MATRIX

The Jacobian Matrix which is referred to in the previous subsection, is symbolised as $V_{cal}'(\sigma)$ or J, which can be expanded to

$$V_{cal}'(\sigma) = J = \frac{\partial V_{cal}}{\partial \sigma} \tag{5.10}$$

The above implies that the Jacobian is equal to the change in voltage for a change in conductivity. Referring back to Chapter 4, the set of equations that were assembled are:

$$Y \bullet V_{cal} = I \tag{5.11}$$

where Y is the global master matrix.

V_{cal} is the calculated voltage matrix.

I is the current matrix.

Taking the inverse and rearranging Equation 5.11 yields:

$$V_{cal} = Y^{-1} \bullet I \tag{5.12}$$

Substituting Equation 5.12 into Equation 5.10 results in an equation that is easier to differentiate:

$$J = \frac{\partial (Y^{-1} \bullet I)}{\partial \sigma} \tag{5.13}$$

Differentiating produces:

$$J = -Y^{-1} \bullet \frac{\partial Y}{\partial \sigma} \bullet Y^{-1} \bullet I \tag{5.14}$$

Simplifying the above yields:

$$J = -Y^{-1} \bullet \frac{\partial Y}{\partial \sigma} \bullet V_{cal} \tag{5.15}$$

Equation 5.15 is now in a solvable form. The partial derivative of the global master matrix can be rewritten as:

$$\frac{\partial Y}{\partial \sigma} = \frac{Y}{\sigma+\Delta\sigma} \tag{5.16}$$

where $\Delta\sigma$ is an extremely small value, typically 1% of σ.

5.4. NOSER ALGORITHM

Noser is the acronym for Newton's One-Step Error Reconstructor. As the name implies this algorithm makes use of the Newton-Raphson algorithm. This algorithm was proposed by Simske (1987) and presented in Cheney et al (1990). The algorithm was proposed in an attempt to reduce the time of the algorithm. The map of the Noser algorithm is described below:

Firstly, a tank with a uniform conductivity is specified. The Jacobian Matrix for this homogenous system is computed. The Jacobian Matrix is then written to a file, or stored in memory. When the user initiates the NOSER algorithm, the Jacobian Matrix is not calculated in the inverse solution but rather substituted with the pre-computed Jacobian Matrix that was written to file (Cheney et al, 1990; Le Hyaric et al, 2000; Le Hyaric et al, 2002; Vauhkonen, 1997). Since the Jacobian Matrix is not computed for every iteration, the accuracy of the reconstructed image is reduced but concurrently the solution time is positively reduced allowing for online imaging to be practical.

5.5. REGULARIZATION

The inverse nature of ERT is ill-posed, i.e. small changes of the data would result in large changes in the solution (Vauhkonen, 1997). Regularization seeks to 'penalize' large changes in the solution and filter out noisy data and hence make the solution a better fit. More specifically, the Hessian matrix in Equation 5.3 (Chapter 5.2) is ill-conditioned which is attributed from the non-linear relationship between conductivity changes and boundary voltages. The positive definiteness of the Hessian matrix is further deteriorated due to numerical rounding errors or from choosing a bad initial conductivity distribution (Randall et al, 2001). To implement regularization means that the objective error function has to be modified. Thus Equation 5.1 with regularization can be written as:

$$\varepsilon(\sigma) = ||V_{cal}(\sigma) - V_{mea}||^2 + \beta^l \bullet ||S \bullet \sigma||^2 \qquad (5.17)$$

where ε is the Objective Error Function.

V_{cal} is the Calculated Boundary Voltage.

V_{mea} is the Measured Boundary Voltage.

β is the Regularization Factor or Smoothness Factor.

S is the Regularization Matrix or Smoothness Matrix.

Equation 5.17 can be written in the discretised form below.

$$\varepsilon(\sigma) = 0.5 \bullet [V_{cal}(\sigma) - V_{mea}]^T \bullet [V_{cal}(\sigma) - V_{mea}] + (\beta^l \bullet \sigma^T \bullet S^T \bullet S \bullet \sigma) \qquad (5.18)$$

Various regularisation techniques such as Levenberg-Marquardt, Tikhonov are successfully used in ERT. These two techniques will be discussed below.

5.5.1. Levenberg-Marquardt Regularization

The Levenberg-Marquardt process firstly normalises the Hessian matrix by multiplying all the row elements and then column elements by $\frac{1}{\sqrt{h_{ii}}}$ where h_{ii} are the original diagonal entries of the Hessian matrix. This forces the diagonal elements to unity (Randall et al, 2001). The updating conductivity equation incorporating regularization can thus be expressed as:

$$\Delta\sigma^i = \frac{-\{ J^T \bullet [V_{cal}(\sigma) - V_{mea}] \}^*}{(J^T \bullet J)^* + (\beta^i \bullet S)} \qquad (5.19)$$

where J is the Jacobian Matrix

S is the Identity Matrix for this case.

* means the matrix is normalised.

The value of β changes with every iteration and is based on the Error Function. If the value of the Error function is greater than the previous iteration then β is increased. As the Error gets closer and closer to zero β is decreased to reduce the amount of regularization needed.

5.5.2. Tikhonov Regularization

In Tikhonov regularization a dampening factor is added to the solution to cancel out the noisy errors (Polydorides, 2000). Detailed information on the Tikhonov regularization can be found in the following references: Golub et al (1999), Kolehmainen et al (1997b), Vauhkonen et al (1998b). The updating conductivity equation incorporating Tikhonov regularization can thus be expressed as:

$$\Delta\sigma^i = \frac{-\{ J^T \bullet [V_{cal}(\sigma) - V_{mea}] - (\beta^i \bullet S^T \bullet S \bullet \sigma^i) \}}{(J^T \bullet J) + (\beta^i \bullet S^T \bullet S \bullet \sigma^i)} \quad (5.20)$$

where S is the Smoothing Matrix or Regularization Matrix.

The Smoothing Matrix S above can be constructed by various means. The most common means is by taking the diagonal of the Hessian Matrix. The next section presents an alternative to this.

5.5.3. Constructing a Smoothness Matrix from Surrounding Elements

Regularization increases the positive definiteness of the Hessian Matrix thus preventing the computation of negative values of conductivity. The following regularization matrix is based on direct relation to neighbouring elements. This idea was presented in the following papers: Pinheiro et al, (1997), Pinheiro et al, (1999).

The conductivity update equation incorporating this matrix is expressed below:

$$\Delta\sigma^i = \frac{-\{ J^T \bullet [V_{cal}(\sigma) - V_{mea}] - (\gamma \bullet D^T \bullet D \bullet \sigma^i) \}}{(J^T \bullet J) + (\gamma \bullet D^T \bullet D) + (\beta^i \bullet S)} \quad (5.19)$$

where D is the Neighbouring Smoothness Matrix

S is the Identity Matrix.

If parameter γ is set to zero then Equation 5.19 reduces to the Levenberg-Marquardt regularization. The diagram on the following page provides a graphical insight to assembling the Neighbouring Smoothness Matrix D.

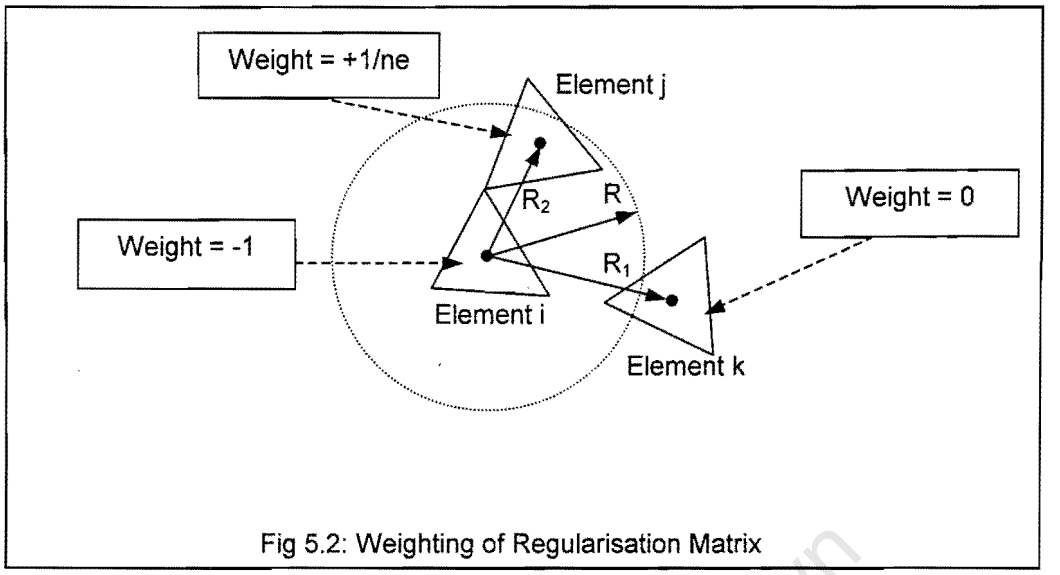


Fig 5.2: Weighting of Regularisation Matrix

Source: Pinheiro et al (1997)

Figure 5.2 with the aid of the following pseudo code will explain the assembly of the Regularization matrix D:

- Input a Smoothness Radius 'R'
- Initialise Matrix D with size 'e x e' (where e is the total number of elements in the mesh)

for a = 1 to e

- Row 'a' and column 'a' of matrix D are assigned the value of -1, i.e. $D_{a,a} = -1$.
- Find the number of elements (ne) that are within the Smoothness Radius 'R'. Note R is calculated from the centroid of triangle 'a'.
- Those elements that are within the circle are assigned a weight of 1/ne, as depicted in Figure 5.2.
- Elements that are outside the radius are assigned a value of zero.

end e

In passing the author would like to note that a subspace regularization technique incorporating prior conductivity distribution has been developed and implemented. Information on subspace regularization can be found in the following references: Kaipio et al (1999), Karjalainen et al (1999), Vauhkonen et al (1996), Vauhkonen (1997). Further, Total Variation regularization has been implemented in ERT. The reader can find information on Total Variational regularization for ERT in Borsic et al, (2001).

6. CALIBRATION AND PRE-PROCESSING OF THE SYSTEM

This section comprises some of the novel ideas of this project. These concepts were conceptualised and developed during formal communications with Dr. Wilkinson (2002) and Randal (2002).

6.1. THE NEED FOR CALIBRATION

For any electronic system there needs to be some form of calibration to minimise the errors. In modelling ERT the following factors can give rise to errors ((Fransolet et al, 2002; Kolehmainen et al, 1997a; Wilkinson, 2002):

i. Misalignment of the electrode locations,

ii. Tolerances from the electronic components,

iii. Build up of debris on the electrode sensors,

iv. Temperature fluctuations.

In effect these errors make the system less sensitive to objects that have a small conductivity difference in comparison to the surrounding medium. The following raw data set depicts the static errors that arise.

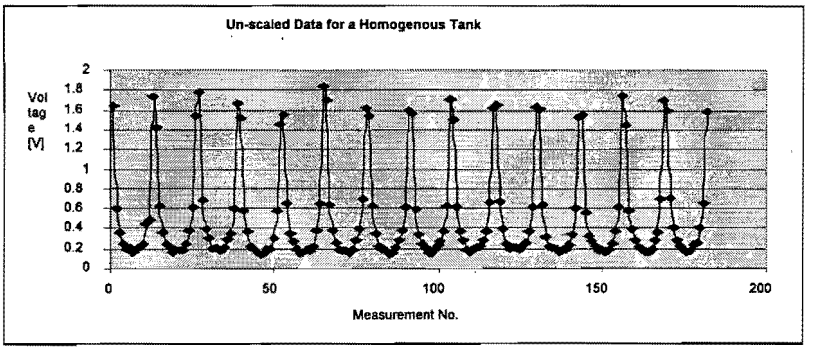


Fig 6.1: Boundary Voltages for an Un-calibrated System.

The data points plotted in Figure 6.1 are captured with the use of the sixteen electrode system described in Chapter 2.1.1 which uses the adjacent measurement strategy illustrated in Figure 2.5. The data capture is in accordance with the measurement sequence table in Appendix A. For a vessel containing a homogenous solution one would expect the boundary voltages to be periodically identical. Figure 6.1, above shows a degree of periodic reproducibility for an undisturbed tank, however it is not very distinct. The reason for this as mentioned previously, is due to the geometry of the electrodes coupled with the unequal operational amplifiers gains in the circuitry.

The above figure further illustrates the arduous task of discerning the presence of objects with small variations of conductivity in the tank. Hence, implementation of the calibration scheme forms an integral part of the software developed in eradicating or minimising some of these typical errors. Thus the next few subsections discuss some of the aspects of calibration. As will be outlined later, calibration also accommodates for the integration of baffles and other objects into the tank without the need for altering the program.

6.2. PRESENTLY IMPLEMENTED CALIBRATION SCHEMES

To the knowledge of the author a minimal number of calibration schemes have been developed for electrical resistance tomography. The most implemented scheme assumes that each electrode pair has a misaligned gain. Therefore data that comes from a particular electrode is multiplied by a scalar factor in an attempt to address the errors (Fransolet et al, 2002).

Electrode models to an extent can be considered as a calibration scheme. Over the years various electrode models have been developed, namely: Continuum

Model, Gap Model, Shunt Model, and Complete Electrode Model. The Continuum model assumed that there are no electrodes present in the tank, which led to an underestimation of the conductivities (Cheng et al, 1989; Vauhkonen, 1997). The Gap model did not take into account the shunt and contact impedance effects of the electrodes and this led to an underestimation of the conductivities (Vauhkonen, 1997). The Shunt model accounted for the shunt effect but did not consider the contact impedance, thus it led to an overestimation of the conductivities (Vauhkonen, 1997).

At present the Complete Electrode Model is recognised as the most correct model. It takes into account both the effects of the contact impedance and shunt effects of the electrodes (Vauhkonen, 1997). A detailed derivation and implementation of the Complete Electrode model can be found in Vauhkonen et al (1998b), Vauhkonen et al (1999b).

6.3. CALIBRATION SCHEMES DEVELOPED FOR UCT CURRENT-PULSE SYSTEM

Several calibration schemes were developed during the course of this work. The first method below performs a cross check between frames and then produces an averaged voltage frame that is passed through the other methods. The three scaling-schemes (described later on in this chapter) extract scaling factors from this averaged frame. If the scaling factors are multiplied with a data frame from a homogenous calibration tank then the resulting voltages should ideally form a perfectly uniform periodic sequence. These three techniques were implemented prior to the authors implementation of a reconstruction algorithm. Initially, the UCT hardware was tested using software developed by UMIST Chemeng, and calibration was carried out as a pre-processing stage.

With the completion of the authors reconstruction software the three scaling algorithms were replaced with a FEM predictive modelling scheme.

6.3.1. Outlier Testing

As mentioned before, the outlier test is different to the three schemes described later. Its main function is for the testing of erroneous and spurious data between frames. The outlier test was developed as a quick fix to a bug that arose in the hardware. It should be noted that the most recent versions of the hardware have addressed this bug and that the outlier testing is now optional. To clearly explain the outlier concept use of the following table is made.

Table 6.1: Example of Data Frames Captured from ERT System

	Frame 1	Frame 2	Frame 3	Average
1	100	101	99	100
2	76	76	76	76
3	55	56	54	55
4	40	22	40	34
5	25	25	25	25
6	10	11	12	11
⋮	⋮	⋮	⋮	⋮
n				

Table 6.1 illustrates typical data for the calibration tank. Each frame consists of 'n' values where 'n' is the number of voltage measurements per frame. In an attempt to minimise errors, the average value of each row is calculated which is then fed into the reconstruction algorithm. For a static tank all the

voltages for a particular row should be relatively identical as depicted in Table 6.1 with the exception of Row 4. The value in Row 4, frame 2 (highlighted cell), is inconsistent with the other values in that row. This erroneous voltage can be caused by timing errors in the circuitry and has the ill prone effect of altering the average voltage value. The outlier test was developed to minimise this error. The following flow chart illustrates the algorithm for the outlier test.

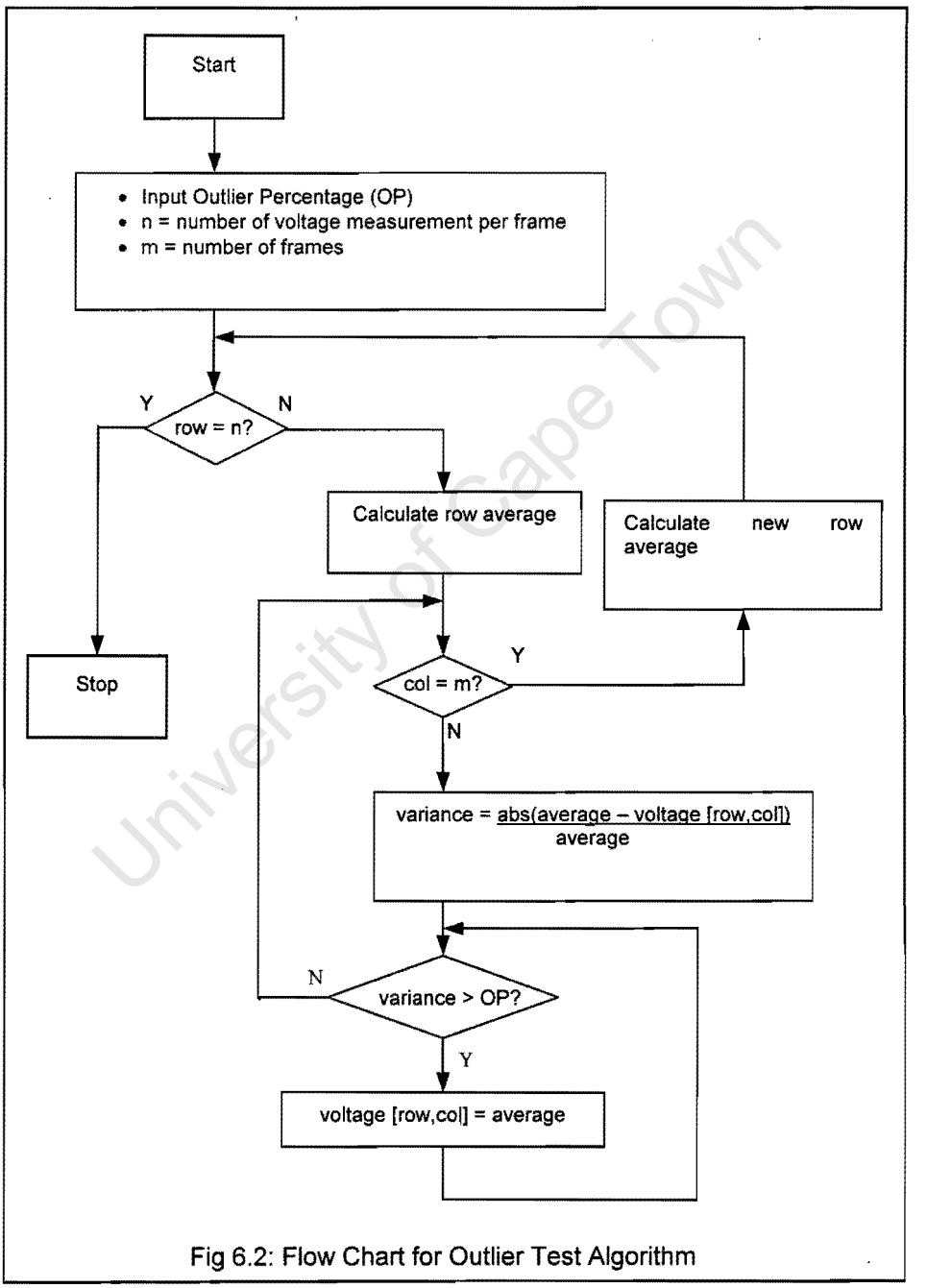


Fig 6.2: Flow Chart for Outlier Test Algorithm

The algorithm depicted in Figure 6.2 is further explained with the use of an example in the table located below.

Table 6.2: Outlier Test Example

	Frame 1	Frame 2	Frame 3	Average
Outlier Percentage (OP)	20%	20%	20%	-
Row 4	40	22	40	34
Variance = (Average – Frame i) / Average	0.18	0.35	0.18	-
V > OP%	N	Y	N	-
New Values	40	34	40	38

Table 6.2 is an example of how the outlier algorithm works. Note that this example comes directly from Table 6.1. Firstly an Outlier Percentage (OP) is specified which limits the deviance that is acceptable with the average voltage (Row 1). Row 2 of Table 6.2 is taken directly from Table 6.1. This row has the erratic data value and will thus need changing.

Next the variance for each voltage is calculated and displayed in Row 3. Row 4 compares the Variance and the Outlier Percentage and accordingly decides if the value needs changing or not. Those cells that need changing are replaced with the average value and the new average is then calculated. In comparison: The initial average was 34. After the test the new average is 38, which might not be the exact answer but it is a closer fit than the initial average. The significance of this change implies that the reconstruction algorithm is able to converge more accurately to the true solution.

6.3.2. Group-Scaling

Note that the scaling will be explained for the adjacent measurement strategy (Chapter 2.3.4), since this is the most widely used strategy and secondly, it is the strategy employed for UCT Current-pulse ERT system. However the methods explained can easily be adapted to the opposite measurement strategy (Chapter 2.3.4). The following table is needed to explain the scaling techniques and will be referred to repeatedly.

Table 6.3: 208 Boundary Voltage Measurement Sequence

Excitation Pairs		Measured Pairs (via input MUX)													Number of pairs
		1	2	3	4	5	6	7	8	9	10	11	12	13	
1	16	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	13
2	1	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	13
3	2	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-1	13
4	3	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-1	1-2	13
5	4	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-1	1-2	2-3	13
6	5	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-1	1-2	2-3	3-4	13
7	6	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-1	1-2	2-3	3-4	4-5	13
8	7	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-1	1-2	2-3	3-4	4-5	5-6	13
9	8	10-11	11-12	12-13	13-14	14-15	15-16	16-1	1-2	2-3	3-4	4-5	5-6	6-7	13
10	9	11-12	12-13	13-14	14-15	15-16	16-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	13
11	10	12-13	13-14	14-15	15-16	16-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	13
12	11	13-14	14-15	15-16	16-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10	13
13	12	14-15	15-16	16-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	13
14	13	15-16	16-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12	13
15	14	16-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13
16	15	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	13
															208

Table 6.3 is the complete measurement sequence for a 16-electrode ERT system. The above table includes the dependent information. A table of the independent measurement sequence can be found in Appendix A.

Group-scaling looks at the first column of voltage values. The first column represents the voltage measurements closest to the injection current thus it can be expected that these readings would represent the peak voltages. For a calibration tank, it is expected that the values in column one are identical.

This can be expected due to the symmetry of the tank and symmetry of the measurement strategy, i.e. injecting through electrodes 1 and 16 and measuring from electrodes 2 and 3 is expected to yield the equivalent readings for injecting through electrodes 1 and 2 and measuring from electrodes 3 and 4. However as can be discerned from Figure 6.1 the peak values are not identical. Reasons for these abnormalities are explained at the beginning of the chapter.

An explanation of group-scaling can be described as: The average voltage for column one is computed. The average voltage is then divided by the individual entries in the first column producing 'p x 1' unique scaling factors where 'p' is the number of current injection patterns used. These scaling factors if multiplied with the corresponding row values of column 1 should all equate to the average voltage. The scaling factor for each row is then used to multiply all the values in their appropriate row to form a new set of scaled voltages, i.e. Row 1 scaling factor is multiplied with columns 1 to 13 of row 1 and so on.

Group-scaling factors are calculated with data from an undisturbed homogenous system and these scaling factors are stored and multiplied with data from subsequent measurements. The reader is referred to Appendix B for examples of the above method. The assumption behind Group-scaling is that the error is a constant that can be eradicated by this Group-scaling.

6.3.3. Individual-Scaling

Individual-scaling is similar to Group-scaling. The logic behind this approach is that the error is not a constant but varying. Referring to Table 6.3, the reader is brought to the attention of the 13 columns of voltage measurements. For a homogenous tank it is expected that the values in the columns are identical. As previously mentioned, the identical values can be expected due

to the symmetry of the tank and symmetry of the measurement strategy, i.e. injecting through electrodes 1 and 16 and measuring from electrodes 3 and 4 is expected to yield the equivalent readings for injecting through electrodes 1 and 2 and measuring from electrodes 4 and 5. However Figure 6.3 illustrates that the correlation in the columns is not perfect.

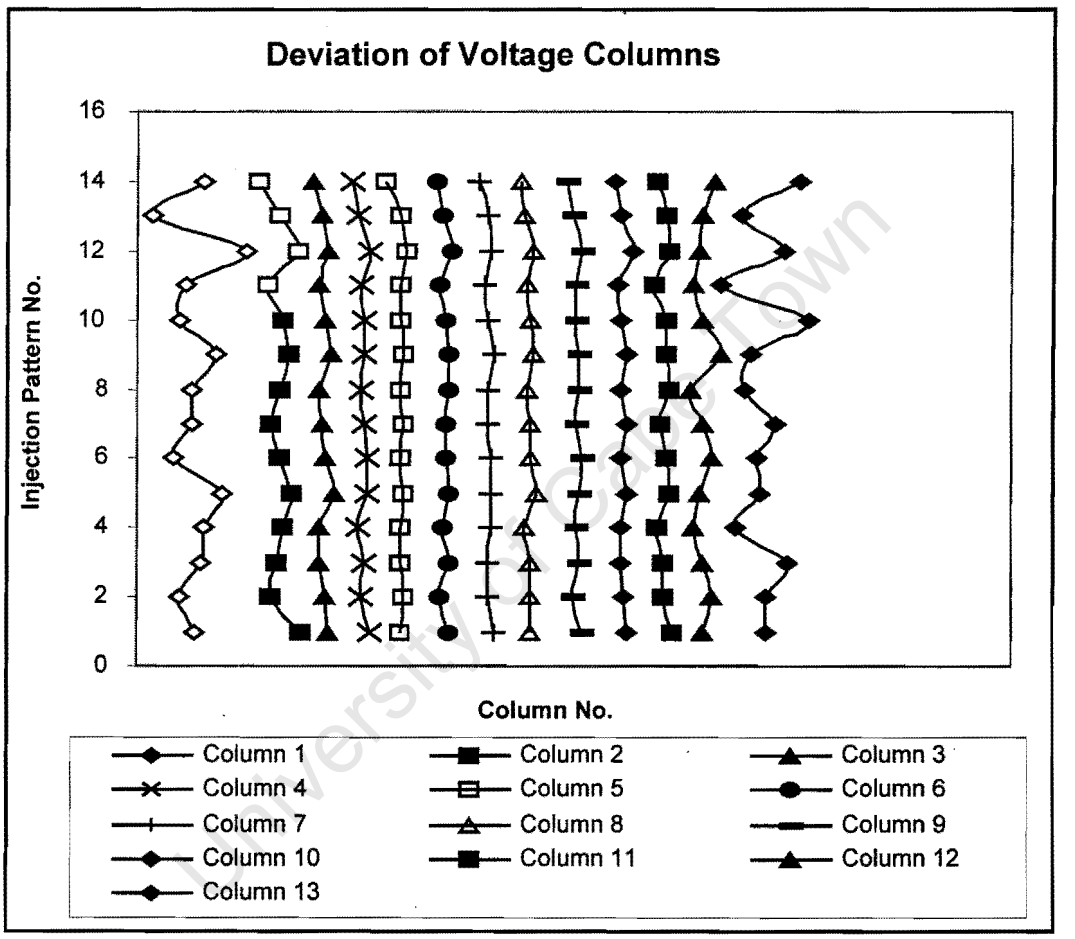


Fig 6.3: Plot of Column Voltages for Table A3.

Figure 6.3 depicts the deviations that are present between values in the columns for Table A3, which can be located in the Appendix A. Under ideal conditions the above graph should depict 13 perfectly straight vertical lines.

In Individual-scaling the values in the columns are totalled and then averaged resulting in 13 averages (1 for each column). The average voltage for each column is then divided by the individual entries in their respective column producing 'p x 13' unique scaling factors (where 'p' equals the number of current injections used). These scaling factors if multiplied with the corresponding row and column values should equate to the corresponding average column voltage. The 'p x 13' scaling factors are then multiplied with the values in their appropriate row and column to form a new set of scaled voltages.

Individual-scaling factors are calculated with data from a homogenous tank and these scaling factors are stored and multiplied with data from subsequent measurements. Once again realising that the above word explanation can be confusing, the reader is referred to Appendix B for examples of the above method.

6.3.4. Twin-Scaling

Twin-scaling is derived from Individual-scaling. This method takes care of compound errors. As with Individual-scaling the averages for each column are computed. The following diagram illustrates the concept behind Twin-scaling.

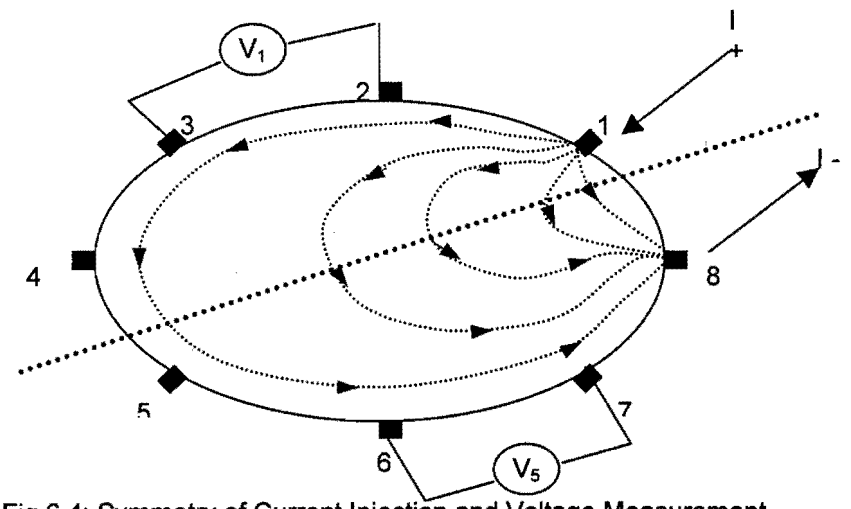


Fig 6.4: Symmetry of Current Injection and Voltage Measurement

Due to the symmetry of the tank as depicted in Figure 6.4, the following point can be made. If current is injected through electrodes 1 and 8 then the readings on V_1 (electrodes 2 and 3) should be identical to the readings on electrodes V_5 (electrodes 2 and 3). Similarly V_2 (electrodes 3 and 4) should be identical to V_4 (electrodes 5 ad 6). V_3 is the only reading that cannot be paired with another reading. This ideology of pairing is the fundamental concept behind Twin-scaling.

Twin-scaling can be applied to Table 6.1 in the following manner. Firstly the averages for each row are calculated. The following row averages are paired together and then averaged: Rows 1 and 13; 2 and 12; 3 and 11; 4 and 10; 5 and 9; 6 and 8. Row 7's average is unpaired. The average voltage that arises from the pairing is used as the average for both the paired rows i.e. The average resulting from pairing Rows 1 and 13 is used as the average for both Rows 1 and 13.

As with Individual-scaling the average voltage for each row is then divided by the individual entries in their respective row producing 'p x 13' unique scaling factors (where 'p' equals the number of current injections used). These scaling factors if multiplied with the corresponding row and column values should equate to the corresponding average row voltage. The 'p x 13' scaling factors are then multiplied with the values in their appropriate row and column to form a new set of scaled voltages.

Twin-scaling factors are calculated with data from an undisturbed homogenous system and these scaling factors are stored and multiplied with data from subsequent measurements. Realising that the above word explanation can be confusing the reader is referred to Appendix B for examples of the above method.

The effects of the scaling can be found in the Results chapter.

6.4. CALIBRATION BASED ON FEM – PREDICTED BOUNDARY VOLTAGES

The schemes described above were originally implemented on the hardware side i.e. the data was modified when it was captured. The images were then reconstructed by software developed at UMIST chemeng. With the implementation of the authors code the possibility arose for the incorporation of a calibration scheme on the software side. This section shows the need and method of integrating with the Tomo_Imager_2D (2002) software. The voltages below are produced by Tomo_2D_Imager.

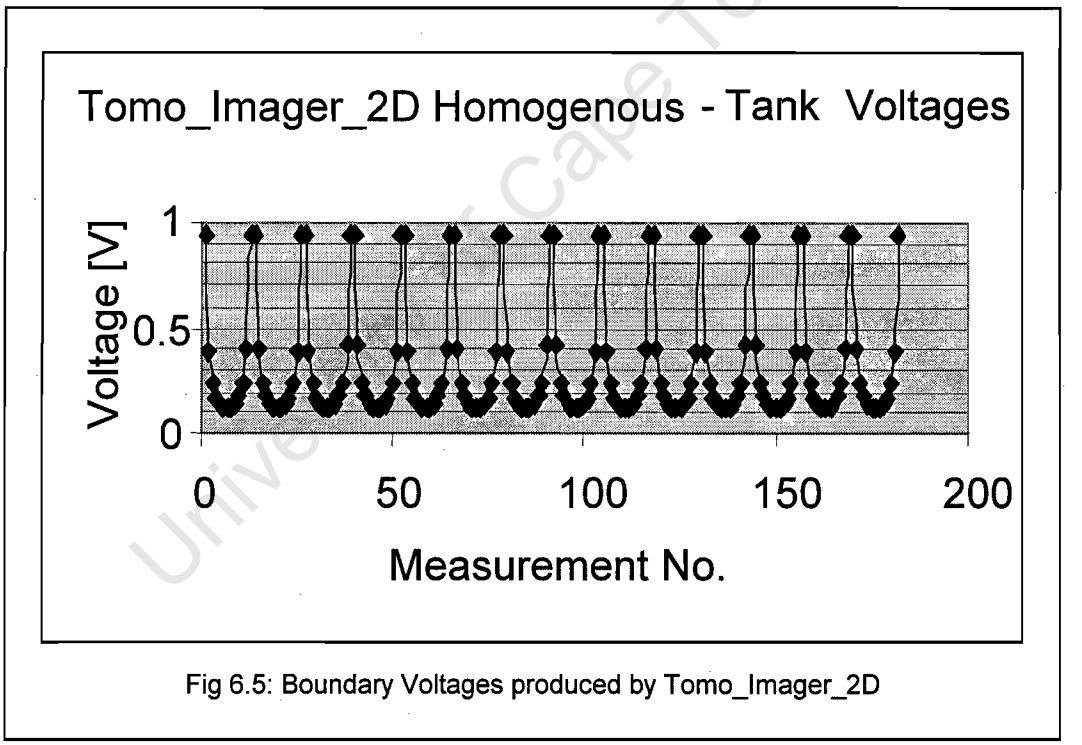


Fig 6.5: Boundary Voltages produced by Tomo_Imager_2D

The boundary voltages in Figure 6.5 were produced by the forward solution of the software. As can be seen the voltages are not periodically identical. This can be attributed to the fact that the FEM grid is not completely symmetrical with respect to the electrodes. To accommodate for these discrepancies these voltages are paired with the voltages of a homogenous tank and scaling factors are found which match the two data sets. These scaling factors are stored in a

file and used to multiply all subsequent data sets before they are reconstructed. This ensures that reconstructed images are more accurate by negating \ neutralizing some of the errors mentioned at the start of this chapter. This method is similar to the scaling techniques described before.

The integration of the calibration scheme with the software was first brought to the attention of the author by his supervisor, Dr. Wilkinson (2002). This scheme is presently used for all reconstructions.

7. PARALLELIZATION OF THE NEWTON-RAPHSON ALGORITHM

This section presents some of the fundamentals of parallelization as well as how the Newton-Raphson algorithm, which is implemented for this software, can be parallelized. The knowledge presented in this section was ingested from the following reference: Geist et al (1994).

Parallelization can be defined as accomplishing two or more tasks simultaneously. The benefits of this concept imply that time can be saved. In computing terms the power behind parallelizing has changed the impractical to practical. Parallelization can play a big part in ERT by reducing the completion time of the algorithms. The need for parallelization in ERT is evident in the timing comparisons for 3D reconstruction, which are presented in Chapter 9.3.

7.1. CONCEPT OF PARALLELIZATION

In computer terms parallelization implies the splitting up of processes that are independent and sending these tasks to various computers to be completed. This concept will be explained further with the following figure.

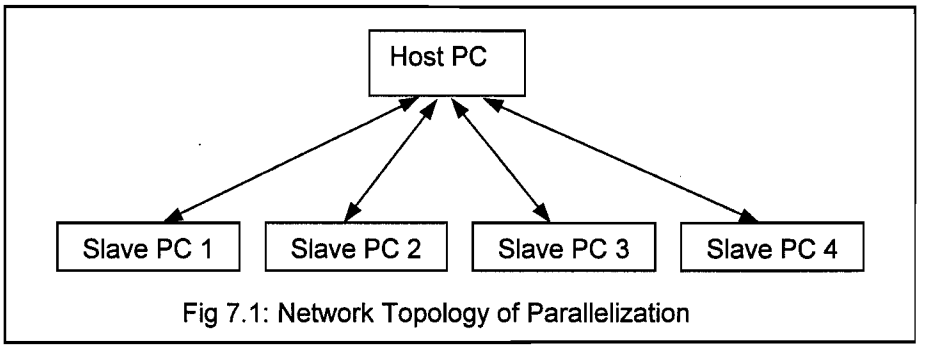


Fig 7.1: Network Topology of Parallelization

In Figure 7.1 the Host PC or Master PC is considered the key PC in the topology. To illustrate the concept of parallelization the use of a simple

arithmetic computation will be used. Assume that the host pc is required to compute the following.

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 \tag{7.1}$$

The host pc has the task of splitting up the above arithmetic sum into 's' processes where 's' is the number of slave pc's that are linked up to the host pc. (In Figure 7.1, 's' = 4). One particular delineation that the host pc can use is depicted below.

Distribute: (Task 1) + (Task 2) + (Task 3) + (Task 4)

$$(1 + 2) + (3 + 4) + (5 + 6) + (7 + 8) \tag{7.2}$$

Once the sum is split up as above then each task is sent by the host pc to the various slave pc's i.e. task 1 is sent to slave pc 1, task 2 to slave pc 2 and so on. When the slave pc has accomplished its task it returns the required answer back to the host pc.

Receive: (Task 1) + (Task 2) + (Task 3) + (Task 4)

$$(3) + (7) + (11) + (15) \tag{7.3}$$

Since the sum is not completed, the host pc can further split the sum again into the following tasks:

Distribute: (Task 1) + (Task 2)

$$(3 + 7) + (11 + 15) \tag{7.4}$$

And the process continues until a final answer is arrived at.

In summary, the host pc is in charge of firstly splitting up the process into 's' tasks, secondly distributing the tasks to the slave pc's and thirdly receiving the tasks from the slave pc's. It should also be noted that the host pc keeps track of the distribution and reception of tasks so that the data is not muddled.

At this stage it should be noted that parallelization does not necessarily require a cluster of computers. A single computer can act as the host and slave computers. This is explained further in the next section.

7.2. SOFTWARE FOR PARALLELIZATION

In this section the software required for coding the parallelization will be discussed. The reader is reminded that the software developed by the author was written in Matlab. Unfortunately at present Matlab does not officially support parallel coding. So an alternative had to be sought. One such alternative is the use of Parallel Virtual Machine (PVM).

7.2.1. Parallel Virtual Machine

Parallel Virtual Machine or PVM is a software system that allows for the networking of computers to be controlled by host computers (Geist et al, 1994). The development of PVM started in 1989 and its popularity has led to 3 versions produced. The most recent release is version 3.4.4.

PVM works on the principle of message passing between computers. It consists of two parts, namely a daemon part and a library part of interface routines. The daemon part is a program that resides in the background of all slave and host computers completing various tasks. The interfacing routines allows for the exchange of data between various computers.

A significant advantage of PVM is that the above parallelization can be tested and implemented by a single pc that is not linked to any other pc. The topology in Figure 7.1 can then be modified to the following figure.

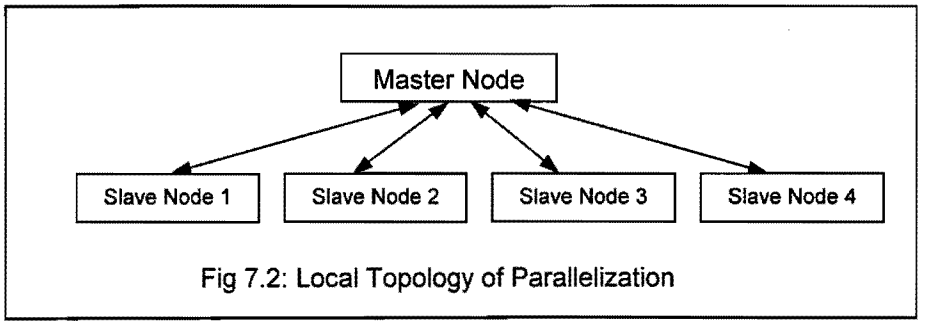


Fig 7.2: Local Topology of Parallelization

In Figure 7.2 the Host PC becomes the Master Node and all the slave pc's become the Slave Nodes. PVM splits the processing power of the pc into 's' slaves thus preserving the concept of parallelization. The topology of the system can be specified by an input file or by routine commands.

PVM is written in C code and the commands to access the library routines have to be written in C or C++ code as well. Since the reconstruction algorithms were written in Matlab it is not possible to directly access the PVM commands. A way around this problem is to write the pertinent data to a file when using Matlab. Matlab would then initiate a C written program that would read the data from the file and execute the parallelization.

After the parallelization is complete, the C code would write the data to a file and command would be returned back to the Matlab program. The Matlab program would read the file and continue with it's routine. The downfall to such an option would be the time lost in writing and reading data from files. Another alternative is to rewrite the entire reconstruction algorithm in C code but this does not seem practical. This leads to the another alternative proposed in the following subsection.

7.2.2. PVM toolkit under Matlab

Recently, Baldermo (2002), programmed a toolbox kit for Matlab that allowed for parallelization. This toolbox was based on the Parallel Virtual Machine software and allowed for the networking of computers under the Matlab program. This implied that the data that is distributed to the various slave computers do not have to be written to and read from a file, hence reducing the completion time by a greater extent.

The author has studied the above techniques but has not implemented them. It is intended that this section will provide a jump-start to parallelizing the algorithm.

7.2.3. Alternate Software

The reader's attention is brought to the fact that there are alternatives to the use of PVM. One such method is the use of Message Passing Interface (MPI). MPI is similar to PVM and is capable of accomplishing the same tasks. Similar to PVM, MPI has a problem interfacing with Matlab but recently, Baldermo (2002), programmed a tool box kit for Matlab that allowed for parallelization based on the Message Passing Interface software. The author has not studied this technique and leaves it as a follow up on this thesis.

7.3. PARALLELIZATION OF THE SOFTWARE

The most time consuming part of the algorithm is the computation of the Jacobian Matrices. The size of the Jacobian matrix depends on the number of independent boundary voltages and the number of elements that are present in the mesh. Since the number of boundary voltages are a constant, the number of elements in the mesh provides is the determining factor for completion time. Before the matrix is parallelized it is important to show how the Jacobian matrix is assembled. The following pseudo code depicts the assemblage of the Jacobian matrix.

```
for i = 1 to num_elements
        for j = 1 to num_elements
                Get σ[j] for the elements
                if i == j
```

```
                σ[j] = 1.01*σ[j]
            end
        end
        Assemble the new Stiffness Matrix ∂Y with Finite Element Method
        Jacobian [:,i] = -Y /  inverse(∂Y•Boundary_Voltage)
    end
```

The above greatly simplified code aptly depicts the independence of the assemblage of the columns of the Jacobian matrix. This independence implies that the above code can be parallelized as explained on the next page.

The pseudo code in the host pc will be as follows:

```
for i = 1 to (num_elements / num_slave_pcs)
        send task i to slave_pc_i
        receive task i from slave_pc_i
end
```

The pseudo code in the slave computers will be as follows:

```
receive task i
for j = 1 to num_elements
        Get σ[j] for the elements
        if i == j
                σ[j] = 1.01*σ[j]
        end
end
Assemble the new Stiffness Matrix ∂Y with Finite Element Method
Jacobian [:,i] = -Y /  inverse(∂Y•Boundary_Voltage)
return task i with completed answer
```

The above implementation will speed up the completion algorithm by 's' times where 's' is the number of slave computers in the network.

8. IMPLEMENTATION OF DEVELOPED ERT SOFTWARE

Tomo_Imager_2D (2002) and Tomo_Imager_3D (2003) have been developed with the idea of flexibility and user friendly environment. The software can be used in both a Windows and Linux operating system. The following subsections describe the features of the program along with the implementation of theory presented in the previous chapters.

8.1. CHOICE OF PROGRAMMING LANGUAGE AND STYLE

The choice of a programming language can affect the complexity of the algorithm, the speed of the algorithm and the visual effects. Thus it is important to choose a language that will optimise the above criteria. However the author's skills were limited to programming in Matlab and C++. C++, which uses a compiler offered the advantage of speed but lacked predefined mathematical functions. Matlab offered the distinct advantage of predefined mathematical functions along with an extensive visual graphics toolbox. However it is significantly slow for iterative processes since it uses an interpreter (Hahn, 1997). The mathematical basis of ERT swayed the author to program the algorithm in Matlab. The author felt that the disadvantage of the slow speed could be overcome by parallelizing the algorithm and use of special Matlab functions discussed later.

8.2. TWO-DIMENSIONAL DOMAIN DISCRETISATION

The theory presented in Chapter 3 stressed the importance of domain discretisation. In the sections to follow the criteria that was discussed in Chapter 3 will be implemented and explained. The author has made use of conformal meshes that are symmetric in all four quadrants. To attain a graphical perspective of the 2D meshes, the reader is referred to Chapter 8.2.6. Some of the work in this section appears in the author's undergraduate work (Naidoo, 2001) but is included for clarity and to provide a better understanding for 3D implementation.

8.2.1. Choice of Elements Implemented

The 2D domain to be discretised is circular in nature. Thus the author had the option of using either rectangles or triangles, or a mixture of both. The author opted for the use of triangles based on the following reasons:

i. Triangular elements are the most widely used for element discretisation in 2D ERT.

ii. Since two triangles can form a rectangle, the use of triangles would imply better accuracy in the reconstruction algorithm.

iii. Triangles can be used to more accurately discretise the boundary than rectangles.

8.2.2. Choice of Element Interpolation Function Implemented

With triangles being the choice of elements to discretise the 2D domain it is left to choose the interpolation function for the triangular elements. The possibilities available were the popular, faster, less accurate linear order interpolation functions (Chapter 4.3) or the less popular, slower, more accurate quadratic interpolation functions. In accordance with the vision of UCT (Wilkinson, 2001, Wilkinson, 2002) to produce fast ERT capturing systems the author has decided to implement the faster linear interpolation functions. It should be noted that the modular programming allows for the interpolation function to be changed to a quadratic form with certain modifications. For more information on linear order interpolation functions the reader is referred to Chapter 4.3.

8.2.3. Size of Elements

The size of the elements as discussed in Chapter 3.3 affects the quality of the reconstructed image. Proof of this concept can be seen in the Results chapter. By means of a graphical user interface the user is able to alter the size of various elements to their requirements. In a manner of saying, this allows for fine-tuning of the reconstruction algorithm.

8.2.4. Number and Density of Elements

The flexibility of Tomo_Imager_2D is further illustrated by its ability to choose the number of elements to use in the algorithm. The minimum number of elements is limited to 104 but there is no restriction on the maximum number of elements that can be used. In compliance with the theory mentioned in Chapter 3.3 and the proof presented in the Results chapter, increasing the

number of elements improves the quality of the reconstructed image at the expense of increased computation time.

Further, since the adjacent current injection strategy is employed the greatest change in voltage can be noticed at the boundary (Chapter 2.3.4). For this reason and the reason of increased accuracy more elements are positioned (modelled) at the boundary.

8.2.5. Numbering of Global and Local Nodes

i. Local Node-Numbering

Each element is numbered locally. The criteria used for such a numbering is depicted in Figure 8.1 and Figure 8.2.

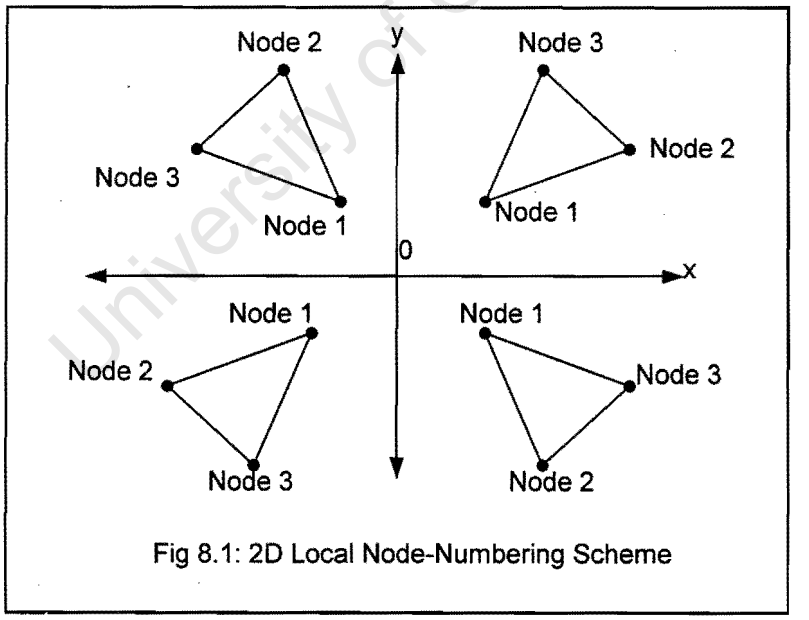


Fig 8.1: 2D Local Node-Numbering Scheme

If the triangle has a vertex pointing towards the origin as in Figure 8.1 then the following local node-numbering scheme is put into practice:

- The first node is selected as the node of the triangle that is pointing towards the origin. In this instance the first node is the node that is closest to the origin.
- The numbering of elements then proceeds in a counter clockwise direction.
- This rule applies to elements in all quadrants.

The figure below depicts the alternate orientation.

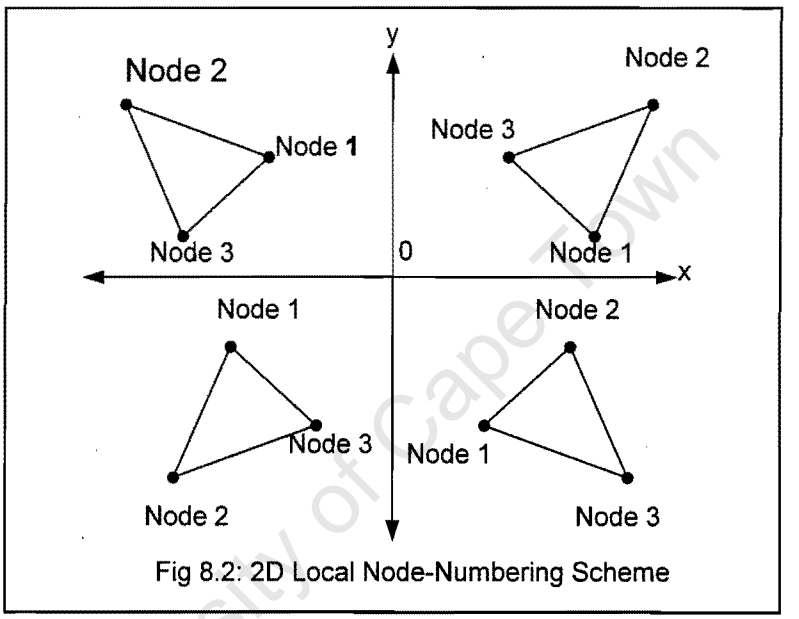


Fig 8.2: 2D Local Node-Numbering Scheme

If the triangle has a vertex pointing away from the origin as in Figure 8.2 then the following local node-numbering scheme is used:

- Moving in a counter clockwise direction, the first node of the triangle that is encountered is selected as the first node.
- The numbering of elements then proceeds in a counter clockwise direction. In this instance the second node is furthest from the origin.
- This rule applies to elements in all quadrants.

ii. Global Element Numbering

In order for a global node-numbering scheme to be used the author had to number each element. This element-numbering scheme is depicted in Figure 8.3 and explained below.

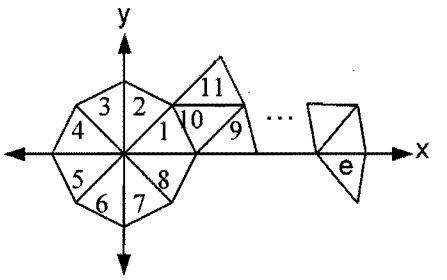


Fig 8.3: 2D Global Element Numbering Scheme

- The elements are numbered from 1 to 'e' where e is the last element in the mesh.
- Analogous to the previously mentioned node-numbering schemes the elements are numbered in a counter clockwise direction.
- The first element starts at the positive x-axis. When a ring of elements is numbered then the next ring of elements is numbered starting with the element on the positive x-axis as depicted in Figure 8.3.

iii. Global Node Numbering

The global node numbering does not follow suit of the local node-numbering scheme. This can be seen in the following illustration.

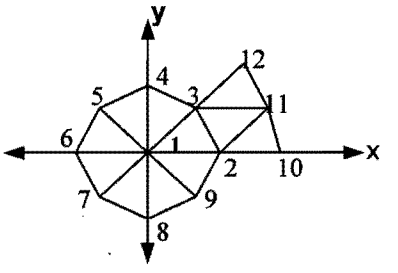


Fig 8.4: 2D Global Node-Numbering Scheme

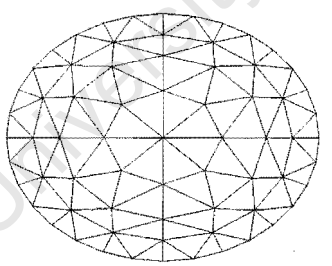
The scheme illustrated in Figure 8.4 can be described as:

- The origin is considered the first node and is labelled 1.
- The next global node is the node on the positive x-axis. The numbering then proceeds in a counter clockwise direction.
- When the last node of the ring is numbered then the next ring starting on the positive x-axis is numbered.

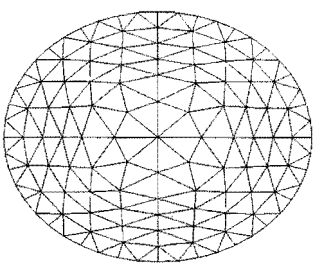
The node-numbering scheme for each element is stored in an 'e' x 3 matrix where 'e' is the number of elements in the mesh.

8.2.6. Examples of Different Meshes Used in Program

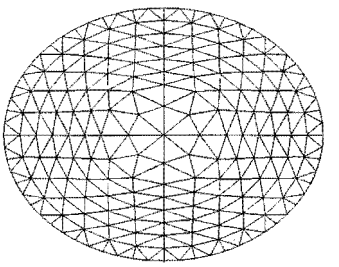
The following are examples of the types of meshes that can be chosen in the program. The meshes differ from each other in the number of elements present and indirectly in the amount of concentric rings.



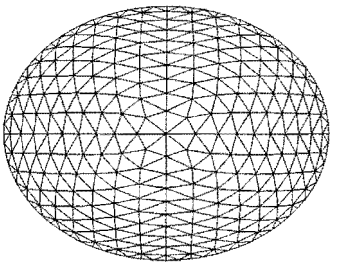
Four Rings – 104 Elements



Six Rings – 208 Elements



Eight Rings – 344 Elements



Ten Rings – 512 Elements

Fig 8.5: Examples of Meshes in Tomo_Imager_2D

8.3. THREE-DIMENSIONAL DOMAIN DISCRETISATION

The theory presented in Chapter 3 stressed the importance of domain discretisation. In the sections to follow the criteria that was discussed in Chapter 3 will be implemented and explained. The author has made use of conformal meshes that are symmetric in all four quadrants and in each layer. To attain a graphical perspective of the 3D meshes, the reader is referred to Chapter 8.3.6.

8.3.1. Choice of Elements Implemented

The 3D domain to be discretised is cylindrical in nature. Thus the author had the option of using volume elements such as rectangular prisms, triangular prisms, tetrahedrons, pyramids, etc. The author tried to simplify the jump from 2D to 3D by using similar discretisation processes. This can be seen in the following figure.

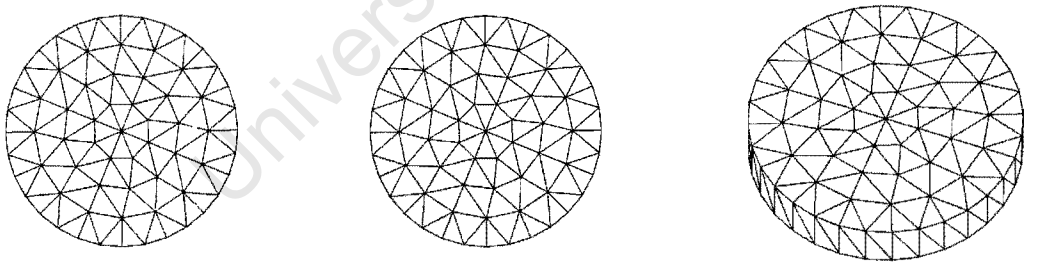


Fig 8.6: 3D Domain with 1 Layer

Figure 8.6 shows that the top view of the three-dimensional discretisation is identical to both the bottom view and the two-dimensional domain discretisation. To explain this further, consider each triangular element as seen from the top view to extend vertically downwards to the bottom triangle. The resulting volume elements are depicted in the following diagram.

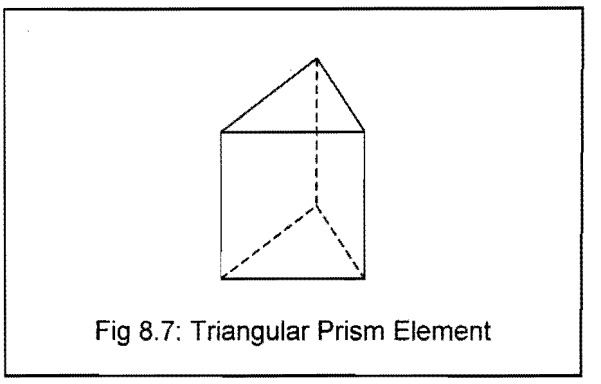


Fig 8.7: Triangular Prism Element

However, the triangular prism in Figure 8.7 would not provide a very accurate estimation of the voltages present within the domain. Thus the author had to discretise the prism into more elements. In an attempt to maintain a consistent geometrical shape the author was able to discretise the prism into three tetrahedral elements that are of different size to each other. This process can be seen below.

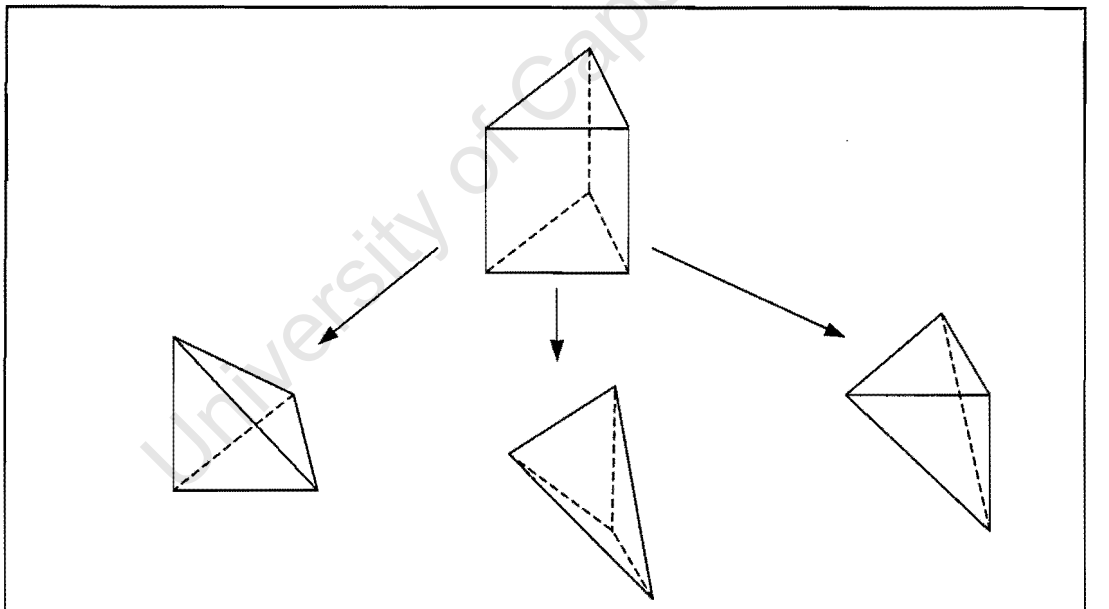


Fig 8.8: Discretisation of Triangular Prism into 3 Tetrahedrals

The author opted for the use of tetrahedrals based on the following reasons:

i. Tetrahedral elements are the most widely used for element discretisation in 3D ERT.

ii. Using tetrahedral elements as opposed to prisms would improve the accuracy of the reconstructed image.

iii. Tetrahedral elements can model the cylindrical boundary more accurately than rectangular prisms.

8.3.2. Choice of Element Interpolation Function Implemented

With tetrahedrals being the choice of elements to discretise the 3D domain it is left to choose the interpolation function for the elements. As with 2D elements the possibilities available were the popular, faster, less accurate linear order interpolation functions (Chapter 4.3) or the less popular, slower, more accurate quadratic interpolation functions. The move from 2D to 3D implied that time would become a more crucial criterion. With this in mind the author has implemented the faster linear interpolation functions. Similar to 2D imaging, with certain modifications the interpolation function can be changed to a quadratic form. For more information on linear order interpolation functions the reader is referred to Chapter 4.3.

8.3.3. Size of Elements

The size of the elements as discussed in Chapter 3.3 affects the quality of the reconstructed image. This can be seen in the Results chapter. By means of a graphical user interface the user is able to alter the size of various elements. The height of each layer can also be adjusted. In a manner of saying, this allows for fine-tuning of the reconstruction algorithm.

8.3.4. Number and Density of Elements

The flexibility of Tomo_Imager_3D is further illustrated by its ability to choose the number of elements to use in the algorithm. Both the number of layers and the number of elements per layer can be adjusted. Just as with the 2D algorithm the minimum number of elements per layer is limited to 104 but there is no restriction on the maximum number of elements per layer. Also, there is no limit to the number of layers that can be used. In compliance with the theory mentioned in Chapter 3.3 and the proof presented in the Results chapter, increasing the number of elements improves the quality of the reconstructed image at the expense of increased computation time.

Further, since the adjacent current injection strategy is employed the greatest change in voltage can be noticed at the boundary (Chapter 2.3.4). For this reason and the reason of increased accuracy more elements are positioned (modelled) at the boundary.

8.3.5. Numbering of Global and Local Nodes

i. Local Node-Numbering

Each element is numbered locally. The criteria used for such a numbering is depicted in Figure 8.9 and Figure 8.10. Note that the prism is discretised into three tetrahedrals that differ in size. As with 2D the local node numbering depends on the orientation of the triangular prism in relation to the origin.

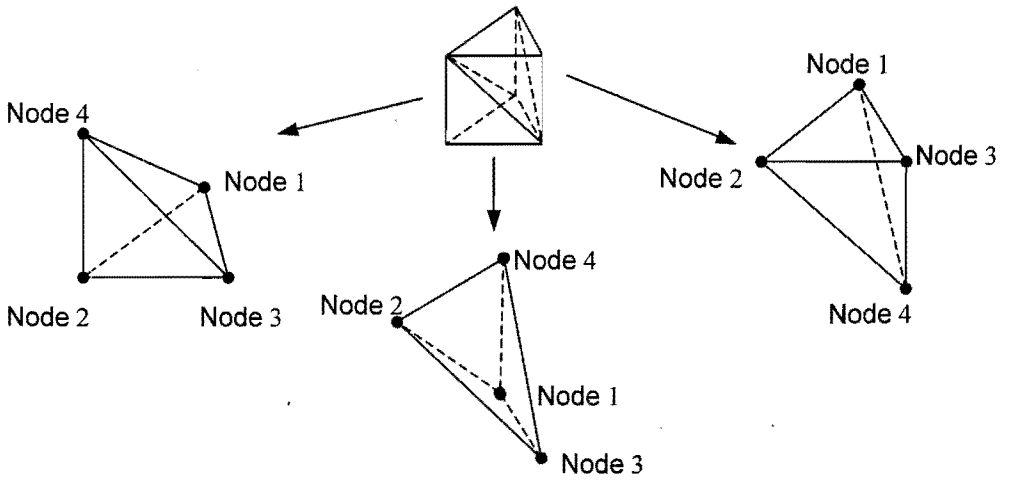


Fig 8.9: 3D Local Node-Numbering Scheme

In Figure 8.9 the origin is located perpendicularly into the page. For such an orientation the following local node-numbering scheme is used.

a. The first node is selected as the node of the element that is pointing towards the origin. In the above instances the first node is the node that is closest to the origin.
b. The numbering of elements then proceeds in a counter clockwise direction.

The figure below depicts the alternate orientation.

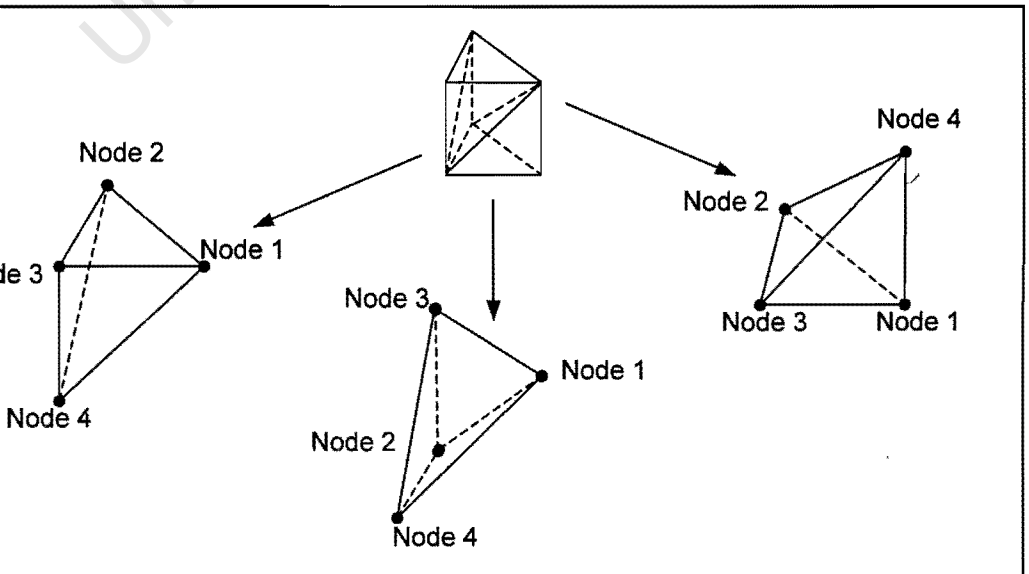


Fig 8.10: 3D Local Node-Numbering Scheme

In Figure 8.10 the origin is located perpendicularly out of the page. For such an orientation the following local node-numbering scheme is put into practice.

a. Moving in a counter clockwise direction, the first node of the element that is encountered is selected as the first node.
b. The numbering of elements then proceeds in a counter clockwise direction. In the above instances the second node is the node that is furthest from the origin.

ii. Global Element Numbering

Elements are numbered firstly according to the triangular prisms that they form. This is depicted below.

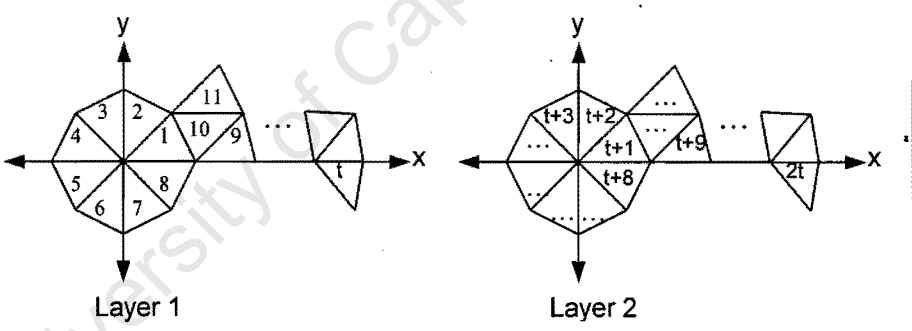


Fig 8.11: 3D Global Element Numbering Scheme

The following explanation is offered for the numbering:

- Elements are numbered firstly by their triangular prisms and then each prism comprising of 3 tetrahedrals are numbered.
- Prisms are numbered from the bottom layer up.
- The triangular prism on the bottom layer closest to the origin is numbered as element one. The numbering then proceeds in a counter clockwise direction until prism 't' is reached, where t is the number of triangular prisms in a layer.

- Once all the prisms on the layer are numbered then the same numbering principle is applied to the next layer except that the next element starts at 't + 1'.
- In each prism the first element numbered is the tetrahedral with the base at the bottom.
- The second element is considered as the tetrahedral with the base at the top.
- Lastly, the third element is the remaining element.

iii. Global Node Numbering

The global node numbering in 3D follows similar suit to the global node numbering of 2D. This can be seen in the following illustration.

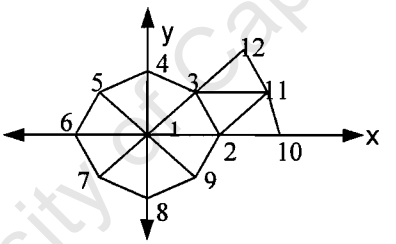


Fig 8.12: 3D Global Node-Numbering Scheme

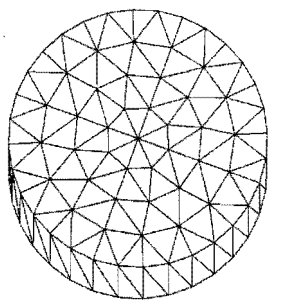
The scheme illustrated in Figure 8.9 can be described as:

- The origin on the bottom layer is considered the first node and is labelled 1.
- The next global node is the node on the positive x-axis. The numbering then proceeds in a counter clockwise direction.
- When the last node of the ring is numbered then the next ring starting on the positive x-axis is numbered.
- When all the rings on the first layer are numbered then the numbering continues to the next layer starting at the origin and following the same rules mentioned above.

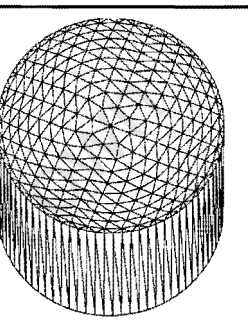
The global node-numbering scheme for all the elements are stored in an 'e' x 4 matrix where 'e' is the total number of elements in the mesh.

8.3.6. Examples of Different Meshes Used in Program

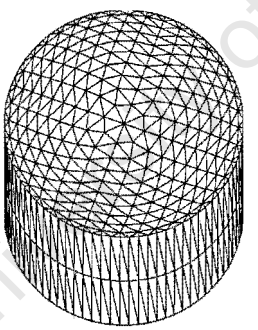
The following are examples of the types of meshes that can be chosen in the program. The meshes differ from each other in the number of elements per layer and the number of layers.



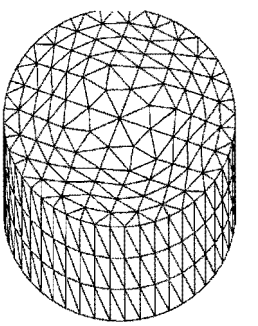
1 Layer – 4 Rings



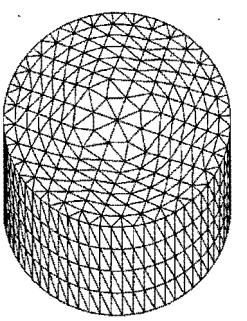
1 Layer – 10 Rings



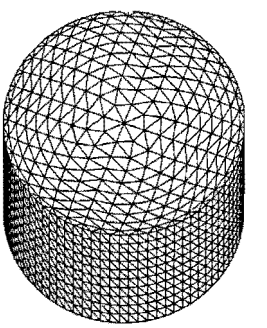
2 Layers – 10 Rings



3 Layers – 6 Rings



4 Layers – 8 Rings



10 Layers – 10 Rings

Fig 8.13: Examples of Meshes in Tomo_Imager_3D

8.4. FINITE ELEMENT METHOD

The finite element method concept of assembling, referencing and solving the linear equation are discussed in the following subsections. Note that the principles below hold true for both two-dimensional and three-dimensional ERT.

8.4.1. Assembling the Linear Matrix Equations

The global master matrix Y is assembled as outlined in Chapter 3 and is of the size 'n x n', where 'n' is the number of nodes in the mesh. The current vector 'I' is a 'n x p' matrix, where p is the number of current injection patterns. Each column of the current matrix has two non-zero entries. Theses entries are the positive and negative current that is injected and removed on the electrodes. The position of these entries is dependent on the pair of electrodes used and the nodes that represent these electrodes.

8.4.2. Referencing the Linear Matrix Equations

Due to the symmetry of the tank the central node namely node one is selected as the reference node for 2D ERT. In 3D the central node on the lowest layer is chosen as the reference node. This implies that in the global master matrix Y all row and column entries corresponding to the central node are set to zero except the diagonal, which is set to unity. This is in accordance with the theory in Chapter 3. Since the Current matrix is initially zeroed and the non-zero values are inserted at nodes on the boundary there is no need to reference the corresponding reference node.

8.5. SOLVING THE LINEAR MATRIX EQUATIONS

The set of linear equations are solved with the aid of Matlab's predefined numerical functions. The Cholesky factorisation method was used to solve the equations. Since there are 'p' current injection patterns the set of linear equations has to be solved 'p' times.

8.6. EXPLANATION OF IMAGING TECHNIQUES

The author developed image-displaying routines to provide a better depiction of the reconstructed image. The following pages explain what the various imaging techniques accomplish. The diagram on the following page is specific to two-dimensional imaging.

Figure 8.14a depicts the 3 plots of independent 104 boundary voltages. The top subplot represents the Measured Boundary Voltages. The middle subplot represents the Calculated Boundary Voltages produced by Tomo_Imager_2D. Lastly, the bottom subplot represents the two voltages superimposed on each other. This subplot gives a visual perspective of the convergence of the algorithm, i.e. the fewer the gaps between the two voltages the more accurate is the reconstructed image.

Figure 8.14b is the 2D triangular mesh on which calculations were performed. The calculated voltages displayed in Figure 8.14a are produced from this triangular mesh. The mesh depicts the conductivity distribution that is present within the tank. The dark triangles represent areas that are less conductive than the surrounding areas. In this specific instance the collative dark region represents the presence of an object namely an insulting rod.

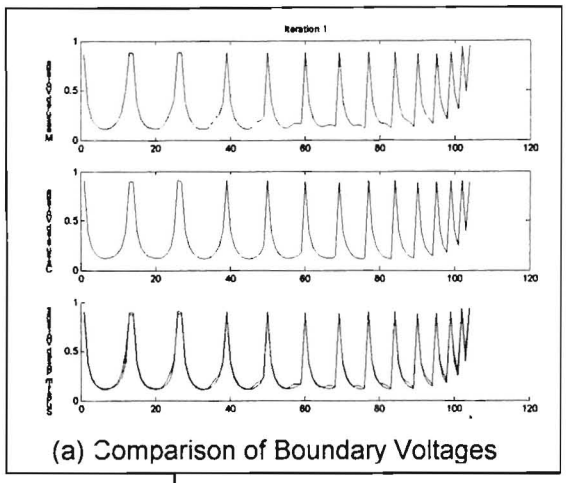
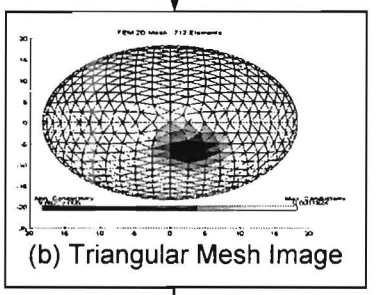
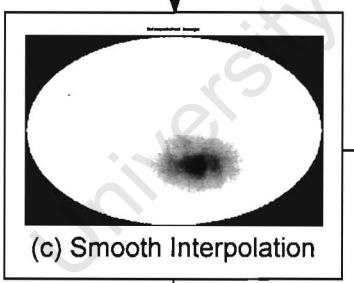


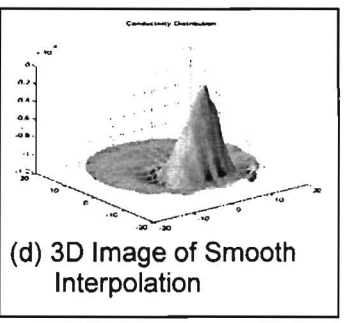
Image Boundary Voltages



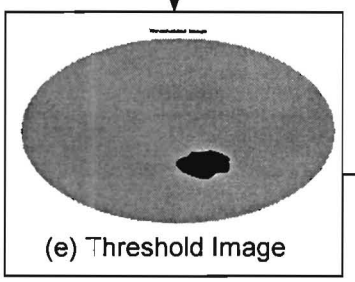
Interpolation



3D



Threshold



3D

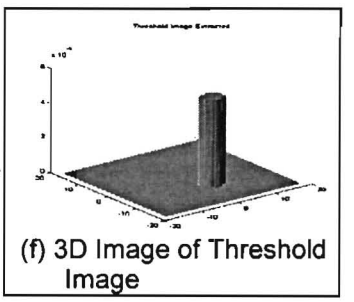


Fig 8.14: Explanation of 2D Imaging Technique

The triangular mesh does not always provide a clear depiction of the tanks contents. For this reason, the author expanded the array of displaying methods. The next image (Figure 8.14c) is obtained from the triangular mesh by using Matlabs 'interp' function. The interp function assimilates the conductivity distribution along with the corresponding Cartesian coordinates for the triangular mesh and resamples the data with the use of a low pass interpolation function producing Figure 8.14c (Matlab, 2000b).

The Smooth Interpolation Image (Figure 8.14c) includes noisy data. Often the user may wish to see a clean picture. This request is accommodated for by taking the threshold of the Smooth Interpolation figure as can be seen in Figure 8.14e. The threshold is dynamic in nature and is calculated in the following manner:

- Firstly, a conductivity histogram of the Smoothed Interpolation Image (Figure 8.14c) is drawn.
- A threshold percentage is selected. (Generally about 25%).
- When a cumulative sum of the histogram reaches the threshold percentage then the respective conductivity 'bin' for the histogram is selected as the threshold point.

The Smoothed Interpolation Image is scanned and those conductivities that surpass the threshold point is extracted and displayed in Figure 8.14e.

Figure 8.14d and Figure 8.14f try to provide a three-dimensional view to Figure 8.14c and Figure 8.14d respectively. The vertical axis in Figure 8.14d represents resistivity. The peak denotes the presence of a resistive object at that particular spot. Noting that the boundary voltage measurements are planar (two-dimensional), an assumption that the object (rod) is vertically standing and not slanting is made. This assumption allows for a predictive three-dimensional image such as Figure 8.14f to be plotted. The user can thus see the objects position and possible three-dimensional shape.

The imaging techniques used for 2D are similar to the imaging techniques employed for 3D. For completion the 3D imaging will be explained with the aid of Figure 8.15.

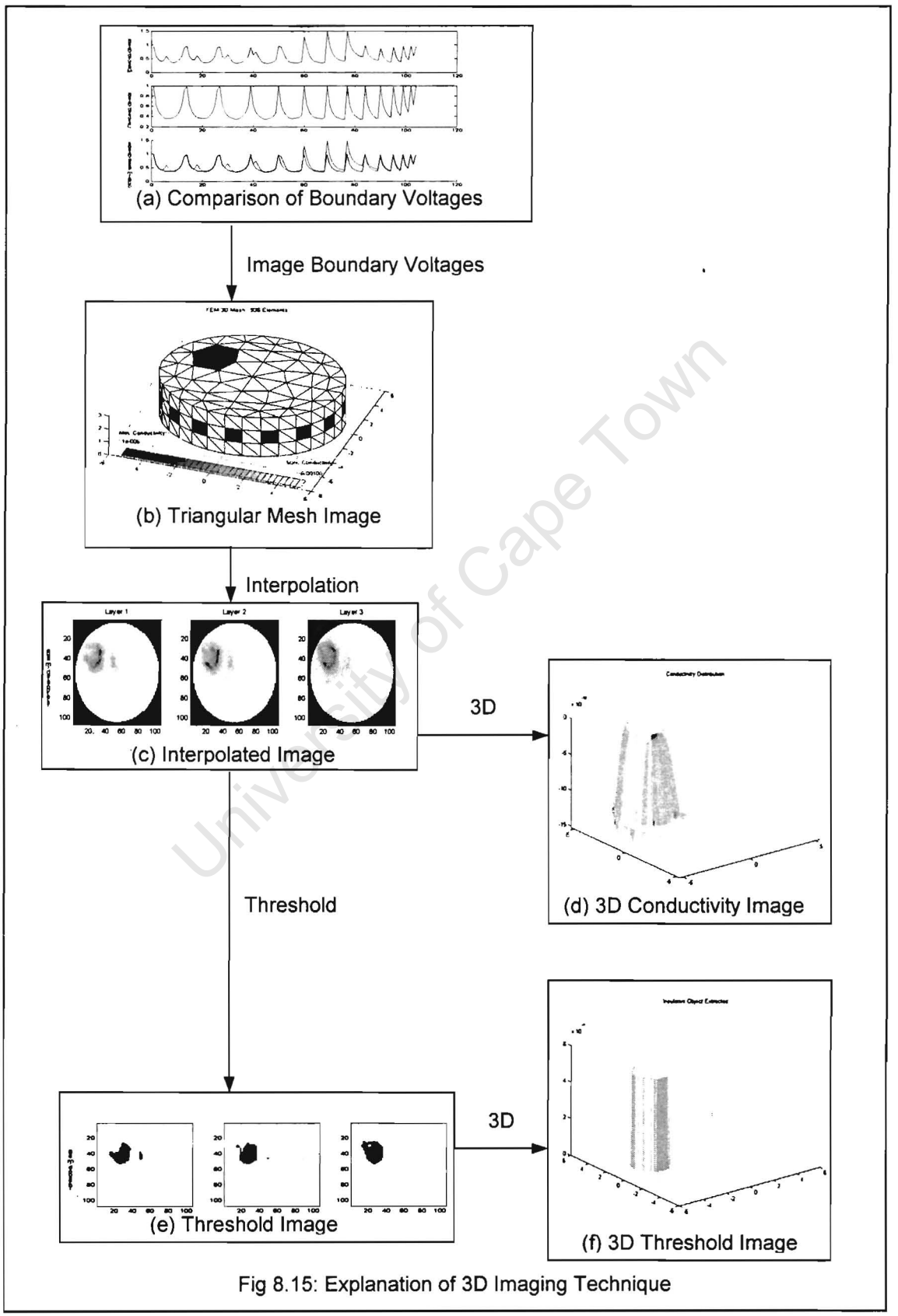


Fig 8.15: Explanation of 3D Imaging Technique

Figure 8.15a depicts the 3 plots of independent 104 boundary voltages. As with 2D the top subplot represents the Measured Boundary Voltages. The middle subplot represents the Calculated Boundary Voltages produced by Tomo_Imager_2D. Lastly, the bottom subplot represents the two voltages superimposed on each other providing a visual perspective of the convergence of the algorithm, i.e. the fewer the gaps between the two voltages the more accurate is the reconstructed image.

Figure 8.15b is the 3D tetrahedral mesh on which calculations were performed. The calculated voltages displayed in Figure 8.15a are produced from this tetrahedral mesh. The dark tetrahedrals represent areas that are less conductive than the surrounding areas. In the above instance the collative dark region represents the presence of an object namely an insulting rod.

The tetrahedral mesh does not provide a clear indication of the conductivity distribution that is present within mesh. Hence the next image (Figure 8.15c) provides 2D slices of the conductivity distribution for each layer. The image on the left represents the first layer, followed by the second and third layers. These images were formed with the use of Matlabs 'interp' function.

As with 2D the Smooth Interpolation Image (Figure 8.15c) is dynamically threshold to produce the extracted objects in Figure 8.15e.

Figure 8.15d and Figure 8.15f try to provide a three-dimensional view to Figure 8.15c and Figure 8.15e respectively. The vertical axis in Figure 8.15d represents height. The image is formed by summing the 2D conductivities for each layer. Figure 8.15f is formed by summing the threshold conductivities in Figure 8.15e.

8.7. TECHNIQUES TO SPEEDUP THE ALGORITHM

The following tricks allowed for the algorithm to complete faster:

i. Manipulation of Sparse Matrices

ii. Parallelization of the Algorithm

iii. Use of the Pre-computed Matrices

iv. Using '0-1' vectors to extract information faster than for loops

v. Using the '\' for computing the inverse of matrices

The completion time of the algorithm can be greatly reduced by manipulating the characteristic sparseness of the global matrices. Matlab has predefined functions that handle sparse matrices. Two such functions are 'symmmd' and 'sparse'. The author has made use of the 'sparse' function on both the global master matrix and the load matrix.

It should be noted that the author has not implemented the parallelization of the algorithm, but that the algorithm can be parallelized with the aid of PVM or MPI. Explanation of this can be found in Chapter 7.

The Noser algorithm as described in Chapter 5 is a perfect example of using pre-computed matrices to reduce the completion time of the algorithm. Another example is in the assembling of the stiffness matrix, where the matrix is written to a file and when needed simply read from the file and then multiplied with the respective element conductivity.

Due to the mathematical nature of Matlab, the use of the '0-1' vector eradicated the need for certain for loops and in Matlab this meant that considerable time could be saved.

Lastly the '\' operator in place of 'inv' allowed for better conditioning of the inverse matrices which indirectly improved the accuracy of the solution. Also as indicated by Hahn (1997) the back slash operator requires fewer flops to complete which translates to the inverse matrix being computed in a shorter time span.

9. RESULTS AND COMPARISONS

This section displays the results obtained from the software programs Tomo_Imager_2D (2002) and Tomo_Imager_3D (2002), developed by the author. The experiments were conducted in a cylindrical tank with the following dimensions: diameter 50cm and height 50cm. The 16 electrodes are located at midway height and have the following dimensions: height 10cm and breadth 2cm. The tank is filled with tap water and typically about 50 milligrams of NaCl (salt) is added, resulting in a conductivity of 10^{-2} S/m.

9.1. RESULTS OF TWO DIMENSIONAL RECONSTRUCTION

The following subsections depict the ability of Tomo_Imager_2D to correctly reconstruct the conductivity distribution present in tanks.

9.1.1. Reconstructed Images for Simulated Data

The author has included simulated data results since the first tests of Tomo_Imager_2D were conducted on this data. The use of a flowchart on the following page is used to describe what is meant by simulated data along with how the simulated results should be interpreted.

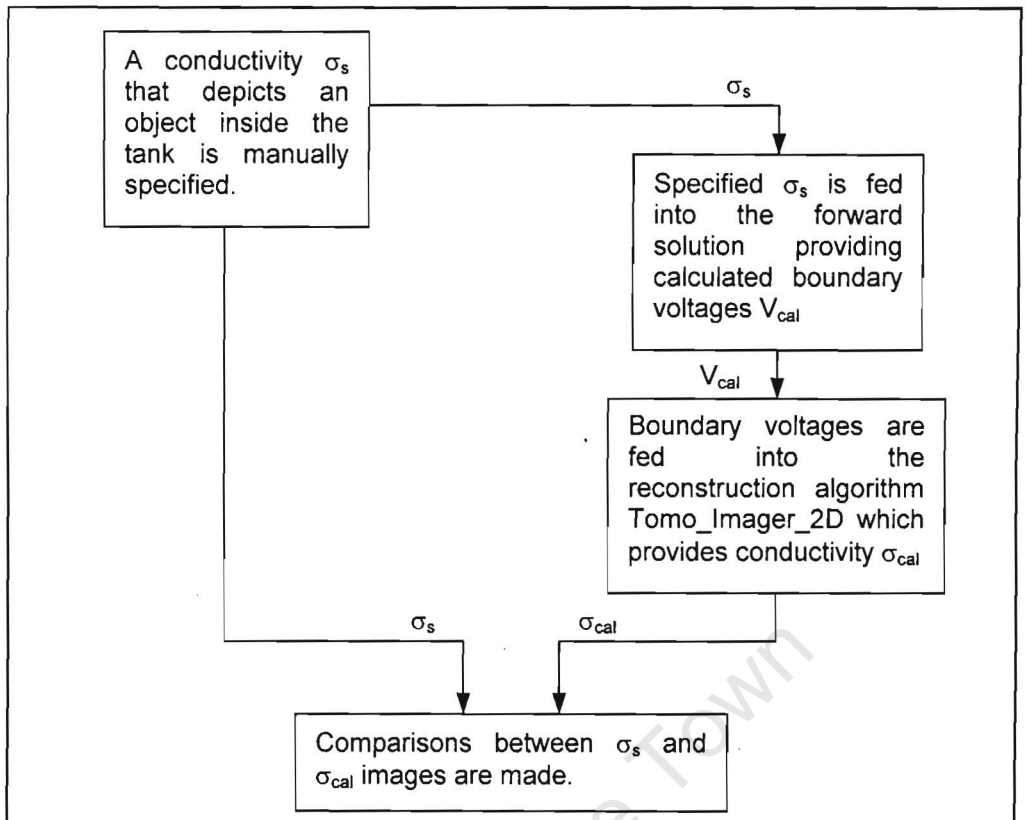


Fig 9.1: Flowchart Explanation of Reconstructed Simulated Data

Hence, results for the simulated data are:

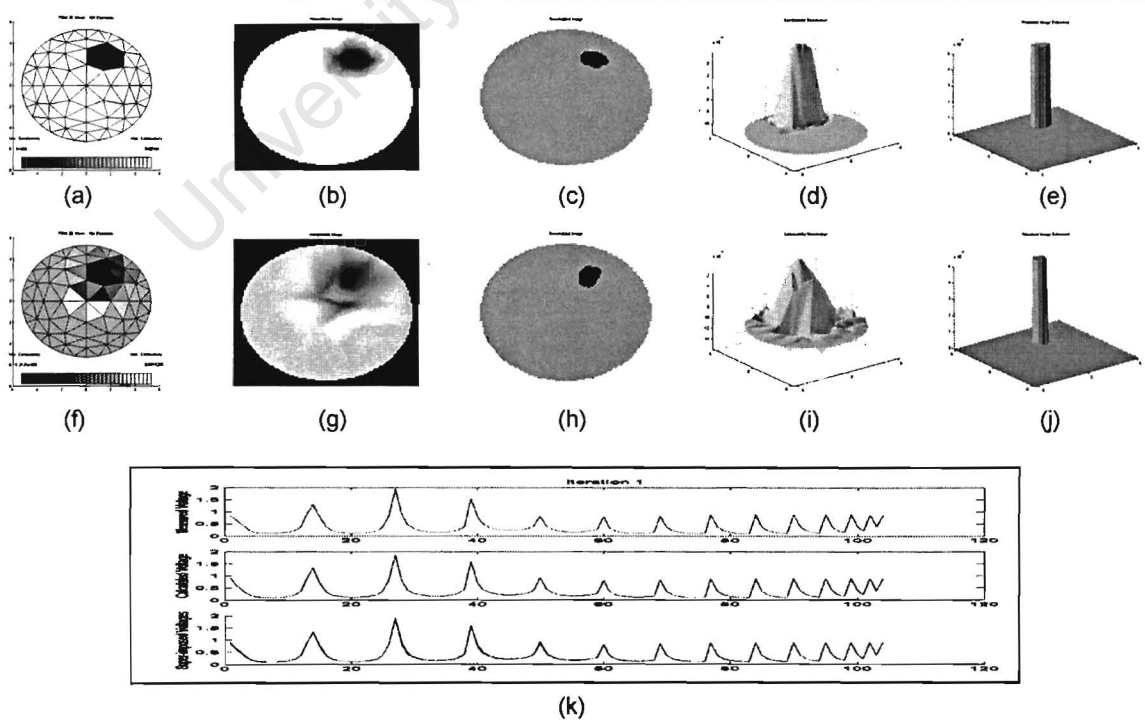


Fig 9.2: Reconstructed Images for Simulated Data

All the diagrams from Figure 9.2 were produced by Tomo_Imager_2D (2002). Figure 9.2a is the user input conductivity σ_s (simulated data) as referred to in Figure 9.1. Diagrams (b) to (e) in Figure 9.2 are the resulting images of Diagram (a). Diagrams (f) to (j) are the reconstructed images for these simulated data. Detailed information of the image representations can be found in Chapter 8.6.

9.1.2. Reconstructed Images for an Insulating Rod

An insulating rod is highly non-conductive and does not allow current to pass through. This variation from the normal current pattern implies that informative data can be extracted from boundary voltages. The results for a purely insulating rod are displayed below:

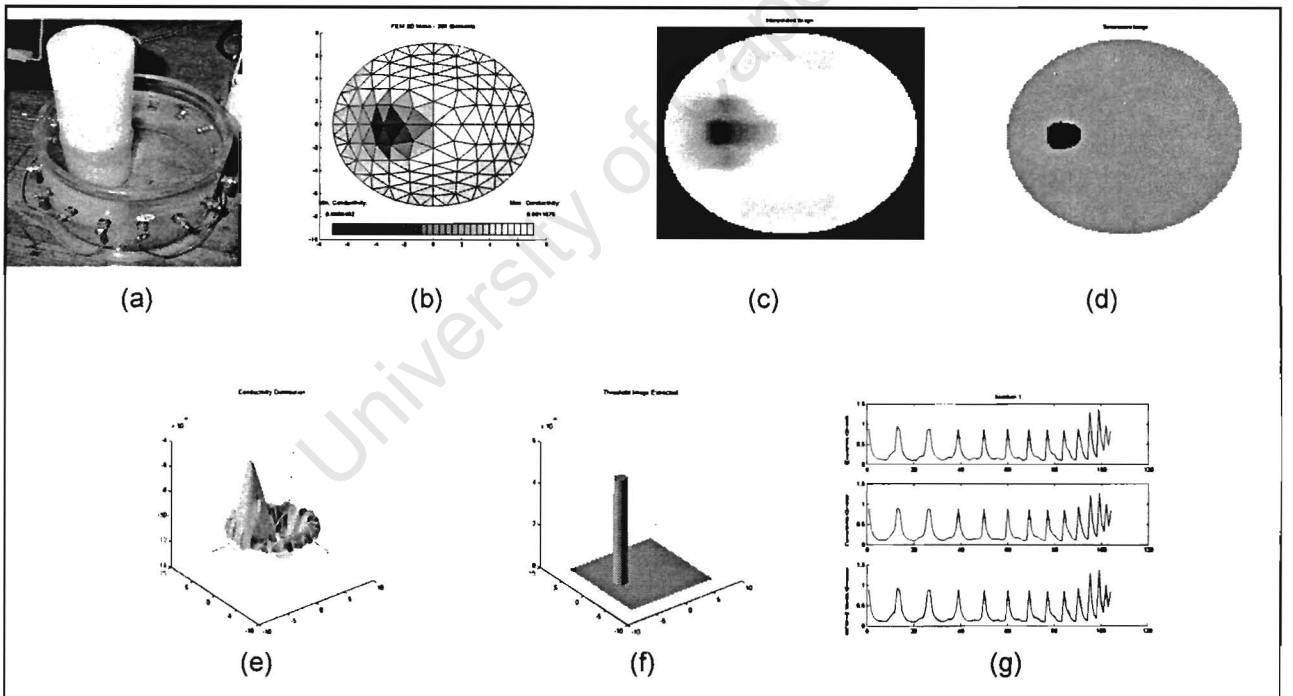


Fig 9.3: Reconstructed Images for an Insulating Rod

The dark regions in Figure 9.3 represent the insulating rod. Diagram (a) is a real picture of the rod in question.

9.1.3. Reconstructed Images for Gaseous Flow

Gases are partly insulative due to the air bubbles and thus it is possible to image bubbles in a tank. This subsection depicts the results of imaging bubbles.

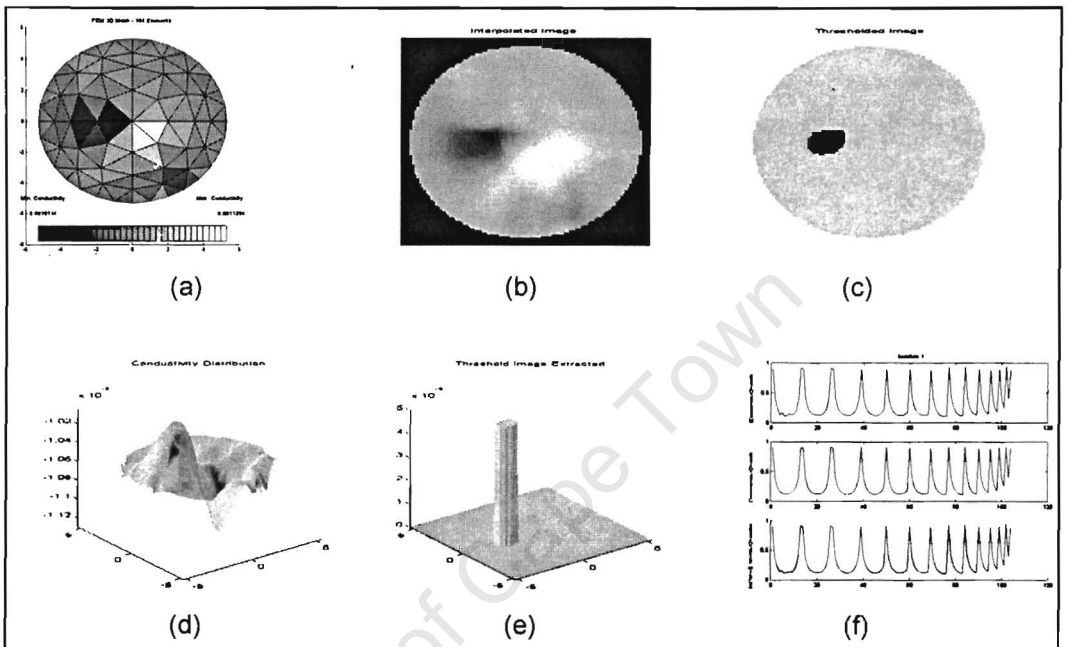


Fig 9.4: Reconstructed Images for Air Bubbles

The dark regions in Figure 9.4 depict air bubbles. The air bubbles were created by injecting pressurised air through a small opening located at the base of the tank.

9.1.4. Reconstructed Images for a Conductive Metal Rod

Metals are on the other side of the spectrum from the insulating rod. Metals can be considered a 'pure' conductor and would disturb the current flow in the tank. This subsection depicts the results of a metal rod placed in the tank.

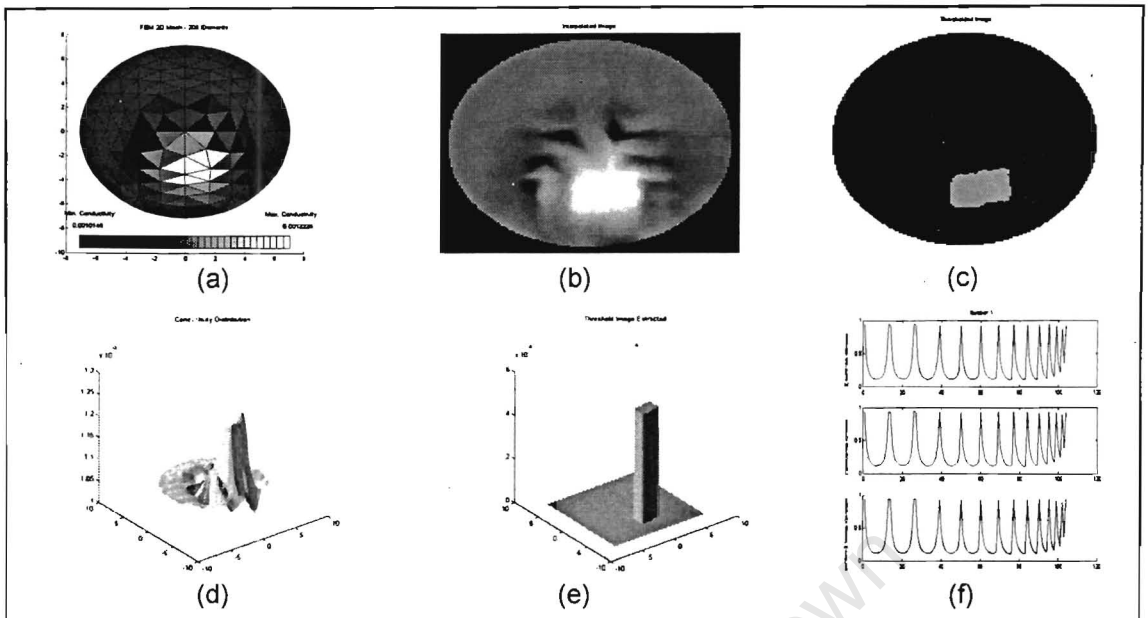


Fig 9.5: Reconstructed Images for a Metal Rod

The lightly shaded areas in Figure 9.5 represent the more conductive regions i.e. the metallic object is located at the light region. Note that Figure 9.5d and Figure 9.5e have been automatically 'flipped' in the program to highlight the region of interest and provide a clearer view of the image.

9.1.5. Comparison of Images for Varying Number of Elements

This subsection presents the results produced for five meshes that differed by the number of elements. The number of elements used for the five meshes are:

i. 104 Elements

ii. 208 Elements

iii. 344 Elements

iv. 512 Elements

v. 712 Elements

The results for five different meshes are shown below.

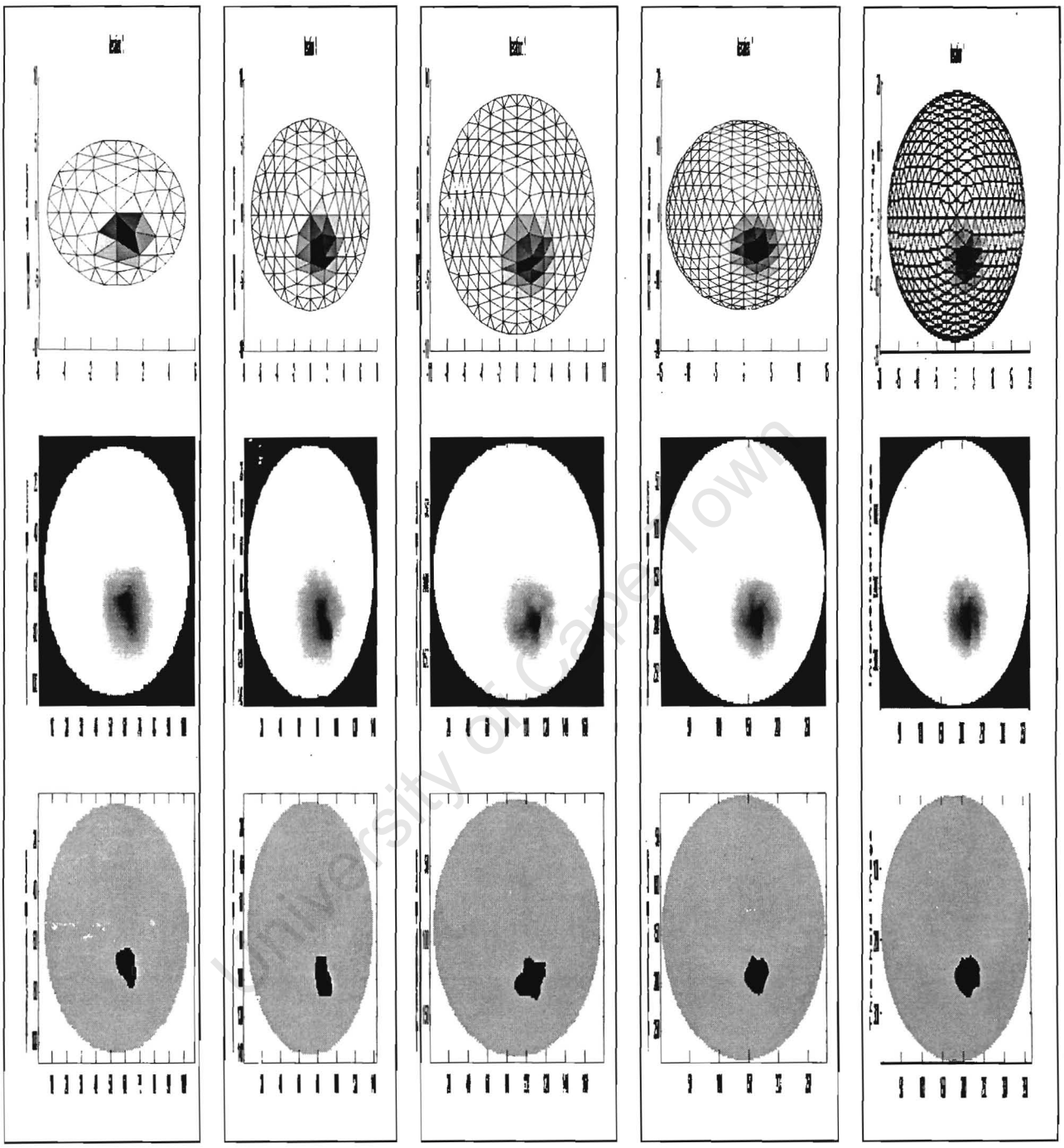


Fig 9.6: Caparison between Varying Number of Elements

In Figure 9.6, Diagram (a) shows the results for the fewest elements (104-elements) whilst Diagram (e) shows the results for the most elements (712 elements). The dark regions in the above figure represent the insulating rod.

9.1.6. Comparison of Time of Convergence for Varying Number of Elements

This section presents the timing results for meshes with varying number of elements.

Table 9.1: Time taken for Completion of Reconstruction Algorithm

Number of Elements	Time in Seconds
104	3.916
208	13.199
344	35.851
512	82.658
712	167.842
944	382.516

The data values in Table 9.1 are plotted on a line graph below.

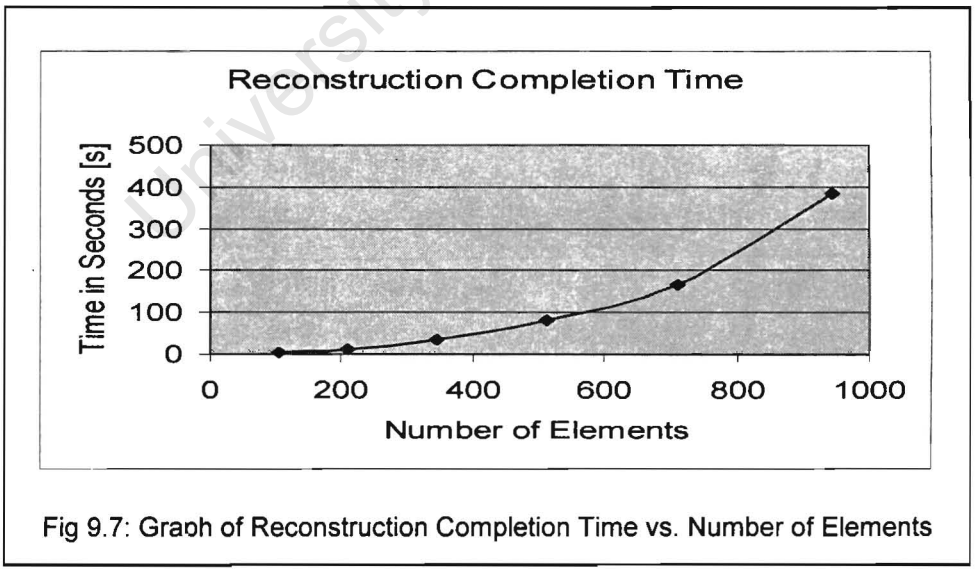


Fig 9.7: Graph of Reconstruction Completion Time vs. Number of Elements

Figure 9.7 depicts an exponential curve. The above measurements were taken with the use of Matlabs 'tic toc' functions. It should be noted that the above were computed on a 500 MHz PC.

9.1.7. Comparison of Images for Varying Size of Elements

This subsection presents the results produced for five meshes that differed by the size of elements.

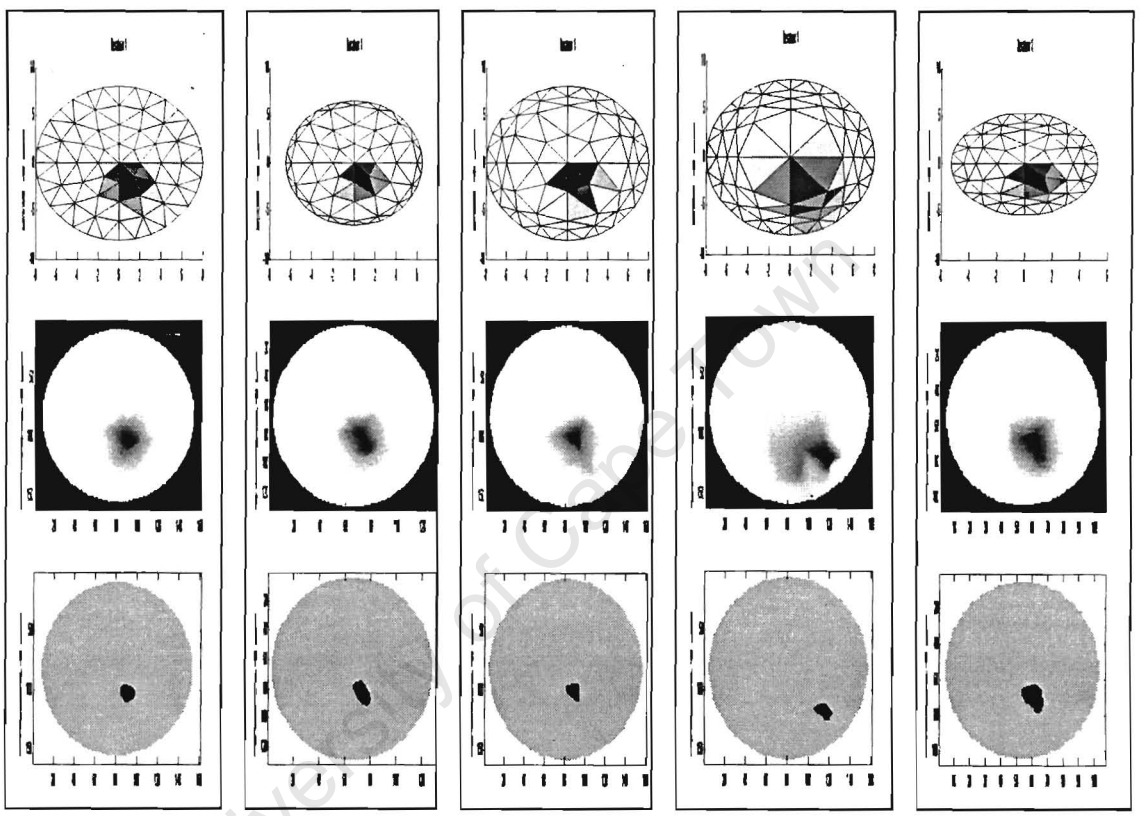


Fig 9.8: Comparison between Varying Element Sizes

The above results were produced for a mesh consisting of a 104 elements. Diagram (a) is a mesh with evenly sized elements, i.e. the aspect ratio of all elements is approximately unity. Diagram (e) in Figure 9.8 is the default mesh size. For this mesh the size of elements decrease gradually as the distance from the origin (centre) increases implying that slightly more emphasis is placed at the sensitive boundary regions.

9.1.8. Calibration Results

This section provides the results for data that has been calibrated with the schemes described in Chapter 6. Below are the boundary voltages that have been scaled and tested for outliers.

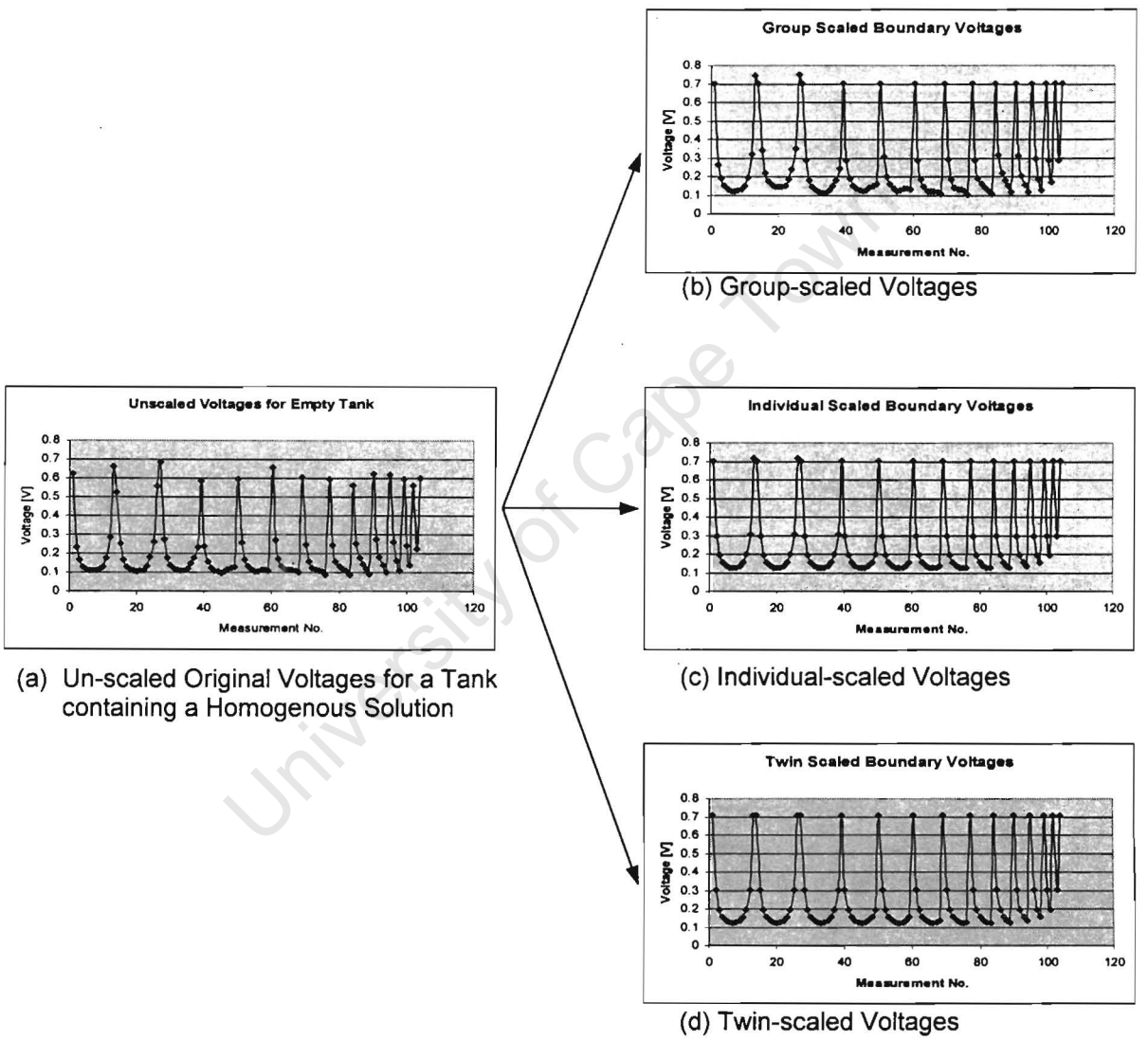
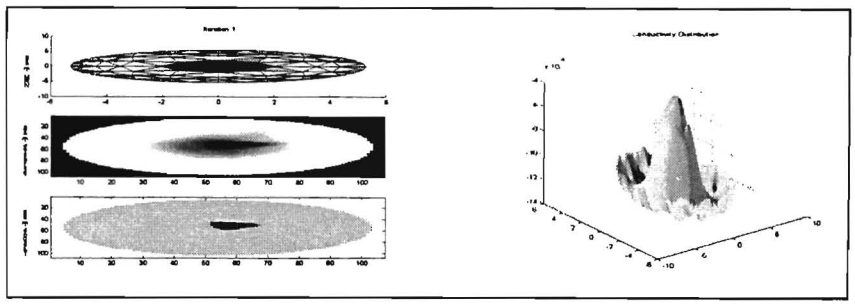


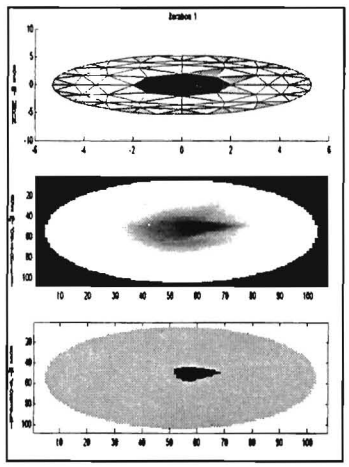
Fig 9.9: Results of Scaling Techniques on Boundary Voltages

Due to the errors described in Chapter 6, it is not possible to discern from Figure 9.9a if an object is present within the tank or not, but with scaling it can be seen that Diagrams (b), (c) and (d) of Figure 9.9 represent a homogenous tank.

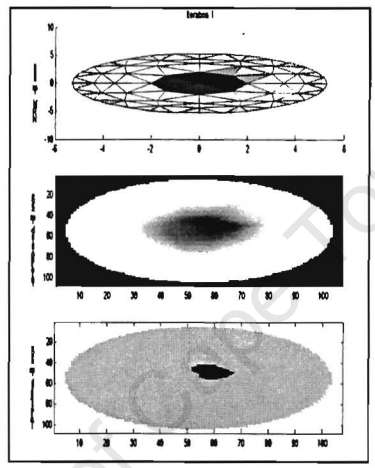
The pictures below depict images incorporating the various scaling techniques for an insulating rod in the tank.



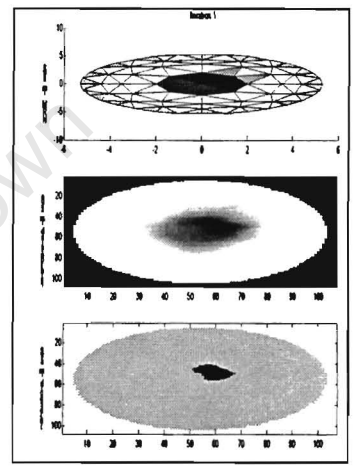
(a) Un-calibrated Data



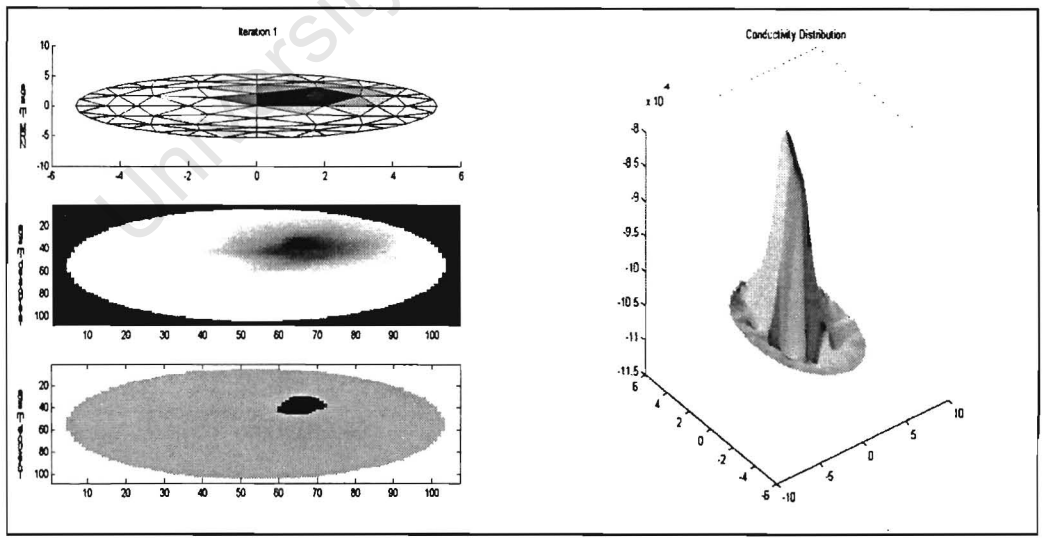
(b) Group-Scaling



(c) Individual-Scaling



(d) Twin-Scaling



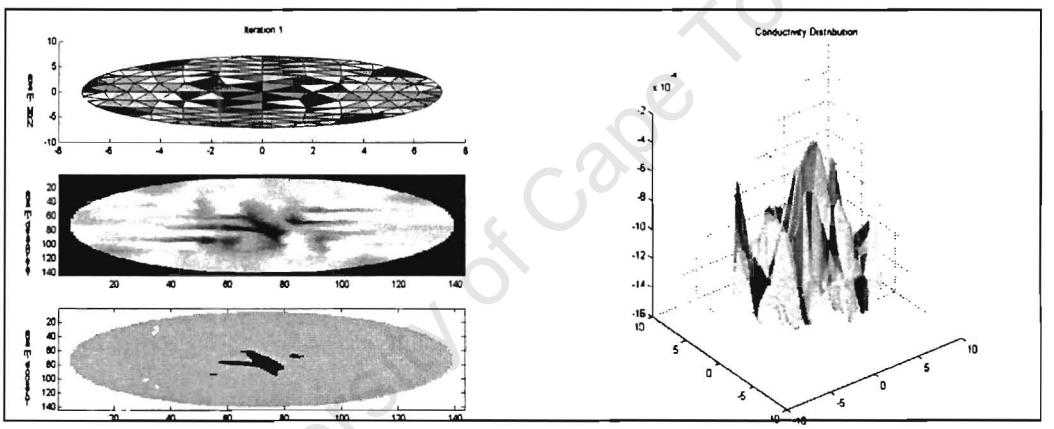
(e) Tomo_Imager_2D Calibrated Data

Fig 9.10: Results of Calibrated Data

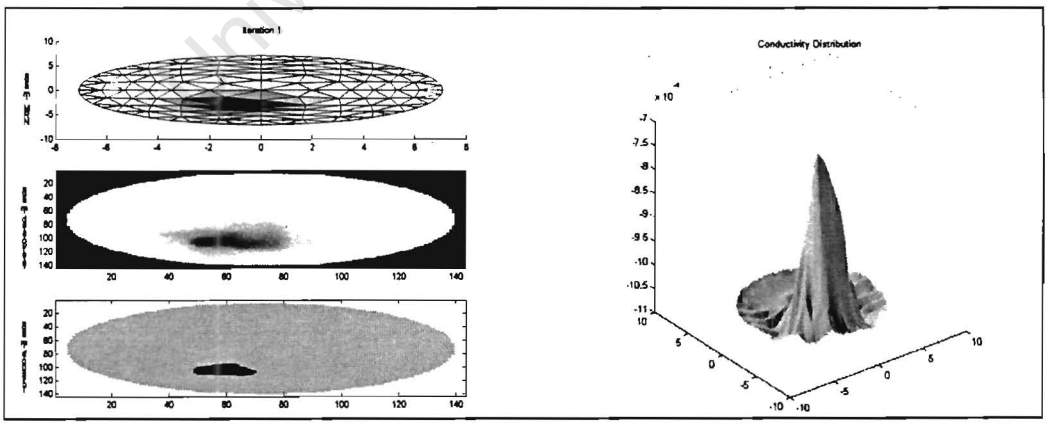
Figure 9.10a represents images for an un-calibrated data set. Diagram (b), (c) and (d) were scaled and then reconstructed with Tomo_Imager_2D. These

data sets were not calibrated with the Tomo_Imager_2D voltages. Diagram (e) shows the results when the boundary voltages were calibrated with the voltages of Tomo_Imager_2D. The calibration method is explained in Chapter 6.4. It should be noted that the 3D image of the conductivity distribution in Diagram (a) has fringing effects at the boundary but in Diagram (e) the fringing has been eradicated due to the calibration scheme.

Baffles are thin, long pieces of plastic that are fixed to the walls of the tanks. Their primary function is to reduce the vortexes created by the stirring of a impeller. The plastic insulating nature of baffles implies that it will alter the current flow in the tank. One method of accounting for these effects are with calibration and the resulting images are shown below.



(a) Images of data that have not been calibrated



(b) Images of data that have been calibrated

Fig 9.11: Calibrated and Un-calibrated Images of an Insulating Rod in a Tank with Baffles

The data for Figure 9.11a was not calibrated with Tomo_Imager_2D whilst Figure 9.11b was calibrated with Tomo_Imager_2D.

The power of calibration can further be demonstrated for metal impellers. In industry, tanks normally contain fixed metallic impellers. The properties of the metal are often very dominant and no useful information regarding the internal conductivities can be attained. In the image below, both a metallic impeller and insulating rod are located inside the tank.

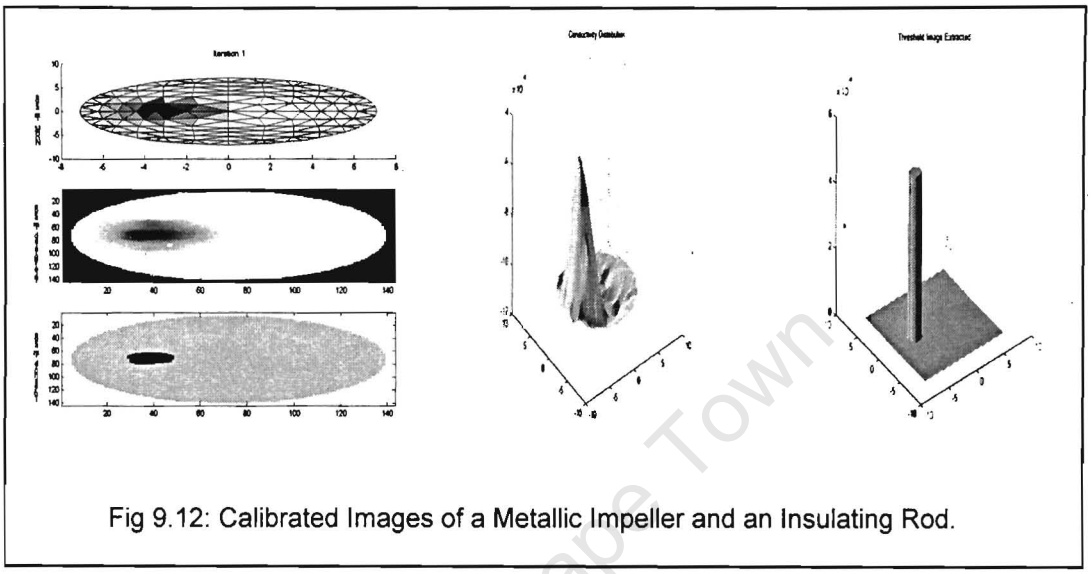


Fig 9.12: Calibrated Images of a Metallic Impeller and an Insulating Rod.

The reconstructed images in Figure 9.12 are for a tank, which contains both an insulating rod and a metallic impeller. The data has been calibrated with Tomo_Imager_2D to disregard the metallic impeller. The dark region in Figure 9.12 depicts the insulating rod. The metallic impeller is not present in these images.

The author is aware that the above calibration method is not the correct method to eradicate the presence of objects. A correct method would be to incorporate the information in the finite element method in the form of prior matrix. However the calibration method provides a much faster, and simpler method than physically modelling the impeller in the mesh.

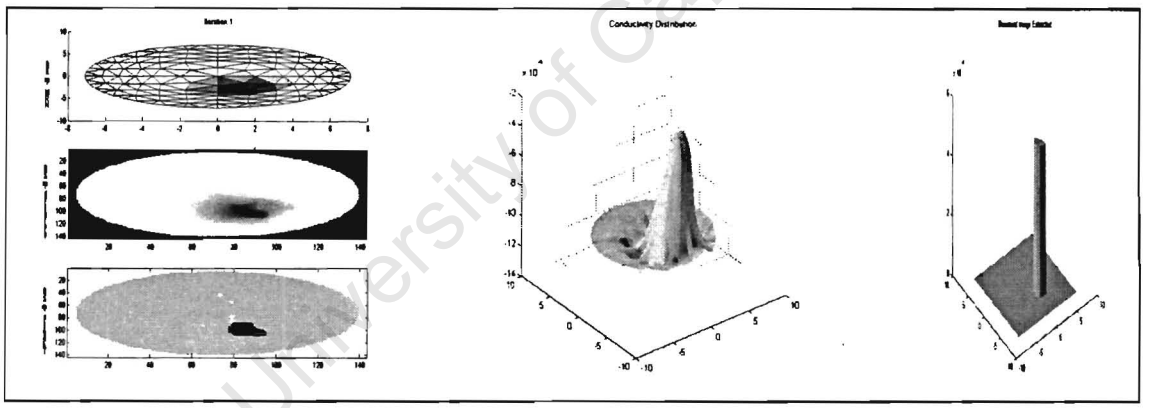
Further, the images produced by the calibration scheme reflects the electrical resistivity that is present in the medium. However the accuracy of this statement needs to be tested. This can be done by varying the conductivity of the medium (by changing the salt concentration) and reimaging.

9.2. Comparison of Two-Dimensional Results between NOSER and Jacobian Algorithm

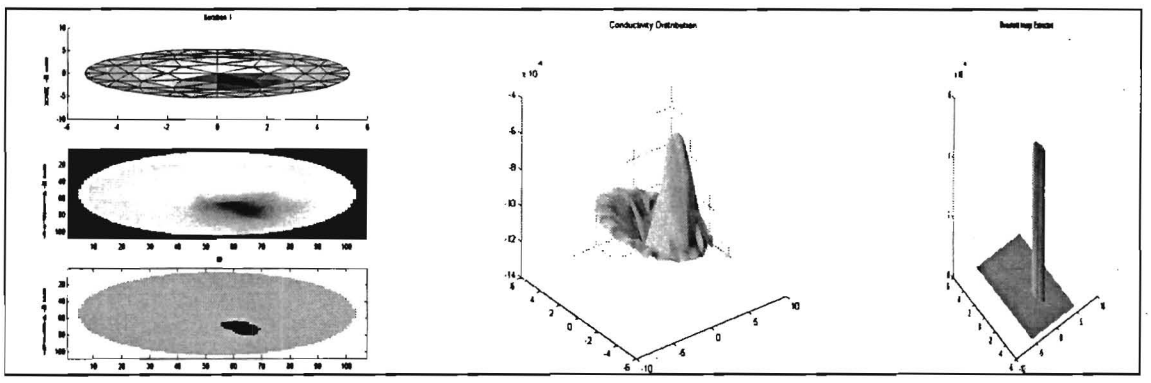
In this section comparison will be made between the NOSER algorithm described in Chapter 5.4 and the normal complete algorithm, which produced the results for Chapter 9.1 above. For brevity and clarity the normal algorithm will be referred to as the Jacobian algorithm.

9.2.1. Image Comparisons

Below are the reconstructed images from the Jacobian and NOSER algorithms for a tank containing an insulating rod.



(a) Images from Jacobian Algorithm



(b) Images from NOSER Algorithm

Fig 9.13: Comparative Images from Jacobian and NOSER Algorithms

The above images were produced after 10 iterations. It can be seen that the images in Diagram (a) of Figure 9.13 are clearer than the images in Diagram (b). The clarity differences can be explained by the 3D conductivity images. The 3D NOSER image depicts fringing at the edges.

9.2.2. Time Comparisons

This section compares the completion times between the two algorithms for meshes with varying number of elements.

Table 9.2: Comparison of Reconstruction Completion Times

	Completion Time in Seconds [s]	
Number of Elements	**Jacobian**	**NOSER**
104	15.9	5.1
208	53.1	9.9
344	141.2	18.8
512	313.7	36.1
712	624.4	65.7

The tabulated values in Table 9.2 are graphed below.

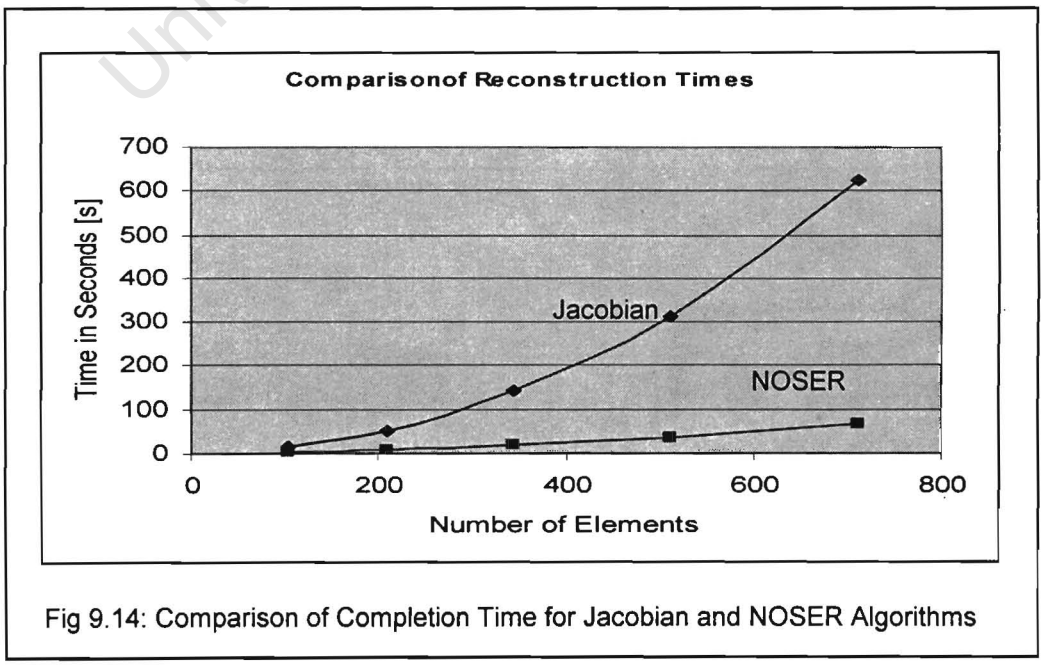


Fig 9.14: Comparison of Completion Time for Jacobian and NOSER Algorithms

The times displayed above were taken after 3 iterations. Both curves follow an exponential polynomial, with the Jacobian having a higher degree than the NOSER. The above times were captured with the aid of Matlabs 'tic toc'. It should be noted that the above were computed on a 500 MHz PC.

9.3. RESULTS OF THREE-DIMENSIONAL RECONSTRUCTION

The following subsections depict the ability of Tomo_Imager_3D to correctly reconstruct the conductivity distribution present in tanks.

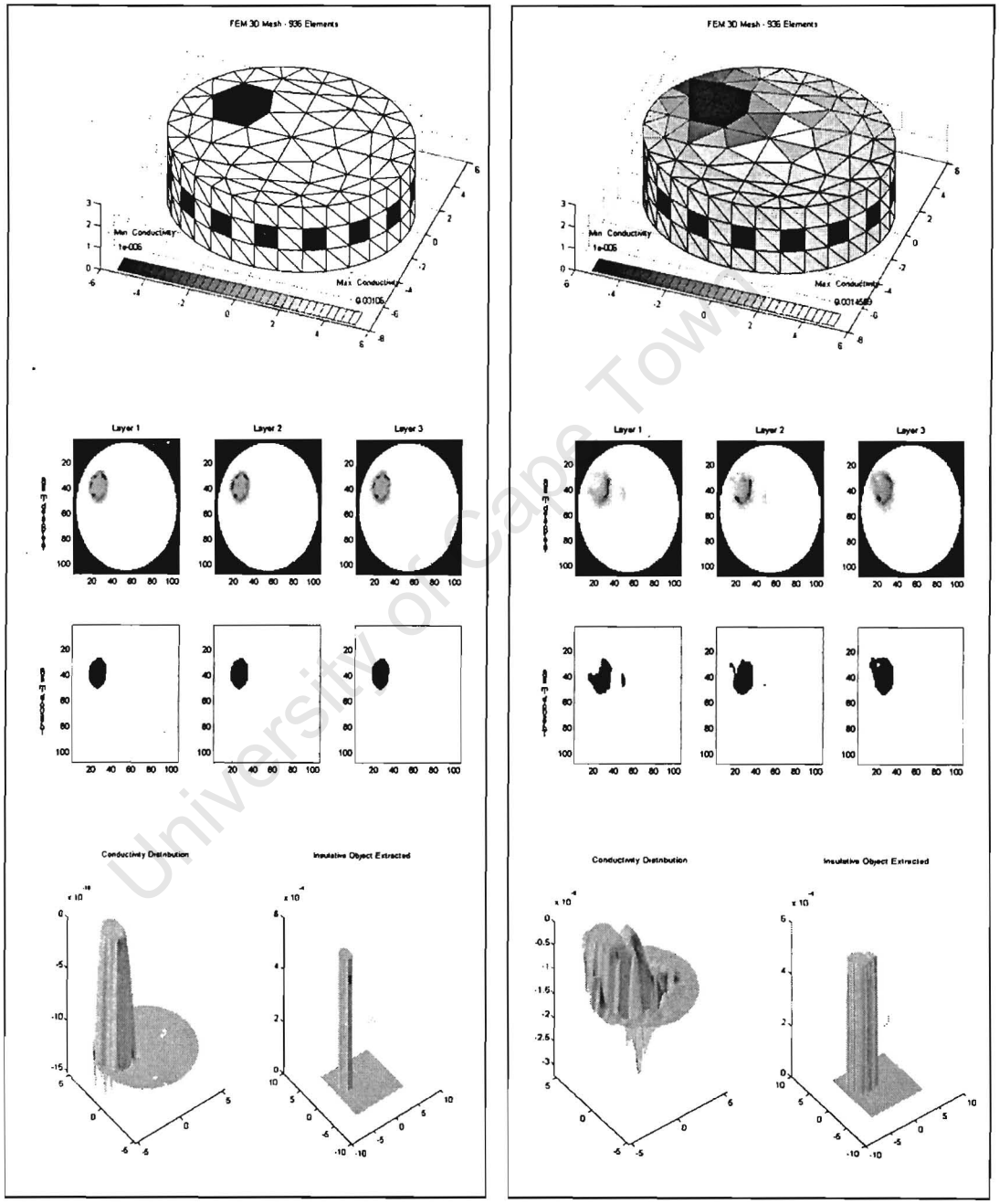
9.3.1. Reconstructed Images for Simulated Data

The reader is referred back to the flowchart in Figure 9.1 for an explanation of how the comparison between real and simulated data is made. For the three-dimensional reconstructions a mesh with three equally sized layers are used. The electrodes are located on the middle layer and are denoted on the mesh by darkly shaded rectangles.

The jump from 2D to 3D implied a significant increase in the computation time and complexity. To minimise this burden it was decided to provide results reconstructed with the NOSER algorithm only. Further, due to a lack of memory the author was not able to present results for meshes that contained more than 3096 elements.

The results for simulated data are displayed on the following page. A mesh consisting of 936 elements were used to reconstruct the user input conductivities. The top image in Figure 9.15 represents the Triangular Mesh Image. The images directly below the triangular mesh represent the

interpolated images and threshold images respectively. Each depicts three images for each layer of the mesh. This image is then given a three dimensional perspective revealing the possible shape of the object present in the tank.



(a) User Defined Conductivity (b) Reconstructed Images

Fig 9.15: Reconstructed Images for Simulated Data

Figure 9.15 represents the user input conductivities on the left and the corresponding reconstructed images on the right. The dark regions represent areas that are less conductive than the surrounding areas.

9.3.2. Reconstructed Images for an Insulating Rod

A mesh with 936 elements was used to reconstruct the images below.

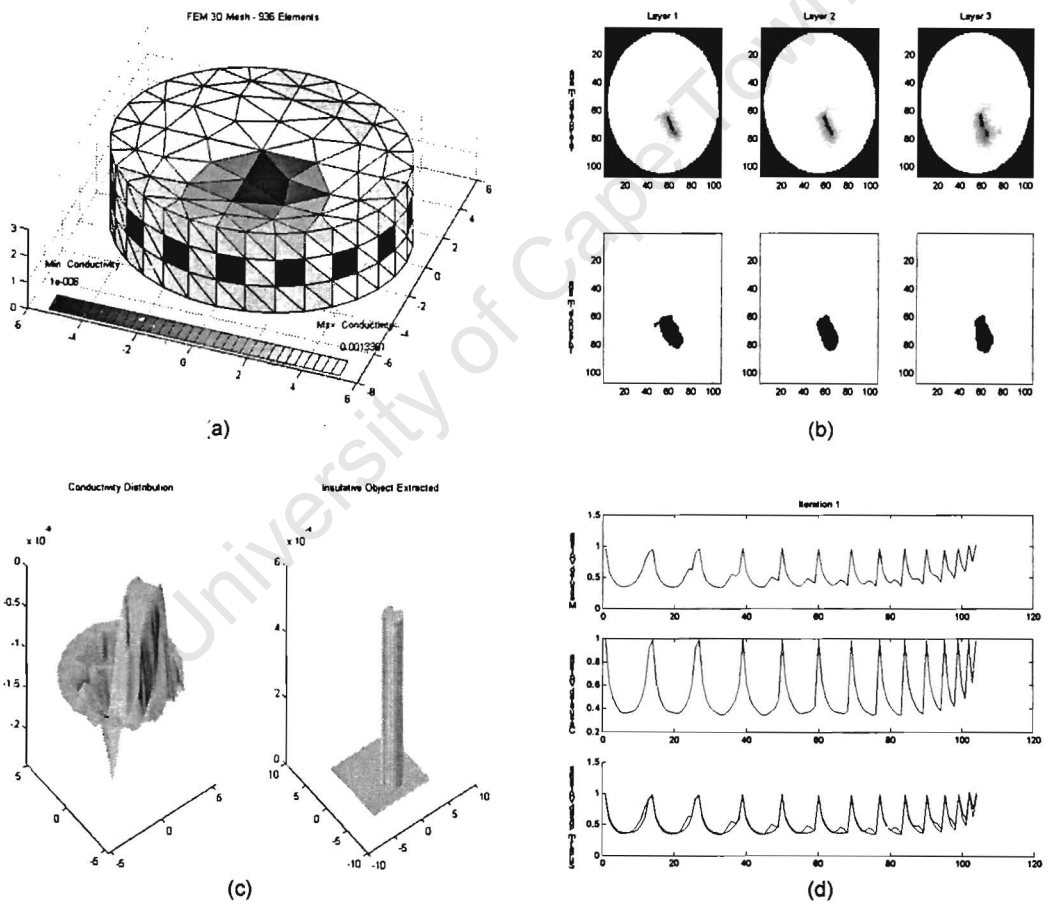


Fig 9.16: Reconstructed Images for an Insulating Rod

The dark region represents the insulating rod, which is the image that is shown in Diagram (c).

9.3.3. Reconstructed Images for Gaseous Flow

The images below represent a gaseous flow.

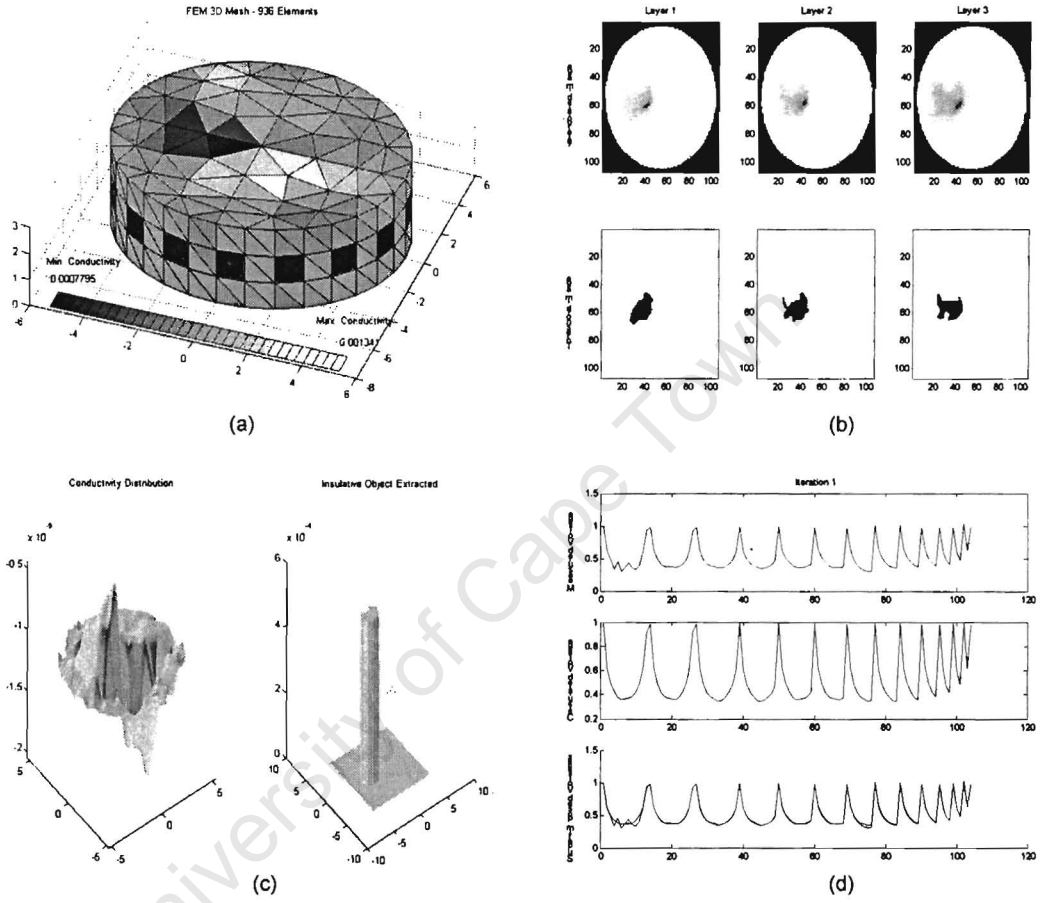


Fig 9.17: Reconstructed Images for Air Bubbles

The dark regions in Figure 9.17 show the concentration of air bubbles in the tank. The air bubbles were created by injecting pressurised air through a small opening located at the base of the tank. The air bubbles moved to the top in a relatively straight stream and thus represent the images of a rod. This can be seen by the images of the various layers.

9.3.4. Reconstructed Images for a Conductive Metal Rod

Metals are on the other side of the spectrum from the insulating rod. Metals can be considered a 'pure' conductor and would disturb the current flow in the tank. This subsection depicts the results of a metal rod placed in the tank.

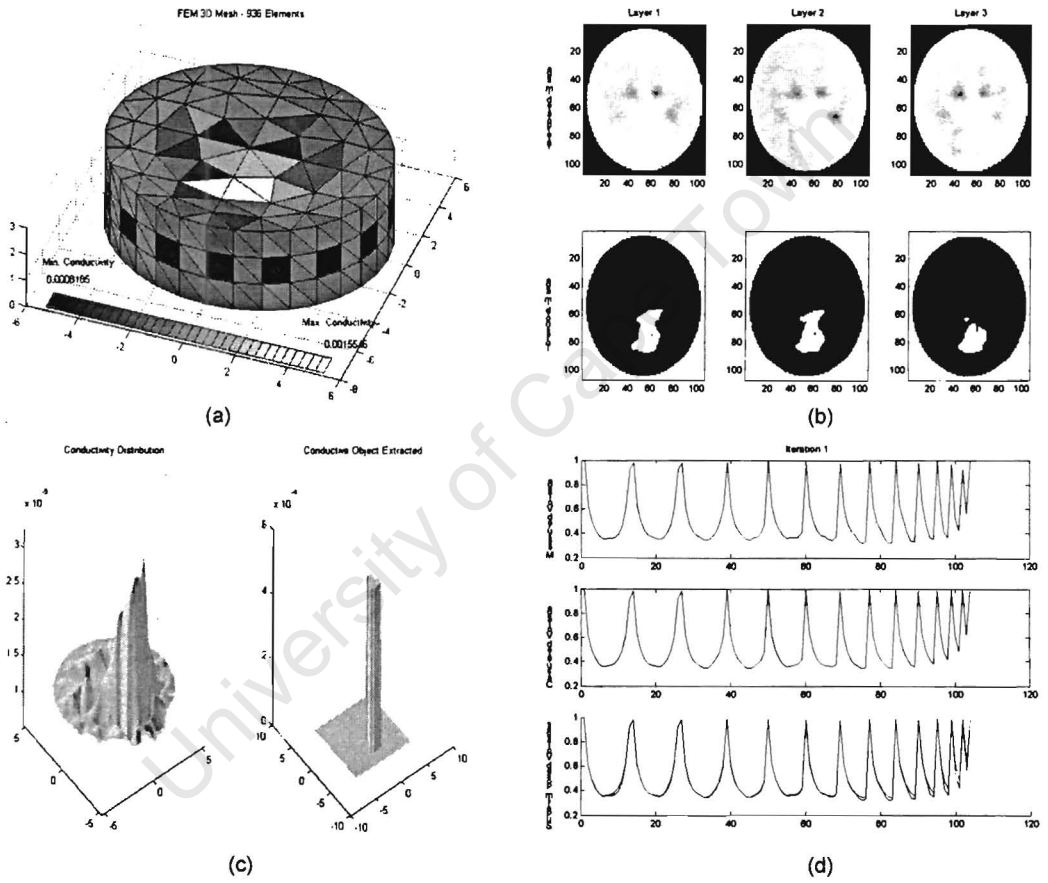


Fig 9.18: Reconstructed Images for a Metal Rod

The lightly shaded areas in Figure 9.18 represent the more conductive regions of the tank and indirectly the position of the metal rod. Note that Figure 9.18c was automatically altered in the program to highlight the region of interest. Normally the conductive region points downward but the image is flipped to provide a clearer visual perspective.

9.3.5. Comparison of Images for Varying Number of Elements

Three different meshes are used to compare the reconstructed images. These results are displayed below.

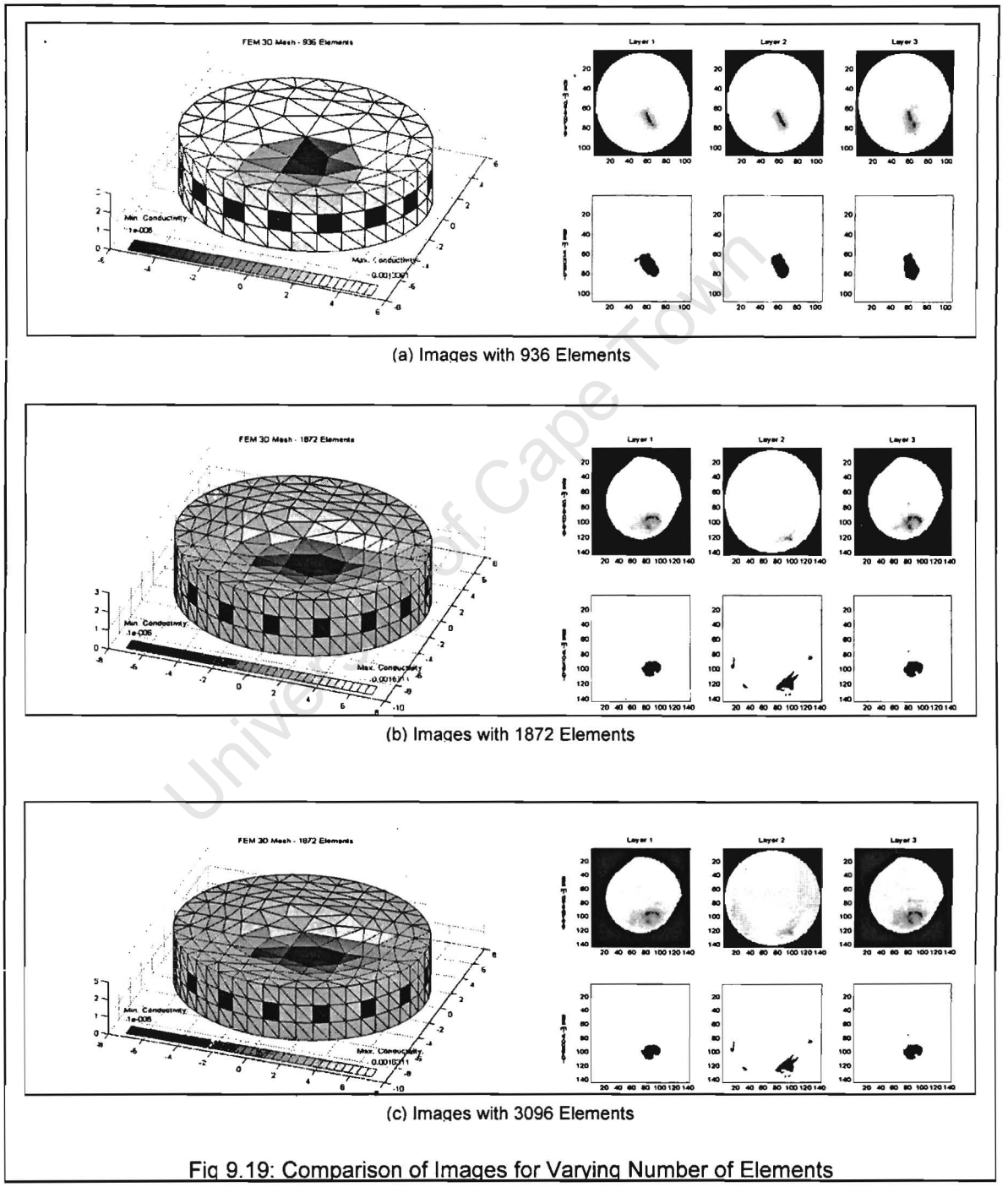


Fig 9.19: Comparison of Images for Varying Number of Elements

The images in Figure 9.19 were produced with the NOSER algorithm.

9.3.6. Comparison of Time of Completion for Varying Number of Elements

This section presents the timing results for meshes with varying number of elements.

Table 9.3: Time taken for Completion of Reconstruction Algorithm

Number of Elements	Time in Seconds
936	51.904
1872	310.186
3096	1259.224

The data values in Table 9.2 are plotted on a line graph below.

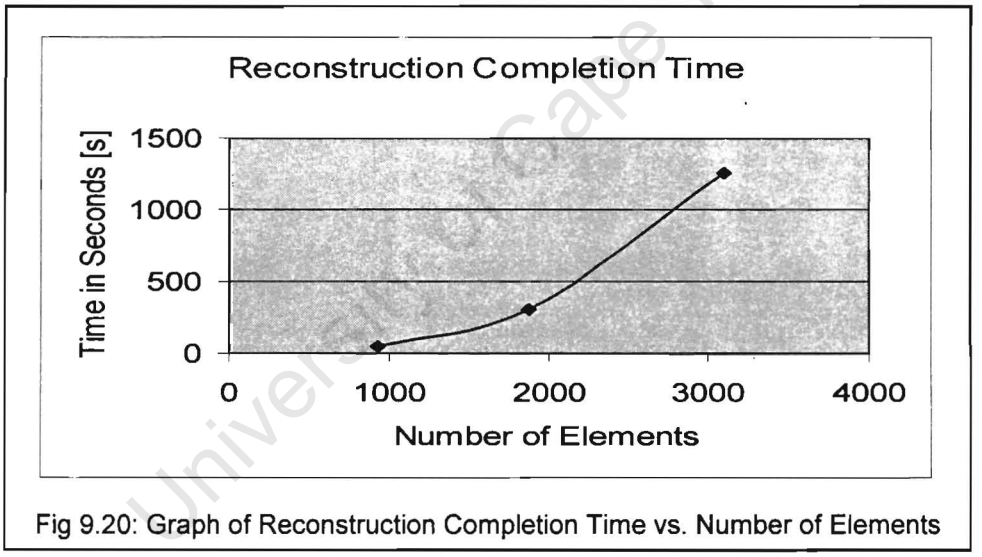


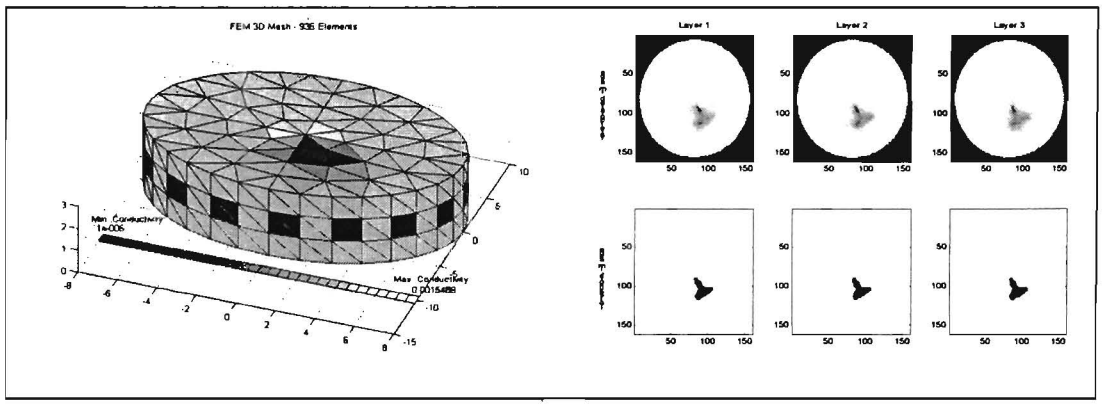
Fig 9.20: Graph of Reconstruction Completion Time vs. Number of Elements

Figure 9.20 depicts an exponential curve. The above measurements were taken with the use of Matlabs 'tic toc' functions. It should be noted that the above were computed on a 500 MHz PC.

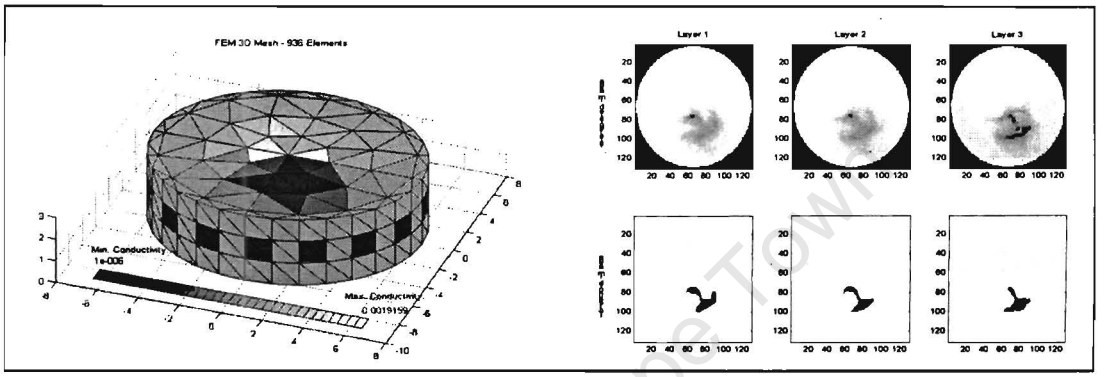
9.3.7. Comparison of Images for Varying Size of Elements

This subsection presents the results produced for four meshes that differed by the size of elements.

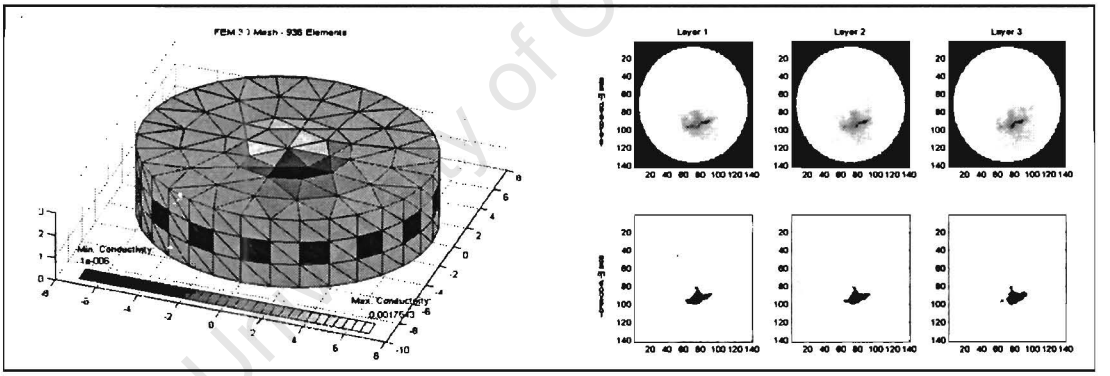
The results were produced for a mesh consisting of a 936 elements. In Figure 9.21 (on the next page) Diagram (a) is a mesh with evenly sized elements, i.e. the aspect ratio of all elements is approximately unity. Diagram (d) is the default mesh size. For this mesh the size of elements decrease gradually as the distance from the origin (centre) increases implying that slightly more emphasis is placed at the sensitive boundary regions. Diagram (b) has a mesh with small tetrahedrals modelled on the boundary. Diagram (c) mesh has small tetrahedrals modelled near the centre. These results were produced with the Jacobian algorithm.



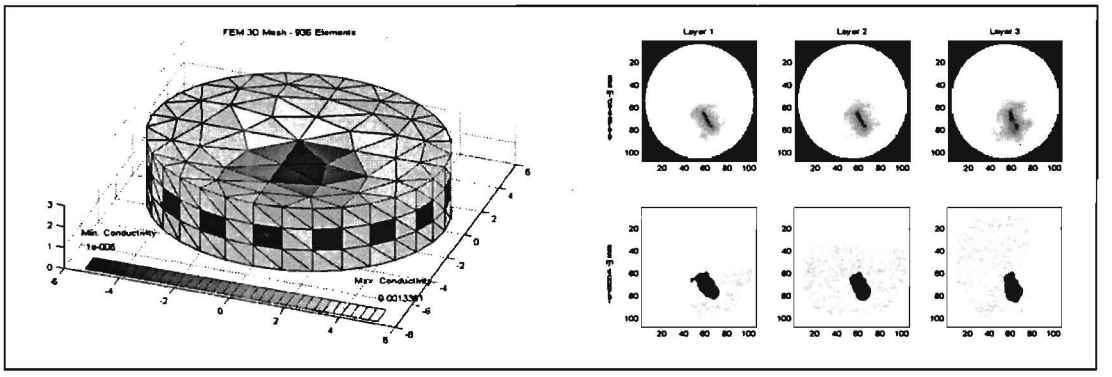
(a)



(b)



(c)



(d)

Fig 9.21: Comparison of Images for Varying Size of Elements

9.4. COMPARISON OF RESULTS BETWEEN 2D AND 3D ALGORITHMS

In this section comparisons between 2D and 3D images will be presented. The images presented in this subsection were produced in the following manner. Firstly, a 3D conductivity distribution is specified (Figure 9.23a). This conductivity distribution is passed through the 3D forward solution resulting in a boundary voltage that represents this particular conductivity distribution. This boundary voltage V_{cal} is then reconstructed with both the 2D and 3D reconstruction algorithms.

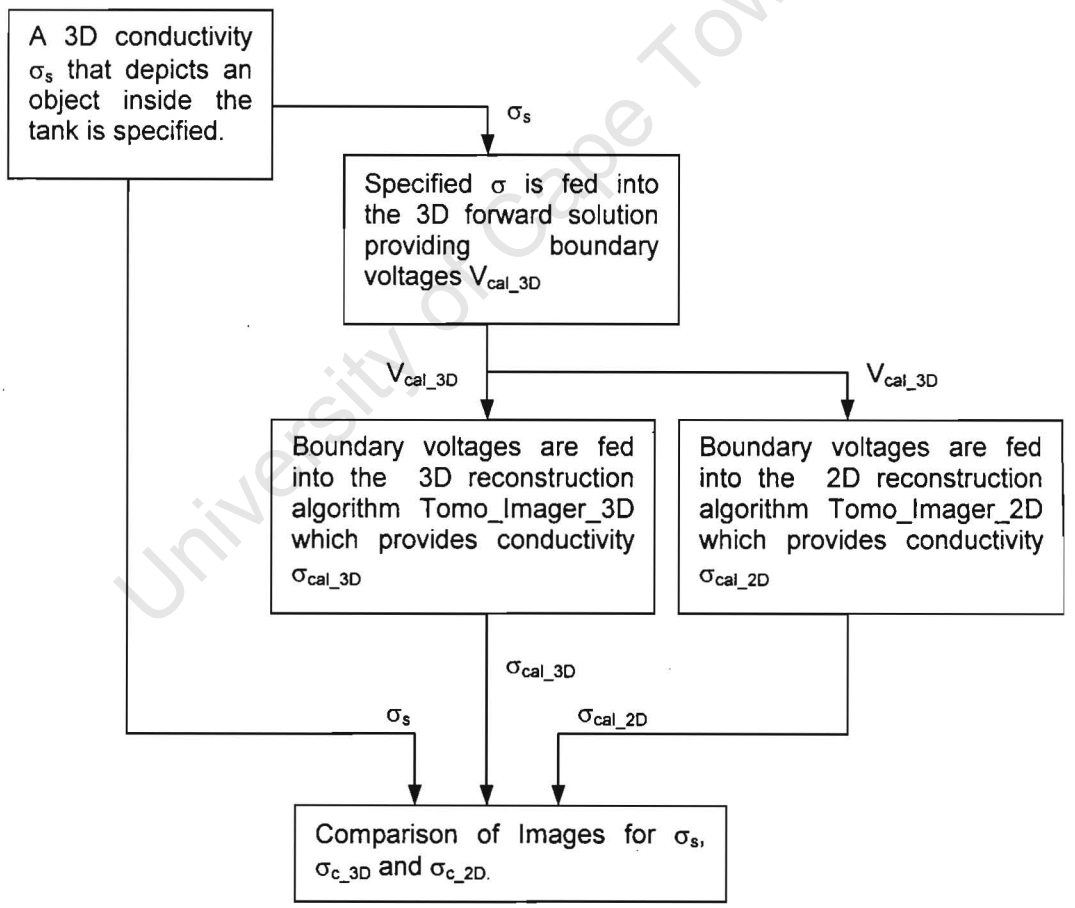
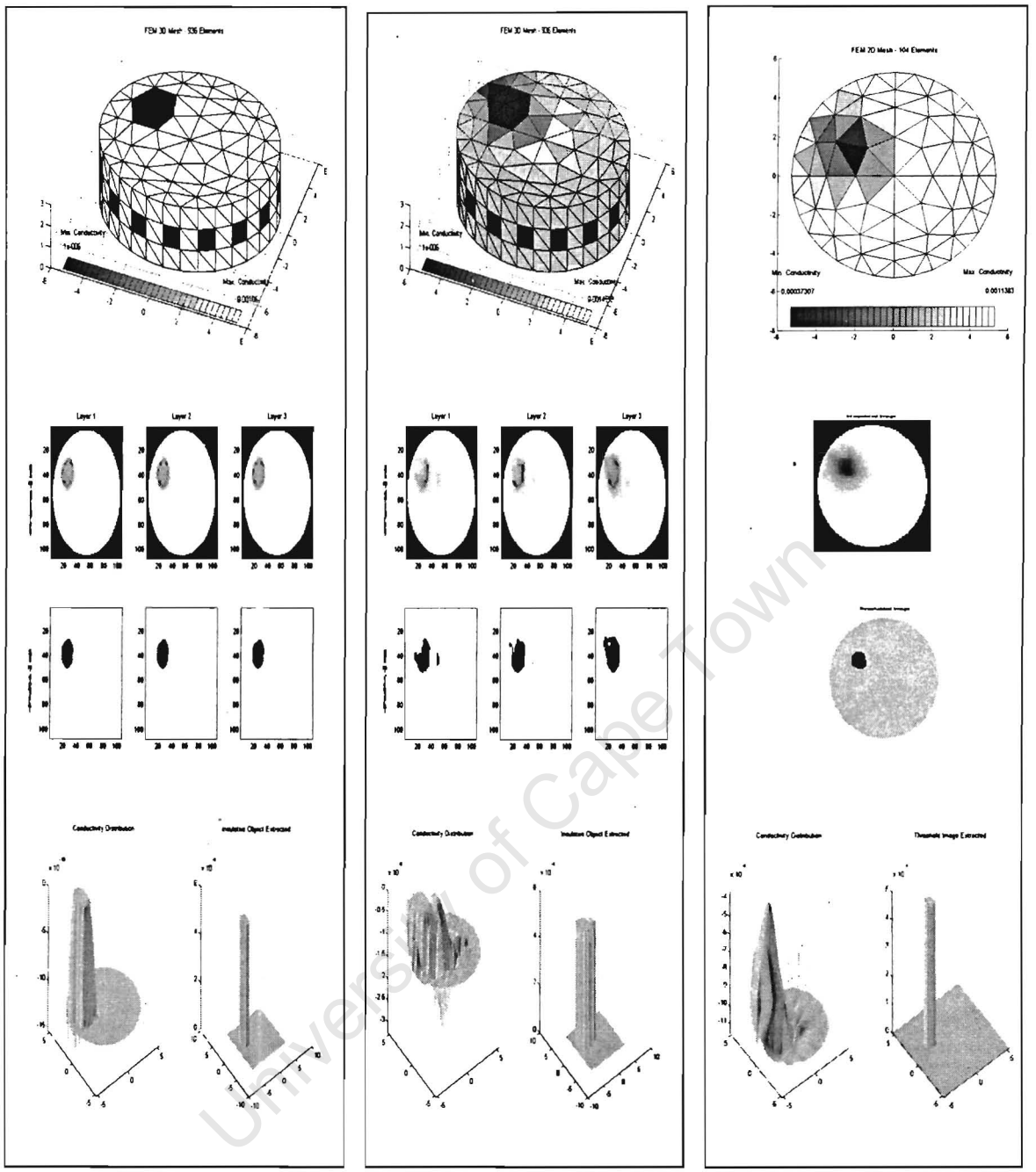


Fig 9.22: Flowchart Explanation of 3D Reconstructed Simulated Data

The resulting images can be seen in Figure 9.23b and Figure 9.23c on the next page.



(a) User Defined Conductivity (b) 3D Reconstructed Images (c) 2D Reconstructed Images

Fig 9.23: Reconstructed Images for Simulated Data

In the above Figure, Diagram (a) represents the simulated data that is specified by the user. Diagram (b) represents the reconstruction of this simulated data with the 3D algorithm. Lastly, Diagram (c) represents the reconstruction of the simulated data with the 2D algorithm.

10. CONCLUSIONS

Based on the findings of this research, the following conclusions have been drawn:

- **The Two-Dimensional Reconstruction Algorithm 'Tomo_Imager_2D' has been Successfully Implemented**

  Based on the results presented in Chapter 9, the author can conclude that the two-dimensional algorithm has been correctly implemented. Tomo_Imager_2D is able to image the location and relative size of objects present in the tank. The algorithm achieved an objective of the project, namely: Convert the boundary voltages to the internal conductivity distribution that can be displayed on a monitor showing relative conductivity of a cross section of the measurement tank.

- **The Three-Dimensional Reconstruction Algorithm 'Tomo_Imager_3D' has been Successfully Implemented**

  Based on the results presented in Chapter 9, the author can conclude that the three-dimensional algorithm has been correctly implemented. Tomo_Imager_3D is able to image the location and relative size of objects present in the tank. As with the 2D algorithm, the 3D algorithm achieved the objective of converting the boundary voltages to the internal conductivity distribution that can be displayed on a monitor showing relative volume conductivity of the tank.

- **Increasing the Number of Elements Improves the Accuracy and Resolution of the Reconstructed Images**

  The images presented in Section 9.1.5 show the theory that increasing the number of elements in a mesh improves the resolution as well as the accuracy. The author is aware that the above statement becomes false as a limit for improvement is reached. This is aptly explained with the aid of Figure 3.5 in Section 3.3.1. Further the images produced with the 3D algorithm in Section 9.3.5 does not comparatively show as much improvement as the 2D algorithm. This could be attributed to increased computational burdening of the 3D algorithm and numerical rounding errors.

- **Increasing the Number of Elements Increases the Convergence Time of the Reconstruction Algorithms**

  The clearer images produced by increasing the number of elements is negated by the increase in computation time. This is in compliance with the theory discussed in Chapter 3.3. From Figure 9.7 and Figure 9.20 it can be concluded that the 2D and 3D algorithms display an exponential increase in completion time as the number of elements in the mesh increase. For practical imaging with sixteen electrodes, a 2D grid consisting of 208 elements was found to be a good compromise between execution time and resolution / accuracy. This grid takes approximately 13 seconds per Newton-Raphson iteration with a Celeron 500 MHz.

- **The NOSER Algorithm Provides Faster Reconstruction with Minimal Degradation of Images**

  The results in Section 9.2 show that the NOSER algorithm produces images that can provide accurate information about the contents of the tank. The images in comparison with the normal Jacobian algorithm are slightly less defined but are satisfactory. In compensation, Figure 9.14 shows that the NOSER algorithm produces these images at a much faster rate.

- **Calibrating with the FEM – Predictive method Outperforms other Calibration Techniques Developed**

  Of the four calibration techniques developed, the FEM – Predictive method (described in Section 6.4) performed the best. Further, all the scaling techniques provide an improvement on the images from the original data set in varying degrees. The images in Figure 9.10 and Figure 9.11 shows that the calibration improves the object location in the tank as well as the clarity of the object shape.

- **The FEM – Predictive Calibration Scheme Partially Compensates for the Presence of Baffles and Impellers in the Tank**

  The images in Figure 9.12 further shows the potential of the calibration scheme. Objects such as impellers and other internal objects of the tank that are of no interest can be 'removed' from the image by calibrating the data with the software. As previously mentioned the author is aware that this is not the optimal manner in which to compensate for objects such as

impellers and baffles but the calibration scheme is easier to implement than modelling a impeller into the FEM mesh.

- **Fast Imaging with High Accuracy can be achieved by Parallelizing the Algorithm**

The study presented in Chapter 7 shows the suitability of the algorithm for parallelization. By distributing the processes over a various number of computers the reconstruction algorithm can easily be sped up and this will allow for online imaging to be a practical situation. With parallelizing it can be expected that a reconstruction rate of 's x 1' images per second can be attained, where 's' is the number of computers that are linked together.

11. RECOMMENDATIONS

Based on the findings and conclusions in this thesis, the following recommendations can be put forward.

1. Recommendations for Speed Improvement

The following recommendations are made to improve the speed of the algorithm.

- **Parallelize the Algorithm**

  Parallelizing the algorithm will allow for a higher reconstruction rate to be achieved. It will also enable the implementation of meshes with more elements to be realised and hence better resolutions to be achieved without a great expense of time.

- **Implement a Bandwidth Node-Numbering Scheme**

  Implementing a node-numbering scheme will reduce the memory requirement of the algorithm. By freeing up more memory, the efficiency of the algorithm is improved and consequently the reconstruction time will be reduced.

2. Recommendations for Improving Image Resolution

The following recommendations are made to improve the image resolution of the algorithm.

- **Implement Higher-Order Elements and Interpolation Functions**

  As explained in Chapter 3.3 the use of higher order elements improves the modelling of mesh. The modelling improvement implies that the accuracy of the solution is improved and thus a clearer image and more accurate conductivity representation can be attained.

- **Increase the Number of Electrodes on the Boundary**

  According to Equation 2.10 increasing the number of electrodes on the boundary increases the number of independent boundary voltages. With a greater number of independent boundary voltages a higher image resolution is achievable.

3. Adapt the Algorithm to Model Differently Shaped Boundaries

At present the boundary is limited to a circular tank in 2D and a cylinder in 3D. The algorithm can be adapted to model alternately shaped boundaries, which implies that the application area can be broadened.

4. Develop an Online System

Currently images are captured via the hardware system and then transferred to the required files to be imaged by the software. The images are not processed in real time. An online system can be developed by integrating this software with the hardware system. Parallelization of this algorithm will increase the possibility of an online system to be developed.

5. Develop a Three Dimensional Current Injection and Voltage Measurement Strategy

In simulated 3D imaging system the measurement strategy used was the same adjacent pair method used for existing sixteen electrode 2D system. A real 3D system with multiple electrode layers is presently being developed at UCT. For this system it is recommended that a different measurement strategy, which will measure between different electrode planes, be employed. There are numerous current injection strategies used for 3D. These strategies will need to be researched and one method chosen and implemented.

6. Test the Accuracy of the Reconstruction Algorithm to Measure Conductivity Values of the Solution

Presently, the imagers are optimised to extract objects location and relative size within the tank. A complete test of the actual conductivity comparisons between the simulated data and the real data has not been studied. Based on the results presented in this thesis the author is confidant that the conductivity results produced by the imager would closely follow those of the real conductivity.

12. REFERENCES

Atkinson K, (1985). *'Elementary numerical analysis'.* New York: John Wiley and Sons.

Avis NJ, Barber DC, (1994). *'Image reconstruction using non-adjacent drive configurations'.* Physiological Measurement. Vol. 16, pp. 153-160, cited by Kolehmainen (2001).

Baldermo J, (2002). Programmer of the Parallel Virtual Machine Tool Box and Message Passing Interface Toolbox for Matlab. University of Granada, Spain, as cited by Goasguen.

Barber DC, Brown BH, Freston IL, (1983). *'Imaging spatial distributions of resistivity using applied potential tomography'.* Electron. Lett. Vol. 19, pp. 993-995, cited by Randall et al (2001).

Barber DC, (1989). *'A review of image reconstruction techniques for electrical resistance tomography'.* Med. Phys. Vol. 16, pp. 162-169, cited by Kolehmainen (2001).

Belward C, (1997). *'Electric impedance tomography: Solving the forward problem'*, pp. 1-12. http://www.maths.uq.edu.au/~cb/anziam97/anziam97.html

Borse GJ, (1997). *'Numerical methods with Matlab: A resource for scientists and engineers'.* Boston: PWS Publishing Company.

Borsic A, McLeod CN, Lionheart WRB, (2001). *'Total variation regularisation in EIT reconstruction'.* 2nd World Congress on Industrial Process Tomography. Hanover, Germany, pp. 433-440.

Breckon WR, (1990). *'Image reconstruction in electrical impedance tomography'*, PhD. Thesis, School of Computing and Mathematical Sciences - Oxford Polytechnic.

Brown BH, Seagar AD, (1985). *'The Sheffield data collection system'.* Clin. Phys. Physiol. Meas. Vol. 8, Supplement A, pp. 36-46, cited by Vauhkonen (1997).

Brown BH, (2001). *'Medical impedance tomography and process impedance tomography: a brief review'.* Meas. Sci. Technol. Vol. 12, No. 8, pp. 991-996.

Burnett DS, (1987). *'Finite element analysis, from concepts to applications'.* Addison-Wesley.

Chapra SC, Canale RP, (1988). *'Numerical methods for engineers'.* New York: McGraw-Hill Book Company.

Cheney M, Isaacson D, Newell JC, Simske S, Goble J, (1990). *'Noser: An algorithm for solving the inverse conductivity problem'.* International journal of imaging systems and technology. Vol. 2, pp. 66-75.

Cheney M, Isaacson D, Newell JC, (1999). *'Electrical impedance tomography'.* SIAM Review. Vol. 41, No. 1, pp 85-101.

Cheng KS, Isaacson D, Newell JC, Gisser DG, (1989). *'Electrode models for electric current computed tomography'.* IEEE Trans. Biomed. Eng. Vol. 36, pp. 918-924, as cited by Vauhkonen (1997).

Cilliers JJ, Xie W, Neethling SJ, Randall EW, Wilkinson AJ, (2001). *'Electrical resistance tomography using a bi-directional current pulse technique'.* Measurement Science and Technology. Vol. 12, No. 8, pp. 997-1001.

Cook RD, (1995). *'Finite element analysis for stress analysis'.* USA: John Wiley and Sons.

Cook RD, Malkus DS, Plesha ME, Witt RJ, (2002). *'Concepts and applications of finite element analysis'.* Fourth Edition. USA: John Wiley and Sons.

Dickin F, Pinheiro P, (1996a). *'Introduction to qualitative reconstruction algorithms'.* pp. 1-37.

Dickin F, Wang M, (1996b). *'Electrical resistance tomography for process applications'.* Measurement Science and Technology, Vol. 7, pp. 247-260.

Durrett D, (2002a). *'A high speed data caption system for electrical resistance tomography'.* Undergraduate Thesis, Department of Electrical Engineering, University of Cape Town.

Durrett D, (2002b). Personal Communication. 30 October 2002.

Fransolet E, Crine M, L'Homme G, Toye D, Marchot P, (2002). *'Electrical resistance tomography sensor simulations: Comparison with experiments'.* Measurement Science and Technology, Vol. 13, No 8, pp. 1239-1247.

Gambolati G, (1980). *'Fast solution to finite element flow equations by Newton iteration and modified conjugate gradient method'.* International Journal for Numerical Methods in Engineering, Vol. 15, pp. 661-675.

Geist A, Beguelin A, Dongarra J, Jiang W, Manchek R, Sunderam V, (1994). *'PVM 3 user's guide and reference manual'.*

Gencer NG, Ider YZ, Kuzuoglu M, (1992). *'Electrical impedance tomography using induced and injected currents'.* Clin. Phys. Physiol. Meas. Vol. 13, Supplement A, pp. 95-99, cited by Vauhkonen (1997).

Gencer NG, Ider YZ, Williamson SJ, (1996). *'Electrical impedance tomography: Induced-current imaging achieved with a multiple coil system'.* IEEE Trans. Biomed. Eng. Vol. 43, pp. 139-149, cited by Vauhkonen (1997).

Gencer NG, Kuzuoglu M, Ider YZ, (1994). *'Electrical impedance tomography using induced currents'.* IEEE Trans. Biomed. Eng. Vol. 13, pp. 338-350, cited by Vauhkonen (1997).

George PL, (1991). *'Automatic mesh generation: Application to the finite element methods'.* Masson: John Wiley & Sons Ltd.

Gisser DG, Isaacson D, Newell JC, (1988). *'Theory and performance of an adaptive current tomography system'.* Clin. Phys. Physiol. Meas. Vol. 9, Supplement A, pp. 35-41, cited by Cheney et al (1999).

Gisser DG, Isaacson D, Newell JC, (1990). *'Electric current computed tomography and eigenvalues'.* SIAM Journal of Applied Mathematics. Vol. 50, pp. 1623-1634, cited by Cheney et al (1999).

Goasguen S, (2002a). *'How to install and use the parallel virtual machine toolkit PVMTB'.* Presentation on 15 May 2002.

Goasguen S, (2002b). *'How to install and use the message passing interface toolkit MPITB'.* Presentation on 26 September 2002.

Golub G, Hansen C, (1999). *'Tikhonov regularization and total least squares'.* SIAM Jour. Matrix Anal. Appl. Vol. 21, pp. 185-194.

Grootveld CJ, Segal A, Scarlett B, (1997). *'Regularized modified Newton-Raphson technique applied to electrical impedance tomography'.* International Journal of Imaging Systems and Technology. Vol. 9, pp. 60-65.

Hahn BD, (1997). *'Essential Matlab for scientists and engineers'.* Second Edition. South Africa: Prentice Hall South Africa.

Hageman LA, Young DM, (1981). *'Applied iterative methods'.* New York: Academic Press.

Healey TJ, Tozer RC, Freeston IL, (1992). *'Impedance imaging of 3D objects using magnetically induced currents'.* 14th Int. Conf. IEEE Eng. Med. Biol. Society. pp. 1754-1755, cited by Vauhkonen (1997).

Heikkinen LM, Vauhkonen M, Savolainen T, Kaipio JP, (2001a). *'Modelling of internal structures and electrodes in electrical process tomography'.* Measurement Science Technology. Vol. 12, pp. 1012-1019.

Heikkinen LM, Vauhkonen M, Savolainen T, Leinonen K, Kaipio JP, (2001b). *'Electrical process tomography with known internal structures and resistivities'.* Inv. Probl. Eng. Vol. 9, pp. 431-454.

Heikkinen LM, Vilhunen T, West RM, Vauhkonen M, (2002). *'Simultaneous reconstruction of electrode contact impedances and internal electrical properties: II. Laboratory experiments'.* Measurement Science Technology. Vol. 13, pp 1855-1861.

Hooper J, (2001). *'Embedded controller for tomography measurements '.* Undergraduate Thesis, Department of Electrical Engineering, University of Cape Town.

Hua P, Webster JG, Thompkins WJ, (1993). *'Effect of measurement method on noise handling and image quality of EIT imaging'.* 9th Int. Conf. IEEE Eng. Med. Biol. Society. pp. 1429-1430, cited by Vauhkonen (1997) and Dickin et al (1996b).

Hua P, Woo EJ, (1990). *'Reconstruction algorithms - electrical impedance tomography'*, ed. J.G. Webster, The Adam Hilger Series on Biomedical Engineering, Bristol: IOP Publishing Ltd. pp.97-136.

Kaipio JP, Kolehmainen V, Vauhkonen M, Somersalo E, (1999). *'Construction of non-standard smoothness priors'.* Inverse Problems. Vol. 15, pp. 713-729.

Karjalainen PA, Kaipio JP, Koistinen AS, Vauhkonen M, (1999). *'Subspace regularization method for the single trial estimation of evoked potentials'.* IEEE Trans. Biomed. Eng. Vol. 46, pp. 849-860.

Kim KY, Kim BS, Kim MC, Lim YJ, Vauhkonen M, (2001). *'Image reconstruction in time-varying electrical impedance tomography based on*

the extended Kalman filter'. Meas. Sci. Tech. Vol. 12, No 8, pp. 1032-1039.

Kleinermann F, Avis NJ, Alhargan FA, (1997). *'Analytical solution to the forward problem for a finite elliptical cylinder'*. World Congress on Medical Physics and Biomedical Engineering. Vol. 35, pp. 322-323.

Kleinermann F, Avis NJ, Alhargan FA, (2000). *'Analytical solution to the three dimensional electrical forward problem for a circular cylinder'*. Inverse Problems. Vol. 16, pp. 461-468.

Kleinermann F, Avis NJ, Alhargan FA, (2001). *'Analytical solution to the three dimensional electrical forward problem for an elliptical cylinder'*. 2nd World Congress on Industrial Process Tomography. Hanover, Germany, pp. 787-794.

Kolehmainen V, Vauhkonen M, Karjalainen PA, Kaipio JP, (1997a). *'Assessment of errors in static electrical impedance tomography'*. Physiol. Meas. Vol. 18, No. 4, pp. 289-303.

Kolehmainen V, Vauhkonen M, Karjalainen PA, Kaipio JP, (1997b). *'Spatial inhomogeneity and regularization in EIT'*. IEEE IEMBS. Chicago, USA.

Kolehmainen V, (2001). *'Novel approaches to image reconstruction in diffusion tomography'*. Doctoral Thesis, Department of Applied Physics, University of Kuopio.

Kwon YW, (1997). *The finite element method using Matlab*. Boston: CRC Press.

Le Hyaric A, Pidcock MK, (2002). *'An image reconstruction algorithm for three dimensional electrical impedance tomography'*. Accessed in April 2002 from: http://citeseer.nj.nec.com/398204.html.

Le Hyaric A, Pidcock MK, (2000). *'A one step image reconstruction algorithm for electrical impedance tomography in three dimensions'*. Physiol. Meas. Vo. 21, No. 1, pp. 95-98.

Logan DL, (2002). *'A first course in the finite element method'.* Third Edition. USA: Brooks/Cole Thomson Learning.

Loh WW, Dickin FJ, (1996). *'Improved modified Newton-Raphson algorithm for electrical impedance tomography'.* Electronic Letters. Vol. 32, No. 3, pp. 206-207.

Matlab, (2000). Mathematical programming language developed by *The MathWorks, Inc.* Version 6.0.0.88, Release 12.

Matlab, (2000b). Built in Help Function. Version 6.0.0.88, Release 12.

'Meshing Research Corner, The', http://www.andrew.cmu.edu/user/sowen/mesh.html

Molinari M, Cox SJ, Blott BH, Daniell GJ, (2001a). *'Efficient non-linear 3D electrical tomography reconstruction'.* 2nd World Congress on Industrial Process Tomography. Hanover, Germany, pp. 424-432.

Molinari M, Fangohr H, Generowicz J, Cox SJ, (2001b). *'Finite element optimisations for efficient non-linear electrical tomography reconstruction'*, 2nd World Congress on Industrial Process Tomography. Hanover, Germany, pp. 406-417.

Molinari M, Cox SJ, Blott BH, Daniell GJ, (2002). *'Comparison of algorithms for non-linear inverse 3D electrical tomography reconstruction'.* Physiol. Meas. Vol. 23, pp. 95-104.

Murai T, Kagawa T, (1985). *'Electrical impedance computed tomography based on a finite element model'.* IEEE Transactions on Biomedical Engineering. Vol. 32, No 3, pp. 177-185.

Naidoo T, (2001). *Image and signal processing in electrical resistance tomography.* Undergraduate Thesis, Department of Electrical Engineering, University of Cape Town.

Pepper DW, Heinrich JC, (1992). *'The finite element method: Basic concepts and applications'.* Series in Computational and Physical Processes in Mechanics and Thermal Sciences. U.S.A.: Tailor and Francis.

Pinheiro PAT, Loh WW, Dickin FJ, (1997). *'Smoothness-constrained inversion for two-dimensional electrical resistance tomography'.* Measurement Science and Technology. No. 8, pp. 293-302.

Pinheiro PAT, Loh WW, Waterfall RC, (1999). *Development of 3-D reconstruction algorithms for ERT'.* 1st World Congress on Industrial Process Tomography. Buxton, Greater Manchester, pp. 569-576.

Plaskowski A, Beck MS, Thorn R, Dyakowski T, (1995). *'Imaging industrial flows. Applications of electrical process tomography'.* Bristol: Institute of Physics Publishing.

Polydorides NP, (2000). *First year report on PhD degree.* Website: http://skoda.ee.umist.ac.uk/nick/first.ps

Polydorides NP, Lionheart WRB, McCann H, (2001). *'Considerations in electrical impedance tomography'.* 2nd World Congress on Industrial Process Tomography. Hanover, Germany, pp. 387-394.

Polydorides NP, Lionheart WRB, (2002a). *'A Matlab toolkit for three dimensional electrical impedance tomography: A contribution to the EIDORS project'.* Measurement Science Technology. Vol. 13, pp. 1871-187.

Polydorides NP, McCann H, (2002b). *'Electrode configurations for improved spatial resolution in electrical impedance tomography'.* Measurement Science Technology. Vol. 13, pp 1862-1870.

Randall EW, Wilkinson AJ, Cilliers JJ, Xie W, Neethling SJ, (2001). *'Current pulse technique for electrical resistance tomography measurements'.* 2nd World Congress on Industrial Process Tomography. Hanover, Germany, pp. 493-501.

Randall EW, (2002). Technical Staff at Chemical Engineering, University of Cape Town. Personal Communication. January-December 2002.

Rao SS, (1999). *The finite element method in engineering: 3rd Edition.* Boston: Butterworth-Heinemann.

Ruan W, Guardo R, Adler A, (1996). *'Experimental evaluation of two iterative reconstruction method for induced current electrical impedance tomography'.* IEEE Trans. Med. Imaging. Vol. 15, pp. 180-187, cited by Vauhkonen (1997).

Savolainen T, Kaipio JP, Vauhkonen M, Karjalainen PA, (1996). *'An EIT measurement system for experimental use'.* Rev. Sci. Instr. Vol. 67, No. 10, pp. 3605-3609.

Silvester PP, Ferrari RL, (1983). *'Finite elements for electrical engineers'*, Cambridge: Press Syndicate of the University of Cambridge.

Simske SJ, (1987). *'An adaptive current determination and a one-step reconstruction technique for a current tomography system'.* MSc. Thesis. Rensselaer Polytechnic Institute as cited by Le Hyaric et al (2000).

Tarvainen M, Vauhkonen M, Savolainen T, Kaipio JP, (2001). *'Boundary element method and internal electrodes in electrical impedance tomography'.* Int. Journal Numerical Methods Eng. Vol. 50, pp. 809-824.

Tattersfield GM, (1999). *'Electromagnetic field theory: A student's manual'.* University of Cape Town.

Thompson JF, Soni BK, Weatherhill NP, (1999). *'Handbook of grid generation'.* USA: CRC Press LLC.

Tomo_Imager_2D, (2002). Software developed by the author at the University of Cape Town as research towards his masters degree.

Tomo_Imager_3D, (2002). Software developed by the author at the University of Cape Town as research towards his masters degree.

Tomoview, (2000). Software developed by students at the University of Manchester Institute Science and Technology for electrical resistance tomography.

van Weereld JJA, Collie DAL, Player MA, (2001). *'A fast resistance measurement system for impedance tomography using a bipolar DC pulse method'.* Measurement Science and Technology. Vol. 12, No. 8, pp. 1002-1011.

Vauhkonen M, Vadasz D, Karjalainen PA, Kaipio JP, (1996). *'Subspace regularization method for electrical impedance tomography'.* 1st International Conference on Bioelectromagnetism. Tampere, Finland.

Vauhkonen M, (1997). *'Electrical impedance tomography and prior information'.* Doctoral Thesis, Department of Applied Physics, University of Kuopio.

Vauhkonen M, Karjalainen PA, Kaipio JP, (1998a). *'A Kalman filter approach to track fast impedance changes in electrical impedance tomography'.* IEEE Trans. Biomed. Eng. Vol. 45, No 4, pp. 486-493.

Vauhkonen M, Vadasz D, Kaipio JP, Somersalo E, Karjalainen PA, (1998b). *'Tikhonov regularization and prior information in electrical impedance tomography'.* IEEE Trans. Med. Imaging. Vol. 17, pp. 285-293.

Vauhkonen PJ, Vauhkonen M, Savolainen T, Kaipio JP, (1999a). *'Static three dimensional electrical impedance tomography',* Annals New York Acad Sciences 873:472-481.

Vauhkonen PJ, Vauhkonen M, Savolainen T, Kaipio JP, (1999b). *'Three-dimensional electrical impedance tomography based on the complete electrode model'.* IEEE Trans. Biomed. Eng. Vol. 46, pp. 1150-1160.

Vauhkonen PJ, Vauhkonen M, Kaipio JP, (2000). *'Errors due to the truncation of the computational domain in static three-dimensional electrical impedance tomogra*phy'. Physiological Measurement. Vol. 21, pp. 125-135.

Vauhkonen M, Lionheart W RB, Heikkinen L M, Vauhkonen P J, Kaipio J P, (2001). *'A MATLAB package for the EIDORS project to reconstruct two-dimensional EIT images'.* Physiological Measurement. Vol. 22, pp. 107-111.

Vilhumen T, Kaipio JP, Vauhkonen PJ, Savolainen T, Vauhkonen M, (2002). *'Simultaneous reconstruction of electrode contact impedances and internal electrical properties: I. Theory'.* Measurement Science Technology. Vol. 13, pp 1848-1854.

Wang M, (2001). *'Sensitivity theorem based inverse solution using conjugate gradients methods for electrical impedance tomography'.* 2nd World Congress on Industrial Process Tomography. Hanover, Germany, pp.395-405.

Webster JG, (1990). *'Electrical impedance tomography'.* Adam Hilger, cited by Vauhkonen (1997).

White RB, Simic K, Strode PR, (2001). *'The combined use of flow visualisation, electrical resistance tomography and computational fluid dynamics modelling to study mixing in a pipe'.* 2nd World Congress on Industrial Process Tomography. Hanover, Germany, pp.610-617.

Wilkinson AJ, (2001). Doctor of Electrical Engineering, University of Cape Town. Weekly Personal Interview. June-October 2001.

Wilkinson AJ, (2002). Doctor of Electrical Engineering, University of Cape Town. Personal Communication. January-December 2002.

Winkler A, (2001). *'Investigation of an electrical resistance tomography (ERT) system'.*

Ying W, (2000). *'Research of the image reconstruction algorithm for electrical resistance tomography based on neural network'.* 15th World Conference on Non-destructive Testing. Roma, Italy.

Yorkey TJ, (1986). *'Comparing reconstruction algorithms for electrical impedance tomography'.* Doctoral Thesis, Department of Electrical and Computational Engineering, University of Wisconsin, as cited by Belward (1997), Hua et al (1990), and Cilliers et al (2001).

Yorkey TJ, Webster JG, Tompkins WJ, (1987). *'Comparing reconstruction algorithms for electrical impedance tomography'.* IEEE Transactions on Biomedical Engineering. Vol. 34, No 11, pp. 843-853.

Yorkey TJ, (1990). *'Electrical impedance tomography with piecewise polynomial conductivities'.* Journal of Computational Physics. Vol. 91, pp. 344-360.

Zienkiewicz OC, Taylor RL, (2000). *'The finite element method. Volume 1: The Basis'.* Fifth Edition. Oxford: Butterworth-Heinemann.

APPENDICES

APPENDIX A: TABLES FOR BOUNDARY VOLTAGE MEASURING SEQUENCE

Table A1: 104 Boundary Voltage Measurement Sequence

Excitation Pairs		Measured Pairs (via input MUX)													Number of pairs
		1	2	3	4	5	6	7	8	9	10	11	12	13	
1	16	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	13
2	1	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	13
3	2	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16		12
4	3	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16			11
5	4	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16				10
6	5	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16					9
7	6	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16						8
8	7	9-10	10-11	11-12	12-13	13-14	14-15	15-16							7
9	8	10-11	11-12	12-13	13-14	14-15	15-16								6
10	9	11-12	12-13	13-14	14-15	15-16									5
11	10	12-13	13-14	14-15	15-16										4
12	11	13-14	14-15	15-16											3
13	12	14-15	15-16												2
14	13	15-16													1
															104

Source: Barber et al (1983).

Table A2: 208 Boundary Voltage Measurement Sequence

Excitation Pairs		Measured Pairs (via input MUX)													Number of pairs
		1	2	3	4	5	6	7	8	9	10	11	12	13	
1	16	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	13
2	1	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	13
3	2	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-1	13
4	3	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-1	1-2	13
5	4	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-1	1-2	2-3	13
6	5	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-1	1-2	2-3	3-4	13
7	6	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-1	1-2	2-3	3-4	4-5	13
8	7	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-1	1-2	2-3	3-4	4-5	5-6	13
9	8	10-11	11-12	12-13	13-14	14-15	15-16	16-1	1-2	2-3	3-4	4-5	5-6	6-7	13
10	9	11-12	12-13	13-14	14-15	15-16	16-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	13
11	10	12-13	13-14	14-15	15-16	16-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	13
12	11	13-14	14-15	15-16	16-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10	13
13	12	14-15	15-16	16-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	13
14	13	15-16	16-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12	13
15	14	16-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13
16	15	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	13
															208

Table A3: Real Voltage Measurements in Table A2 Format

Current Injection Pattern No. \ Voltage Measurement as per Current Injection Pattern No.	1	2	3	4	5	6	7	8	9	10	11	12	13
1	992	344	219	158	112	117	105	122	127	164	240	451	1096
2	846	403	247	172	153	134	137	130	149	182	267	417	932
3	1113	459	265	207	170	159	145	152	170	215	274	413	1052
4	942	374	243	178	155	129	127	137	158	177	227	390	866
5	920	411	252	191	150	139	137	149	156	180	261	415	1120
6	1021	436	274	189	163	151	153	150	160	204	266	467	960
7	957	406	235	181	156	148	136	139	159	185	269	383	939
8	957	379	245	186	161	140	134	144	156	202	243	415	1023
9	902	408	255	199	157	141	142	144	170	184	264	441	972
10	1043	445	279	195	160	154	145	159	159	201	276	412	982
11	988	413	239	174	155	136	140	132	153	186	236	393	913
12	985	395	237	190	156	149	133	144	163	183	255	415	1056
13	918	377	254	179	158	127	134	142	149	188	253	445	997
14	966	463	268	207	156	150	148	146	169	200	283	421	996

Please note that the above table is the data for a undisturbed homogenous system and that these voltages are plotted in Figure 6.1.

APPENDIX B: EXAMPLES OF SCALING TECHNIQUES

Note that the data used in the following examples were extracted from Table A3 in the previous appendix. The results of these scalings can be found in the Results chapter. Please note that the data is for a homogenous tank.

Table B1: Example of Group Scaling

	Column 1 Voltages		G.S. Factors = Average / Voltage C1		New Column 1 Voltages	New Column 2 Voltages	New Column 3 Voltages	...
					GS x Voltage C1	GS x Voltage C2	GS x Voltage C3	...
1	992		0.9758		968	336	214	
2	846		1.1442		968	461	282	
3	1113		0.8697		968	399	231	
4	942		1.0276		968	384	249	
5	920	Average = 968	1.0522		968	432	265	
6	1021		0.9481		968	414	259	
7	957		1.0115		968	411	238	...
8	957		1.0115		968	384	248	
9	902		1.0732		968	438	274	
10	1043		0.9281		968	413	259	
11	988		0.9798		968	404	234	
12	985		0.9827		968	388	233	
13	918		1.0545		968	398	268	
14	966		1.0021		968	464	268	

Table B2: Example of Individual Scaling

	Column 1 Voltages	Column 2 Voltages	Column 3 Voltages	...	Individual Scale Factors = Average Voltage Ci			...	New Column 1 Voltages	New Column 2 Voltages	New Column 3 Voltages	...
					C1	C2	C3	...	IS C1 x Voltage C1	IS C2 x Voltage C2	IS C3 x Voltage C3	...
1	992	344	219		0.9758	1.18648	1.14408		968	408	251	
2	846	403	247		1.1442	1.01326	1.01617		968	408	251	
3	1113	459	265		0.8697	0.88970	0.94624		968	408	251	
4	942	374	243		1.0276	1.09149	1.03414		968	408	251	
5	920	411	252		1.0522	0.99304	0.99451		968	408	251	
6	1021	436	274	...	0.9481	0.93514	0.91703		968	408	251	
7	957	406	235		1.0115	1.00516	1.06637		968	408	251	
8	957	379	245		1.0115	1.07551	1.02438	...	968	408	251	...
9	902	408	255		1.0732	1.00051	0.98222		968	408	251	
10	1043	445	279		0.9281	0.91716	0.89863		968	408	251	
11	988	413	239		0.9798	0.98882	1.05128		968	408	251	
12	985	395	237		0.9827	1.03234	1.05709		968	408	251	
13	918	377	254		1.0545	1.08223	0.98815		968	408	251	
14	966	463	268		1.0021	0.88087	0.93635		968	408	251	
Averages	968	408	251									

Table B3: Example of Twin Scaling

	Column 1 Voltages	Column 13 Voltages	Column 2 Voltages	Column 12 Voltages	...	Twin Scale Factors = Twin Average / Voltage Ci				...	New Column1 Voltages	New Column 13 Voltages	New Column 2 Voltages	New Column 12 Voltages	...
						C1	C13	C2	C12	...	TS C1 x Voltage C1	TS C1 x Voltage C1	TS C1 x Voltage C1	TS C1 x Voltage C1	...
1	992	1096	344	451		0.98836	0.89475	1.20357	0.91713		981	981	414	414	
2	846	932	403	417		1.15861	1.05241	1.02785	0.99391		981	981	414	414	
3	1113	1052	459	413		0.88096	0.93174	0.90251	1.00263		981	981	414	414	
4	942	866	374	390		1.04106	1.13250	1.10721	1.06171		981	981	414	414	
5	920	1120	411	415		1.06529	0.87554	1.00734	0.99849		981	981	414	414	
6	1021	960	436	467		0.95992	1.02187	0.94861	0.88603		981	981	414	414	
7	957	939	406	383		1.02498	1.04474	1.01964	1.08106		981	981	414	414	
8	957	1023	379	415	...	1.02458	0.95819	1.09100	0.99785	...	981	981	414	414	...
9	902	972	408	441		1.08764	1.00831	1.01492	0.93867		981	981	414	414	
10	1043	982	445	412		0.94028	0.99861	0.93037	1.00408		981	981	414	414	
11	988	913	413	393		0.99212	1.07408	1.00306	1.05313		981	981	414	414	
12	985	1056	395	415		0.99532	0.92845	1.04721	0.99742		981	981	414	414	
13	918	997	377	445		1.06795	0.98332	1.09781	0.92995		981	981	414	414	
14	966	996	463	421		1.01489	0.98409	0.89355	0.98331		981	981	414	414	
Average	**968**	**993**	**408**	**420**											
Twin Average	**981**		**414**												

APPENDIX C: DERIVATION OF THE ELEMENT MATRIX

In this appendix the derivation of the following element matrix will be provided. This chapter should be read in conjunction with Chapter 4.3 since many of the references are listed in that chapter.

$$y = \frac{\sigma}{4 \bullet Area} \begin{bmatrix} b_1^2 + c_1^2 & b_1b_2 + c_1c_2 & b_1b_3 + c_1c_3 \\ b_1b_2 + c_1c_2 & b_2^2 + c_2^2 & b_2b_3 + c_2c_3 \\ b_1b_3 + c_1c_3 & b_2b_3 + c_2c_3 & b_3^2 + c_3^2 \end{bmatrix}$$

The above equation will be derived from the analytical equation below.

$$y_{iij} = \sigma \bullet \iint_A \left(\frac{\partial H_i}{\partial x}\frac{\partial H_j}{\partial x} + \frac{\partial H_i}{\partial y}\frac{\partial H_j}{\partial y} \right) dA \quad \text{for I,j = 1,2,3.}$$

The element shape functions are presented below.

$$H_1 = \frac{(a_1 + b_1x + c_1y)}{2 \bullet Area} \qquad H_2 = \frac{(a_2 + b_2x + c_2y)}{2 \bullet Area} \qquad H_3 = \frac{(a_3 + b_3x + c_3y)}{2 \bullet Area}$$

The partial derivatives for the above element shape functions with respect to x are:

$$\frac{\partial H_1}{\partial x} = (b_1) \qquad \frac{\partial H_2}{\partial x} = (b_2) \qquad \frac{\partial H_3}{\partial x} = (b_3)$$

The partial derivatives with respect to y are:

$$\frac{\partial H_1}{\partial y} = (c_1) \qquad \frac{\partial H_2}{\partial y} = (c_2) \qquad \frac{\partial H_3}{\partial y} = (c_3)$$

Substitution of these partial derivatives into the analytical equation results in the element matrix presented at the top. Similarly the 3D element matrix can be derived and is thus not shown.