Parallel Algorithms for Three Dimensional Electrical Impedance Tomography

by
K. S. Paulson BSc MSc

School of Computing and Mathematical Sciences
Oxford Brookes University.

A Thesis submitted to the Oxford Brookes University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

November 1992.

Declaration

This dissertation has not been, nor is currently being submitted for the award of any other degree or similar qualification.

K. S. Paulson.

Glossary.

Symbol	Usual meaning
χ	characteristic functions of nodes in finite element mesh
χ_i	current driving characteristic function of electrode i
ξ_i	voltage measurement characteristic function of electrode i
$\Delta\sigma$	conductivity update
$\partial\Omega$	the boundary of the region to be imaged
E	experimental measurement
ϕ	electric potential
γ_i	the portion of the boundary covered by electrode i
HBW	half band width of a matrix
H^n	Sobolev space of smoothness n
I_i	current on electrode i
J	current density normal to the boundary of a region; or Jacobian matrix
K	system stiffness matrix
K_e	element stiffness matrix
LL^T	Choleski factorisation
M	measurement pattern
Mflops	million floating point operations per second
N	shape function
μ	Tikhonov regularisation factor
Ω	the region to be imaged
Ω_e	the region corresponding to one element
Ψ	lead field
ρ	resistivity
$R(\sigma)$	transfer impedance operator or matrix
σ	conductivity as a function of position
θ_i	polar angle at centre of electrode i
$U\Lambda V^T$	Singular value decomposition, SVD
V_i	voltage on electrode i
z	contact impedance

Abstract

Parallel Algorithms for Three Dimensional Electrical Impedance Tomography.

K. S. Paulson.

This thesis is concerned with Electrical Impedance Tomography (EIT), an imaging technique in which pictures of the electrical impedance within a volume are formed from current and voltage measurements made on the surface of the volume. The focus of the thesis is the mathematical and numerical aspects of reconstructing the impedance image from the measured data (the reconstruction problem).

The reconstruction problem is mathematically difficult and most reconstruction algorithms are computationally intensive. Many of the potential applications of EIT in medical diagnosis and industrial process control depend upon rapid reconstruction of images. The aim of this investigation is to find algorithms and numerical techniques that lead to fast reconstruction while respecting the real mathematical difficulties involved.

A general framework for Newton based reconstruction algorithms is developed which describes a large number of the reconstruction algorithms used by other investigators. Optimal experiments are defined in terms of current drive and voltage measurement patterns and it is shown that adaptive current reconstruction algorithms are a special case of their use. This leads to a new reconstruction algorithm using optimal experiments which is considerably faster than other methods of the Newton type.

A tomograph is tested to measure the magnitude of the major sources of error in the data used for image reconstruction. An investigation into the numerical stability of reconstruction algorithms identifies the resulting uncertainty in the impedance image. A new data collection strategy and a numerical forward model are developed which minimise the effects of, previously, major sources of error.

A reconstruction program is written for a range of Multiple Instruction Multiple Data, (MIMD), distributed memory, parallel computers. These machines promise high computational power for low cost and so look promising as components in medical tomographs. The performance of several reconstruction algorithms on these computers is analysed in detail.

Acknowledgements.

I would like to express my thanks to my Director of Studies, Mike Pidcock, and to my Supervisors Bill Lionheart and David Barber, for their help, criticism, encouragement and support over the period this work has taken to complete. I would also like to express my gratitude to Oxford Polytechnic, and in particular to the School of Computing and Mathematical Sciences for providing financial support and facilities. I am grateful also to all those involved with the European Concerted Action on EIT who have provided a constant reminder of the unknowns yet to be conquered. Special thanks go to the other members of the EIT group at Oxford Polytechnic who have been, and continue to be, a pleasure to work with; Ching Zhu, John Lidgey, Chris Mcleod and Tim Davey-Winter who joined us for a short time.

My colleagues at Oxford Polytechnic have been a source of encouragement. My fellow postgraduates, Avril Smith, Nick Wilson, Andy Smale, Paul Roach, and John Sandeman have often been a source of good-humour and diversion. The Post-Graduate Society has been a font of support and entertainment. Thanks are due also to my teaching colleagues, and students, who have made allowances for the conflicting constraints on me as I completed this project.

Special thanks deservedly belongs to my oldest friends and, soon to be, fellow doctors; Rene Onrust, Julian Ballance, Alistair Young and Don Grainger. Thanks also belong to my landlady, Philippa Lee, for putting up with me and to my close friends Tanya Nuttall and Nora Cranston. My parents and sisters deserve thanks for not complaining about how little they have seen of me over the past years. A great multitude of other people, too numerous to name, have contributed to my well-being during this period. All of them deserve my thanks and gratitude.

Contents

Chapter 5 Parallel Computing

Chapter 6 Parallel Algorithms for EIT

Chapter 1

Introduction

1.1 Electrical Impedance Tomography

Electrical Impedance Tomography (EIT) is a technique for imaging the interior of a region by the application and measurement of electric fields at the surface of the region. The calculated image reflects the spatial variation of the electrical impedance within the region. Such images have application in medical research, diagnosis and patient monitoring. Other industrial applications include non-destructive testing, process monitoring and geophysical exploration.

In EIT a volume Ω with boundary $\partial\Omega$ and unknown conductivity distribution σ is imaged by applying electric current to a finite number of electrodes fixed to the boundary. Within a source-free conductor the potential ϕ is governed by the conduction equation:

$$\nabla \cdot \sigma \nabla \phi = 0 \qquad \text{in } \Omega.$$

This is a second order partial differential equation in ϕ which, for arbitrary conductivity distributions, can only be solved numerically. For a unique solution to exist, a complete set of boundary conditions needs to be known. The determination of ϕ given the conductivity distribution, σ, and these boundary conditions is known as the forward problem. In EIT the inverse problem is solved; σ is calculated from boundary measurements of ϕ and $\partial\phi/\partial n$.

1.2 EIT in Medicine

Non-invasive techniques for gauging the form and function of the human body have long been popular with both patients and doctors. Listening to the sounds of the human chest has provided information about the heart and lungs to clinicians for hundreds of years. Technical developments lead to the production of images of the interior of the human body. X-rays pass through human tissue in straight lines until they are scattered or absorbed. Medical X-ray photographs are images of the proportion of X-rays that pass through the body without interacting with tissue. Each point on the image indicates the X-ray interaction characteristics along a straight line path through the body. More recently, techniques have been developed that

produce images of cross-sectional planes through the body. The first and most widely known example of these *Tomographic Imaging* techniques is X-Ray Computerized Axial Tomography, or CAT scan. Other tomographic imaging techniques are Ultrasound and Magnetic Resonance Imaging, MRI. Each of these techniques images a different property of the tissue under investigation and so has different application. MRI images the water contained in tissue while Ultrasound images its acoustic reflectance. Thus, different tissues may be resolvable by one imaging technique and not by others. Both CAT and MRI tomographs are physically large and require investments of the order of millions of pounds. All three imaging techniques are labour intensive to use.

EIT aims to image tissue impedance. This property is very difficult to measure *in vitro*, as changes in tissue impedance are strongly linked to changes in moisture content and temperature after death. The impedance changes at different frequencies can vary enormously as the cell membranes decay. Measurement of the impedance of living tissue *in vivo* poses equal problems. Despite these difficulties tables of approximate conductivities of human and animal tissue have been published, see [3]. These indicate conductivity contrasts between soft tissues of the order of 10:1. The maximum conductivity contrast between soft tissue and adult bone is of the order of 50:1. An impedance tomograph will be able to distinguish organs with conductivity contrasts of this magnitude given sufficiently accurate surface measurements.

An impedance tomograph will be compact, portable, cheap and harmless. Unlike CAT and MRI, EIT could be used for the continuous monitoring of patients over extended periods of time. Although EIT will never compete with these physiological imaging techniques in the resolution of images, it can compete with other functional imaging techniques such as Emission Computed Tomography, ECT. Functional techniques image the operation of organs rather than their physiology. ECT typically images the distribution of radioactive substances introduced into the body and concentrated by the action of organs. EIT has potential application in the measurement and monitoring of lung perfusion, lung ventilation, gastric emptying and cardiac output as each of these involve the circulation of highly conductive or non-conductive fluids through the body. Measurements of these functions are difficult with other techniques. An important application of EIT could be the measurement of lung water content. Physiological imaging techniques give no information about this important parameter. Other medical applications that have been proposed or tested are the measurement of pelvic congestion in women, gastric motility and bladder filling. A review of these and many other possible medical applications of EIT is given by Brown [12].

1.3 EIT in Process Tomography

EIT will certainly have many applications in the monitoring and control of industrial processes. Applications exist in the monitoring of conducting liquids in vats or their flow through pipes. Industrial applications lack the constraints imposed on medical EIT systems to ensure the complete safety and comfort of the patient and unlike human patients, vats may have electrodes fixed or built into them with their locations known precisely. Once the electrodes are in position they may be left there indefinitely. Vats may have electrodes placed inside their volume, something patients object to vociferously! This allows images with much greater resolution to be calculated. Greater currents can be applied to vats of chemicals and so larger, more accurate signals can be measured. Much of the fail-safe electronics required to protect a human patient from injury is totally unnecessary when imaging industrial processes. All these factors make EIT an attractive imaging technique for industry where it can achieve much better results than is possible working under the constraints imposed by human subjects. In the past twelve months many papers have appeared in the literature describing industrial applications ranging from the measurement of solid material transport in sewers [20] through velocity measurement in two-phase flow through pipes for the petro-chemical industry [40] to distributed pressure measurement [25].

1.4 Description of EIT

The general structure of an electrical impedance tomograph can be represented as in Figure 1.4a. Electric currents are applied to a region to be imaged through electrodes attached to its surface. The current passes between electrodes and induces a potential field throughout the region. The potential is measured on, possibly the same, surface electrodes. These electrode current and voltage measurements are the data used for image reconstruction. An iterative reconstruction algorithm includes a model capable of predicting the voltages that would be measured on a region of arbitrary conductivity distribution, during the application of currents to surface electrodes. Image reconstruction proceeds by comparing the voltages measured on the region to be imaged with those predicted by the model using the present *best estimate* of the conductivity distribution. Where no *a priori* information is available the best estimate could be a uniform conductivity distribution. For medical imaging the best estimate could be a uniform distribution with the mean conductivity of human tissue or a conductivity distribution consistent with a typical placement of organs. A correction to the model conductivity is calculated from the the difference in the experimentally measured potential and that predicted by the model. The corrected

model conductivity forms the image after a single iteration of the reconstruction algorithm. The process can be repeated, reducing the difference in the experimental and model voltage measurements at each stage. When no currents applied to the region to be imaged yield voltage measurements different from those predicted by the model, the method is said to have converged and the model conductivity is the resulting image.

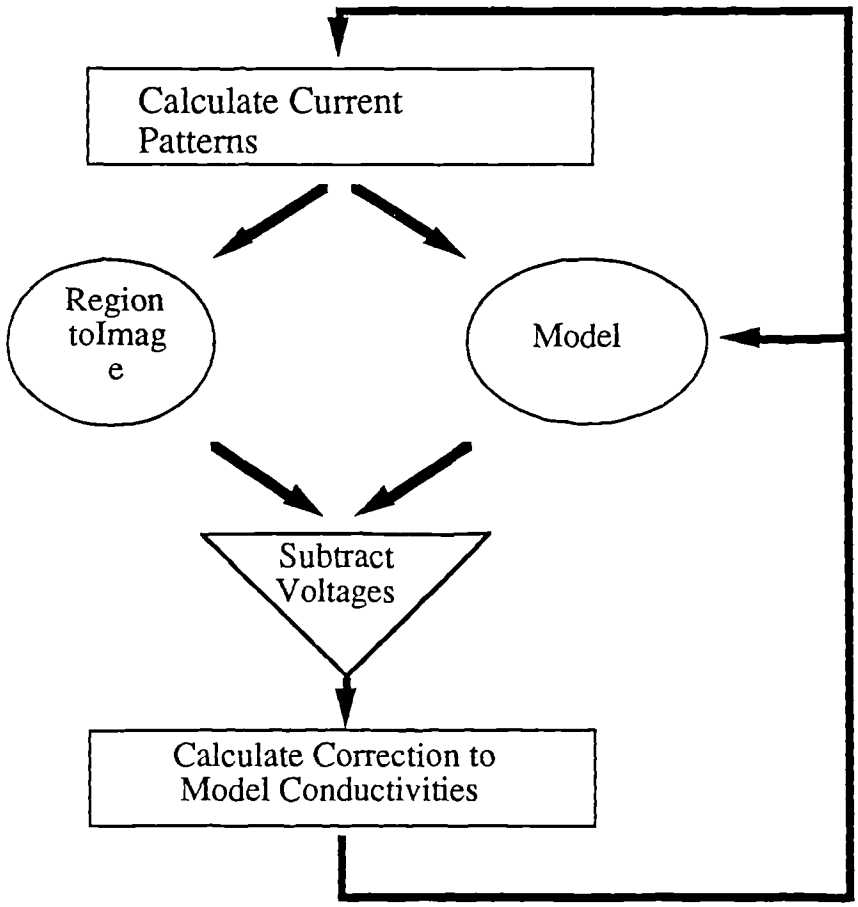


Figure 1.4a *The general structure of an electrical impedance tomograph.*

1.4.1 Dynamic Imaging

Two forms of impedance imaging are prevalent yielding *differential* or *absolute* images. Differential imaging produces images of conductivity differences between two regions or changes within one region between two times. This form of imaging was originally advocated by Barber and Brown [4] [11], working at the Royal Hallamshire Hospital in Sheffield, and has gained wide application. The

differences in the two sets of voltage measurements are the data used for reconstruction. The reconstruction algorithm developed by the Sheffield group, known as Back Projection, calculates a single correction to a uniform first estimate of the difference in the two conductivity distributions. These voltage differences are effectively "back projected" along equipotentials to form an image. Kim *et al*, [48], suggest an alternative algorithm which back projects along current stream lines. Both these algorithms use a linearised approximation for the conductivity to voltage measurement map is used and so reconstruction is a single matrix-vector operation. The method has four major strengths. Firstly, as only a single iteration is performed, the matrix that transforms the measured data into a conductivity update can be pre-calculated and stored. Secondly, many of the systematic errors in the data cancel to first order when the difference in the voltage measurements is calculated. In particular, the method appears to be relatively insensitive to errors in electrode placement. This is a great simplification in clinical situations where electrodes are attached to patients. Thirdly, image reconstruction using back projection is numerically trivial and so images can be calculated very rapidly on small and inexpensive computers. Fourthly, the equipment used to acquire the data for Back Projection applies current between adjacent pairs of electrodes. These are known as APT current patterns and are accomplished by multiplexing a single current source to adjacent pairs of electrodes. The simplicity of this arrangement leads to inexpensive and compact tomograph electronics.

The Sheffield system has been extremely successful with many research centres throughout the world testing its performance on a wide range of applications. Perambulatory systems have been built to monitor patients during typical daily activities, [51], and real-time systems have been constructed capable of acquiring the data for, and calculating 25 images per second [13]. Many of the draw-backs of the APT technique are those common to all forms of differential imaging. At best, difference images represent changes in conductivity from one moment to the next. The function of organs, such as the profusion of the lungs with blood, can, in theory, be imaged but disorders such as tumours, whose conductivity does not change during the imaging period, are invisible. Important quantities such as the water content of the lungs cannot be measured with differential imaging techniques. The back-projection algorithm also transforms changes of other variables into changes of conductivity. In particular, changes in boundary shape are interpreted as changes in conductivity. Lastly, back-projection along equipotentials is inherently a two dimensional reconstruction algorithm and no way is known to extend its application to three dimensions.

1.4.2 Static Imaging

Static, or absolute, imaging attempts to calculate the true impedance everywhere in the region to be imaged. This problem is considerably more difficult than differential imaging. The data used for reconstruction is the difference in the voltages measured on the physical region and those predicted by a numerical model. This data includes many of the systematic errors that differential imaging avoids. For multiple iteration reconstruction algorithms to converge to the correct conductivity distribution the forward model must correctly and accurately mimic the physical processes involved in passing electric current between electrodes attached to the region to be imaged. Where complicated conductivity distributions or geometries are being modelled, this part of the reconstruction is complex and numerically intense.

The research group at Rensselaer Polytechnic Institute have developed an imaging system with reconstructions based on a single iteration starting from a uniform first estimate of the conductivity distribution. This system has been shown to produce useful images on phantom and dog trials. As with Back Projection, the matrix system linking the electrical measurements and the conductivity update may be pre-calculated analytically and be used to calculate each image, as long as current patterns are used which lie in a space spanned by predetermined basis. Thus their reconstruction algorithm, known as NOSER - an acronym for Newton One Step Estimated Reconstructor [14], is potentially as fast as back-projection. Unlike Back Projection the algorithm is equally applicable in three dimensions as in two. The images indicate conductivity contrasts within the region to be imaged but do not yield correct conductivities. More iterations of the reconstruction algorithm would be necessary for the model conductivities to be an accurate indication of the region's conductivity distribution. However the images may prove to be medically useful for identifying tumours or embolisms and there may be correlations with NOSER images and other useful medical parameters.

1.4.3 Adaptive Imaging

One of the developments pioneered by the Rensselear group has been the use of *optimal currents*. Absolute imaging is more susceptible to distortion by noise than difference imaging and it is, therefore, more important for an absolute tomograph to collect high quality data. Isaacson [44], formulated a method for calculating the optimal currents to apply to the electrodes to yield the highest quality data. Optimal currents are calculated from voltage measurements made on the region to be imaged and the numerical model. The applied currents are adapted to maximise a quantity called *distinguishability* (see Section 2.2.7). This measures the size of the difference

in the voltages measured on the region and the model for current patterns of unit power. Unlike APT patterns, the use of optimal current patterns typically means applying current to all the electrodes attached to the region. A large number of matched current sources are necessary to apply optimal current patterns to a region, and so the driving electronics are considerably more complex than the APT system. The systems built by the Rensselaer group, known as ACT I, ACT II and ACT III, dedicate a matched current source to each electrode [64]. Some loss of data quality is incurred as voltage measurements now take place on electrodes that are applying current to the region. Any variation of the contact between electrodes and the region contaminates the voltages measured on current carrying electrodes. At the present time it is not known if the benefit gained by using optimal current patterns compensates for this degradation of the voltage measurements.

When both the physical and model conductivity distributions are rotationally symmetric disks, the optimal current patterns are trigonometric; $J(\theta)=\sin(k\theta)$ or $J(\theta)=\cos(k\theta)$ where J is the current density normal to the boundary at the point whose polar angle is θ and k is an integer. The first two optimal currents, yielding the largest distinguishability, are $J(\theta)=\sin(\theta)$ and $J(\theta)=\cos(\theta)$. Higher values of k yield decreasing distinguishabilities. The NOSER algorithm typically assumes that trigonometric current patterns have been applied.

1.4.4 Permittivity Imaging

At the frequencies generally used for medical impedance imaging, 10 to 100 kHz, the reactive part of the body is relatively small. Typical phase angles between the current and voltage measured on electrodes are approximately 2^{o}. Commonly, the real part of the voltage measurements will be used to reconstruct an image of the conductivity, σ, of the region. However, at lower frequencies or in industrial applications, the reactive component of the measured voltages may not be insignificant and both the real and imaginary part of the region's impedance distribution can be imaged. Although the small reactive component makes static imaging of tissue difficult, it is possible to produce differential images from difference measurements. Griffiths [36] has demonstrated the reconstruction of conductivity and permittivity images from synthetically generated data and later, [38], from real data collected from the thorax of human subjects. A Back Projection algorithm was used acting on voltage data measured at two different frequencies. Reconstructions of absolute permittivity images produced by a Newton type algorithms have been reported in industrial applications.

1.4.5 EIT at Oxford Polytechnic

The work of the group at Oxford Polytechnic has had much in common with the activities at Rensselaer. Like the Rensselaer group our aim has been to produce a tomograph capable of forming absolute images of conductivity from data acquired in clinical applications. From the beginning it was decided that the system should be capable of applying optimal current patterns. The reconstruction algorithms have used multiple iterations to produce images of true conductivities. By 1988 a tomograph, designed by Murphy [62] and known as OXPACT I, had been constructed and tested on several phantoms in the laboratory. Despite the fact that this system never produced a recognizable image, much was learnt from the experience and the group was encouraged to continue. OXPACT II, designed and built by Zhu [90], was completed in 1991. Much of this thesis is concerned with theoretical considerations affecting the specification of this system and reconstructions from data acquired with it.

1.5 Computing in EIT

For single step reconstruction algorithms such as back-projection and NOSER the computing requirements are minimal. Existing impedance tomographs are capable of imaging a few hundred conductivity parameters. To compute the image a pre-factored matrix system of this dimension needs to be solved. The Sheffield and Rensselaer systems require of the order $O(10^5)$ floating point operations to calculate an image. The most modest desk top computer can perform this task in less than one second. This task becomes more numerically intense as the resolution of the system increases. A system with N electrodes can theoretically calculate an image with $O(N^2)$ conductivity parameters. This requires $O(N^4)$ floating point operations. Thus the computation required increases dramatically with increasing numbers of electrodes. A three dimensional imaging system can easily have ten times as many electrodes as a two dimensional system and so the reconstruction could take 10000 times longer. If two dimensional reconstruction took one second, three dimensions would require almost three hours. The use of sophisticated computing equipment can reduce the reconstruction time using these algorithms to acceptable levels.

Multiple iteration reconstruction algorithms face a more daunting task. For each iteration a numerical forward model needs to be formulated and solved. The results of this operation are used to calculate the voltage differences and hence, the conductivity update. This system needs to be calculated and factorised at each iteration. The forward problem is usually solved using the Finite Element method.

This involves the construction and factorisation of a square matrix whose dimension is proportional to the number of electrodes squared. The factorisation is an $O(n^3)$ operation where n is the dimension of the matrix thus the solution of this system requires $O(N^6)$ floating point operations. The computer capable of performing two dimension, single step, reconstruction in one second would require several weeks to execute a single iteration of a multiple iteration reconstruction scheme. Extending this calculation to three dimensional reconstruction would require a much longer period of calculation again.

1.6 Parallel Computers.

The system envisaged by the Oxford group requires formidable computational power to calculate an image in a reasonable amount of time. The obvious way to achieve this is to employ a super computer. However the constraints placed upon a clinical system make this approach unacceptable. Among the advantages of the proposed system are its cheapness and portability. The incorporation of a super-computer into the system nullifies both these potential desirable features. For use in a hospital environment the system needs to be compact, robust and "medic-friendly".

A key to the continued increase in the speed of computing devices in the last few years has been utilisation of *parallelism*. This applies the principle of "divide and conquer" to the computational work that needs to be accomplished. The principle can be applied at all levels of computing. The fast microprocessor and digital signal processing chips that are now available achieve their high speeds, in part, by performing many independent operations concurrently where the hardware inside the chip allows. Typically, arithmetic operations can be executing on the ALU (Arithmetic Logic Unit) while data or instructions are fetched from memory. More highly granular parallelism is possible where a parallel computer is constructed from a network of processors, each capable of independent operation. If the computational work is split between the processors then the time required to perform the calculation decreases as the number of processors is increased. Some overhead is introduced by the requirement for the processors to pass data and intermediate results around the network. In general it is a non-trivial task to distribute a given computation between a set of inter-connected processors so as to minimise this overhead. The success with which this task is performed determines the computation rate achieved by the parallel computer for that calculation.

In the last five years components designed to be the processors in parallel computers have been available. These components are microprocessors with the added hardware necessary for inter-processor communication and in many cases these components have been fabricated on single chips. In the case of the Inmos Transputer these chips have been powerful computers in their own right. In addition they have the hardware necessary to communicate data between pairs of Transputers at very high rates. For a few thousand pounds, boards containing several Transputers, each as powerful as a typical workstation, could be purchased as extension boards to IBM PC's. A parallel computer, possibly hosted in a micro-computer, would appear to be a solution to the problems posed by the requirements of a medical electrical impedance tomograph.

1.7 Summary

The problems of difference imaging and single step, two dimensional, absolute imaging appear to be solved in the sense that the prototype tomographs exist and are presently being tested by other research groups. Absolute imaging in both two and three dimensions has not been achieved. Work is required both to make it possible and practical.

This thesis investigates algorithms for absolute imaging. In Chapter 2 reconstruction based on Newton's Method is investigated and algorithms designed to minimise the computation and execution time are explored. Chapters 3 and 4 investigate mathematical and numerical models for EIT. In Chapter 5 some of the issues involved in choosing parallel hardware and parallel software are discussed. Reconstruction algorithms are looked at in detail and parallel algorithms for each stage are explored in Chapter 6. In Chapter 7 the OXPACT II system is described. Several Newton type algorithms are detailed and their performance is analysed by comparing images reconstructed from both synthetic data and data measured on physical test objects.

Chapter 2

EIT Reconstruction

2.1 Introduction

Electrical impedance tomography poses the problem of reconstructing the conductivity distribution inside a region from electrical measurements made on the boundary. In practice this boundary data is obtained by applying currents to electrodes placed against the boundary and measuring the voltages induced on, possibly the same, electrodes. It is well known, [6] and [7], that the determination of the interior conductivity distribution from electrical measurements on the boundary is a highly non-linear and ill-posed inverse problem.

The calculation of conductivity images, known as reconstruction, can be achieved by explicitly non-linear techniques, such as the method of Nachman [63] or the more recent Layer Stripping algorithm of Cheney and Isaacson [15] and [77]. Alternatively, methods based on iterative improvement can be used. These methods repeatedly calculate corrections to an estimate of the conductivity distribution. At each step voltage measurements are made on the region to be imaged and the same experiment is simulated on a model based on the present best estimate of the conductivity distribution. A correction to the model conductivity distribution is calculated which minimizes a cost function based on the difference between the simulated and measured voltage measurements. This process is repeated until the differences are less than the experimental error. These methods are commonly known as iterative "linearized algorithms" since the conductivity correction is often calculated assuming linear variation of the voltage measurements with conductivity changes.

Linearised reconstruction algorithms may be broken down into four stages: determination of the experimental measurements to perform, performing the measurements on the region to be imaged, the simulation of the same measurements on a model, and the calculation of the correction to the conductivity distribution. In Chapter 2 the first and last of these steps are considered. Chapters 3 and 4 investigate the forward modelling component necessary in iterative reconstruction methods.

2.2 Which Measurements to Make?

For an EIT system capable of applying any current pattern and measuring the voltage anywhere on the boundary, an important question is, what are the best current patterns to apply and the best voltages to measure? Clearly, current patterns that produce the same voltage measurements on both the imaged region and the reconstruction model reveal little about the differences between their conductivities. Another consideration is that small voltage measurements are likely to have a large relative error due to both random and systematic error. Thermal fluctuations and digital quantization add a random background of noise to every voltage measurement. Similarly, modelling, electronic gain and electrode placement errors add systematic noise to the reconstruction data. Systematic errors scale with the size of measurements while random errors often do not.

2.2.1 The Forward Model

In EIT a volume Ω with boundary $\partial\Omega$ and unknown conductivity distribution σ is imaged by applying electric current to a finite number of electrodes fixed to the boundary. Within a source-free conductor the potential ϕ is governed by the conduction equation:

$$\nabla\cdot \sigma\nabla\phi = 0 \qquad \text{in } \Omega. \qquad (2.2a)$$

This is a second order partial differential equation in ϕ which, for arbitrary conductivity distributions, can only be solved numerically. For a unique solution to exist, a complete set of boundary conditions needs to be known. These may be Dirichlet conditions in the form of potentials on the boundary or Neumann conditions in the form of current densities on the boundary or a mixture of both. The potential needs to be subject to at least one constraint for a unique solution to exist. The boundary conditions associated with the injection of current through a finite number of electrodes are explored in Cheng *et al* [16] and Paulson *et al* [70]. In EIT the inverse problem is solved; the conductivity distribution σ is calculated from knowledge of the currents injected into the region and measurements of the boundary voltages.

To describe mathematically the operator which maps boundary currents to boundary voltages it is necessary to define norms for the spaces that these functions occupy. The Sobolev norm of the function f, $\|f\|_m$, where m is a positive integer is

defined as:

$$\|f\|_m^2 = \sum_{|\alpha|\le m} \int |D^\alpha f|^2 dx_1 \ldots dx_n$$

$$D^\alpha f = \left(\frac{\partial}{\partial x_1}\right)^{\alpha_1} \ldots \left(\frac{\partial}{\partial x_n}\right)^{\alpha_n} f \quad \text{of order } |\alpha| = \alpha_1 + \ldots + \alpha_n \le m$$

Folland [28], details the extension of the definition of the Sobolev norm, $\|f\|_m$, to all real m. The Sobolev norm not only measures the size of a function but also the size of its derivatives and so it describes the smoothness of a function. If m is the largest number such that $\|f\|_m$ is finite then f is said to lie in the spaces H^n $\forall n \le m$.

The voltage, V, induced on the boundary of a region with conductivity σ can be expressed in terms of the transfer impedance operator, $R(\sigma)$ acting on the applied current pattern, J, $V = R(\sigma)J$. To include all current patterns of finite power the transfer impedance operator acts on current patterns in the Sobolev space $H^{-1/2}$ and yields voltage patterns in the space $H^{+1/2}$, [32]. If the transfer impedance operator is restricted to $R(\sigma): H^0 \to H^0$, it is self adjoint and compact and thus, not invertible.

As there are no sources or sinks in the interior of the region, the net current crossing the boundary is zero. This is a constraint on the current patterns we can apply. Similarly, as the potential is only defined up to an additive constant we can eliminate the ambiguity by choosing the average potential on the boundary to be zero. This is a constraint on the voltage patterns we can measure. These constraints remove one dimension from the spaces of current and voltage patterns. Thus, for quantities defined on the boundary, H^s is understood to be the subspace of H^s orthogonal to the constant function 1. These constraint equations can be written:

$$\int_{\partial\Omega} \phi = \int_{\partial\Omega} \sigma \frac{\partial \phi}{\partial n} = 0.$$

2.2.2 Measurements Made on Electrodes

If the boundary current and voltage patterns are approximated in bases of functions, $\{\chi_i\}: \chi_i \in H^{-1/2}$ and $\{\xi_i\}: \xi_i \in H^{+1/2}$ respectively, then the transfer impedance operator can be represented by a matrix:

$$V=R(\sigma J \qquad \text{where} \qquad (R(\sigma))_{ij} = <\xi_i, R(\sigma)\chi_j>.$$

The domain and range of the transfer impedance matrix are the spaces spanned by the bases for the voltage and current patterns. If $\chi_i=\xi_i$ then the restricted transfer impedance operator is self adjoint.

If the bases are orthonormal, i.e.

$$<\chi_i,\chi_j> = <\xi_i,\xi_j>=\delta_{ij}$$

and $$<\chi_i,1> = <\xi_i,1>=0$$

then the dual pairing of two functions, F and G, approximated in the same basis may be written as a vector dot product, $<F,G>=F\cdot G$. If F and G are both H^0 functions then:

$$<F,G>=\int_{\partial\Omega} FG.$$

For a full description of Sobolev spaces and dual pairings the reader is directed to Folland, [28].

For real EIT systems, current is applied and voltages are measured via electrodes attached to the boundary of the region. If current is applied through n electrodes and voltages measured on m electrodes then an invertible transfer impedance matrix can be defined, $R(\sigma)\in C^{(m-1)(n-1)}$, using a current and voltage basis function associated with each electrode:

$$V = R\ \sigma\ I$$

$V\in C^{n-1}$ and $I\in C^{m-1}$ are vectors of the voltages and currents measured on the electrodes. The current and potential on the last current driving and voltage measuring electrodes are determined via the constraints:

$$\sum_{i=1}^{m} I_i = \sum_{i=1}^{n} V_i = 0$$

The electrode bases are not orthonormal and even when currents and voltages are measured on the same set of electrodes the transfer impedance matrix is not self adjoint.

2.2.3 Experimental Measurements

An experiment in EIT can be defined as a measurement of a component of the difference between the voltage pattern induced on the surface of the region to be imaged and that predicted by a numerical model. Each experimental measurement involves the application of a current pattern to the boundary of the region. A component of the resultant boundary voltage pattern is measured with respect to a particular basis of the space of measurements. An experimental measurement thus results in a single number.

$$E_{ij}(\sigma_m) = \langle M_i, (R(\sigma_m) - R(\sigma_e))\, J_j \rangle \qquad (2.2b)$$

where:

σ_m is the model conductivity distribution,
σ_e is the test volume conductivity distribution,
$R(\sigma)$ is the transfer impedance operator,
M_i is a measurement pattern,
J_j is an applied current pattern,
$\langle \bullet, \bullet \rangle$ is the appropriate dual pairing,
Ω is the volume to be imaged and $\partial\Omega$ is its boundary,

The subscripts i and j on the measurement and current patterns range over the patterns used. The number of independent patterns will be determined by the number and position of electrodes used to apply current to the region and to make voltage measurements on the region, see Section 3.7.3.

2.2.4 Current Patterns and Measurement Patterns

Three different forms of current pattern, J_j, are in common use. The Back-Projection reconstruction algorithm of Barber and Brown, [3] and [4], assumes current patterns approximating current di-poles on the boundary. Their APT current patterns achieve this by driving current through adjacent electrodes attached to the surface. Isaacson, [44], derived an algorithm for calculating optimal current patterns which maximise the norm of the difference between the voltages measured on the region to be imaged and those predicted by the numerical model. These current patterns vary smoothly around the surface of the region and are approximated by driving current through all the electrodes simultaneously. Some researchers, including those at Rensselaer Polytechnic Institute and at Oxford Polytechnic, have

used trigonometric current patterns, $I_i^{trig}=\cos(k\theta_i)$ k=1,2,3..., where I_i^{trig} is the current delivered to the i'th electrode and θ_i is its angular position.

All other researchers known to the author use measurement patterns corresponding to measuring the voltage between pairs of electrodes. Barber and Brown measure the voltage difference between adjacent electrodes. In this thesis we consider trigonometric measurement patterns $\mathbf{m^k}$:

$$\mathbf{m^k}=\sum_{i=1}^{N-1} \cos(k\theta_i)\mathbf{e_i} \quad k=1,2,3..$$

and optimal measurement patterns.

In practice, trigonometric measurements are calculated from a linear combination of voltage measurements made between pairs of electrodes. Thus, the trigonometric measurements have a larger noise component than the physical measurements made between pairs of electrodes.

2.2.5 Optimal Current and Measurement Patterns

The experimental measurements which provide the most reliable information on which to base a reconstruction step are those for which the experimental measurement, E_{ij} as defined in Equation 2.2b, is largest. Similarly, for measurements with a background error of fixed amplitude the measurements with the highest relative precision are those for which E_{ij} is largest.

An understanding of the relationship between current and measurement patterns and the resulting experimental measurements can be gained by considering the singular value decomposition, SVD, of the difference in the transfer impedance operators. The SVD of operators is described in detail in Groetsch, [39], and the SVD of matrices is described in Golub and Van Loan, [33]. There exist functions, U_i and V_i, and positive real numbers λ_i such that:

$$(R(\sigma_m) - R(\sigma_e))\, U_i = \lambda_i V_i$$
$$(R(\sigma_m) - R(\sigma_e))^*\, V_i = \lambda_i U_i$$
$$\lambda_i \geq \lambda_j \geq 0 \quad \forall\ i < j$$
$$\langle U_i, U_j \rangle = \langle V_i, V_j \rangle = \delta_{ij}.$$

where R* is the adjoint of R. The U_i's and the V_i's are called the right and left

singular functions of $(R(\sigma_m) - R(\sigma_e))$ and form orthonormal bases for the spaces of current patterns and voltage patterns respectively. The λ_i's are singular values of the difference in the current to voltage maps of the model and imaged region. Equation (2.2b) can be rewritten in terms of this singular decomposition:

since
$$J_j = \Sigma_k <U_k,J_j> U_k$$
$$E_{ij} = <M_i, \Sigma_k V_k \lambda_k <U_k,J_j> >$$
$$= \Sigma_k <M_i,V_k> \lambda_k <U_k,J_j> \qquad (2.2c)$$

If $M_i = V_i$ and $J_j = U_j$ then the measurement made using the i'th singular measurement pattern and the j'th singular current pattern is:

$$E_{ij} = \Sigma_k <V_i,V_k> \lambda_k <U_k,U_j>$$
$$= \Sigma_k \delta_{ik}\lambda_k\delta_{kj} = \lambda_i \delta_{ij} \qquad (2.2d)$$

From Equation 2.2c it is clear that the supremum value of E_{ij} over all orthogonal bases of current and measurement patterns is λ_1 and that this is attained when $M_i = V_1$ and $J_j = U_1$. Once this choice has been made, the next highest value of E_{ij} :i,j≠1 occurs when $M_i = V_2$ and $J_j = U_2$. Again we can set i=j and continue this process. All measurements E_{ij} with i≠j are zero. Further, if the restricted, self adjoint operator $(R(\sigma_m) - R(\sigma_e)):H^0 \rightarrow H^0$ is used, the optimal current and measurement patterns are the same, i.e. $V_k=U_k$ and belong to the space C^∞, [32].

Using the discrete quantities, Equation 2.2b may be more concisely written:

$$E_{ij} = \mathbf{M_i}^T V \Lambda U^T \mathbf{J_j}$$

where U and V are basis matrices whose columns are orthonormal vectors and $\Lambda=\text{Diag}(\lambda_1,\lambda_2,\lambda_3,...,\lambda_{N-1})$, although the singular values will not be the same as the continuous ones. $\mathbf{J_j}$ is the vector of electrode currents and $\mathbf{M_i}$ is a vector of the weights used when forming the weighted sum of the electrode voltages.

Thus, there exists orthonormal bases of current pattern vectors and measurement pattern vectors which diagonalise the transfer impedance matrix. These vectors can be used to form optimal experimental measurements, E_{ii}, which are the largest possible given constraints on the size of the applied currents. All experimental measurements, E_{ij}, with i≠j are zero.

2.2.6 An Example of Optimal Current and Measurement Patterns

To demonstrate these optimal current and measurement patterns we can consider two unit disks with uniform conductivities σ_m and σ_e; $\sigma_m < \sigma_e$. The application of a boundary current density of $J^k = \cos k\theta$ to a unit, uniform disk produces a boundary voltage pattern of $\phi^k = 1/(\sigma k) \cos k\theta$. The left and right singular functions of $(R(\sigma_m) - R(\sigma_e))$, in this case, are $U_k=V_k=(1/\pi) \cos k\theta$ or $U_k=V_k=(1/\pi) \sin k\theta$ and the singular values are $\lambda_k = 1/(\sigma_m k) - 1/(\sigma_e k)$, k=1,2,3,.... In this case the value of the optimal experimental measurements are the difference in the power applied to the regions by the optimal current pattern.

$$E_{kk}=<V_k,(R(\sigma_e) - R(\sigma_m))\ U_k>=<V_k,\lambda_k V_k>=<U_k,\lambda_k V_k>= \int_{\partial\Omega} J^k(\phi_e{}^k-\phi_m{}^k)$$

where $\phi_e{}^k$ and $\phi_m{}^k$ are the potentials induced in the region to be imaged and the modelled region respectively.

2.2.7 Distinguishability

The current and measurement patterns above are optimal in the sense that they yield the largest voltage measurements. Other workers in the field have introduced current patterns that optimize measurements given other constraints. Isaacson *et al*, [44], optimize a measure called *distinguishability*. Two conductivity distributions, σ_m and σ_e, are said to be distinguishable by measurements of precision ε if there exists a current density pattern J of unit norm which produces a voltage pattern with norm greater than ε.

$$\delta(J) = \frac{\| (R(\sigma_e) - R(\sigma_m))J \|}{\| J \|} > \varepsilon$$

The number δ is called the distinguishability. The "best" current patterns to apply, in terms of maximising the distinguishability, are the eigenfunctions of self adjoint operator $|R(\sigma_e)-R(\sigma_m)|:H^0 \rightarrow H^0$. If the SVD is performed on the restricted operator then:

$$| (R(\sigma_e) - R(\sigma_m)) | J = \Sigma_k\ U_k\lambda_k<U_k,J>$$

It is clear that the U_i's are eigenfunctions of this operator and so Isaacson's optimal current patterns are the same as those calculated in Section 2.2.5.

The formulation proposed by Isaacson *et al* includes a range of definitions of "best" measurements in the norms used in the definition of distinguishability. Usually the L_2 norm is used for the current, $\|J\|$, as this corresponds to the power applied to the region. The L_2 norm of the difference in the voltage patterns induced by the application of the corresponding optimal current is the same as the optimal measurement derived earlier.

$$\|(R(\sigma_m) - R(\sigma_e))\, U_i\| = \langle \lambda_i V_i, \lambda_i V_i \rangle^{1/2} = \langle M_i, \lambda_i V_i \rangle = \lambda_i$$

Using the L_∞ norm for the numerator in the definition of distinguishability yields optimal currents which give the largest electrode voltage measurements. In the same way other definitions of <F,G> will result in other, possibly interesting, definitions of optimal patterns.

In a later paper, [31], Gisser *et al* refine the definition of distinguishability to cover all current patterns of finite power:

$$\delta(J) = \frac{\|(R(\sigma_m) - R(\sigma_e))J\|_{+1/2}}{\|J\|_{-1/2}}$$

where $\|A\|_m$ is the Sobolev norm of A. As the unrestricted operator is not self adjoint, the optimal currents are defined as the eigenfunctions of the self adjoint operator $(R(\sigma_m)-R(\sigma_e))^*(R(\sigma_m)-R(\sigma_e))$. These optimal currents are the same as the right singular functions of the unrestricted operator.

2.2.8 Conclusions

In this section, the transfer impedance operator, which maps boundary currents to boundary voltages, has been defined. A discrete transfer impedance matrix has been introduced based on the use of electrode characteristic functions to model the behaviour of electrodes attached to the boundary. Optimal current and measurement patterns have been defined which maximise the size of experimental measurements. These optimal experimental patterns use the currents which maximise the distinguishability and the measurement patterns which translate the resulting boundary voltages into the largest experimental measurements.

The advantage of the formulation based on singular value decomposition, as presented in this thesis, is that it makes explicit the properties of different measurement schemes. A number of research groups apply optimal currents yet all

groups known to the author use voltage measurements made between pairs of electrodes. Physically, voltage measurements are always made between pairs of electrodes, yet it must be remembered that optimal currents were chosen to maximise the distinguishability, not these pair voltage measurements. Breckon, [10], describes the calculation of pair optimal currents which maximise voltage differences for measurements made between pairs of electrodes. These current patterns are more appropriate if the pair voltage measurements are going to be used for reconstruction. Combining pair voltage measurements to simulate the use of optimal measurement patterns yields larger, albeit more noisy, experimental measurements but compresses the data into a single measurement for each current pattern. This compression is used by a novel reconstruction algorithm, known as POMPUS, described in Section 7.5.2.

2.3 Reconstruction based on Newton's Method

At each stage of an iterative reconstruction algorithm a correction to the present best approximation to the conductivity field needs to be calculated. The data used to calculate this correction are the differences between experimental measurements that have been made on the region to be imaged and the results of simulating these measurements on a computer model. Typically, Newton based algorithms are used to calculate a correction consistent with these data.

2.3.1 Newton Iteration

If $C(x)$ is a function of one variable and we wish to find x^* such that x^* is a turning point of the function C, i.e. $C'(x^*)=0$, then the standard Newton iteration scheme applied to the derivative of C may be used to achieve this. It may be stated as:

REPEAT

- Find Δx: $C''(x)\,\Delta x = -C'(x)$
- $x \leftarrow x+\Delta x$

If the initial value of x is within a sufficiently small neighbourhood of x^* and $C \in C^4$, then this algorithm is well known to converge quadratically to the turning point of C, $x \rightarrow x^*$. Where $C(\mathbf{x})$ is a function of several variables a multidimensional form of the Newton method can be used:

REPEAT

- Find $\Delta\mathbf{x}$: $H_C\Delta\mathbf{x} = -\nabla_x C$
- $\mathbf{x} \leftarrow \mathbf{x}+\Delta\mathbf{x}$

where H_C is the Hessian matrix of C. This process will converge to either a local or a global minimum of C depending upon the initial value of **x**. Variations of this procedure which perform line minimizations of the cost function along search directions generated by the Newton method will minimise a quadratic function with positive definite Hessian matrix in a finite number of steps, see Lootsma *et al*, [55]. If $C \in C^4$ is a function which is bounded below, then the multi-variable Newton method given above will converge quadratically to the global minimum of the function C as long as the initial value of **x** lies within a sufficiently small neighbourhood of it.

2.3.2 The Error Function

For EIT, an error function may be defined by $C(\Delta\sigma_m) = \sum_{ij} (E_{ij}(\sigma_m+\Delta\sigma_m))^2$, where $\Delta\sigma_m$ is the conductivity update and ij indexes the experimental measurements used for reconstruction. This error function always takes non-negative values. If the error function is equal to zero, C=0, then the model conductivity field and that of the region to be imaged cannot be distinguished by the experimental measurements we perform. If the current and measurement patterns are bases of their respective spaces then the error function measures the Frobenius norm of the difference in the transfer impedance operators. The use of optimal current and measurement patterns allows a simplification of the error function as:

$$C(0)=\sum_i \sum_j E_{ij}^2=\|R(\sigma_m)-R(\sigma_e)\|_F^2=\sum_i \lambda_i^2=\sum_i E_{ii}^2$$

We wish to find the conductivity update that is the global minimum of this function. The functions $E_{ij}(\sigma_m+\Delta\sigma_m)$ are highly non-linear and so typically, their behaviour can only be approximated locally. It is usual to work with an approximation to the function $C(\Delta\sigma_m)$, usually linear, and to repeatedly calculate corrections to the model conductivity distribution until C(0) is sufficiently small.

2.3.3 Newton's Method and EIT

From the multi-dimensional Newton's method and the definition of the error function the following relationships can be derived:

$$(H_C)_{KL} = 2\sum_{ij=1}^{M}\left(\frac{\partial E_{ij}}{\partial s_K}\frac{\partial E_{ij}}{\partial s_L} + E_{ij}\frac{\partial^2 E_{ij}}{\partial s_K \partial s_L}\right) \approx 2\sum_{ij}^{M}\left(\frac{\partial E_{ij}}{\partial s_K}\frac{\partial E_{ij}}{\partial s_L}\right)$$

$$\nabla C_K = 2\sum_{ij=1}^{M}\left(E_{ij}\frac{\partial E_{ij}}{\partial s_K}\right)$$

where K and L index the rows and columns of the Hessian matrix and the gradient vector. It is assumed that M experimental measurements are used and that these are indexed by ij. Typically the conductivity and the conductivity update are expressed in a finite basis of continuous functions $\{B_i\}$, $\sigma=\Sigma s_i B_i$. The derivatives are thus with respect to this parameterisation of the conductivity field. Methods of calculating the derivatives are described in detail in Section 2.5. The second order terms in the calculation of the Hessian matrix are usually omitted as they are both relatively small and expensive to calculate, see Breckon [9] for an explicit formula for the Hessian matrix. Alternatively, the second term is approximated by $\mu^2 I$ where μ is the Tikhonov regularisation parameter. Dennis *et al*, [21], approximate the second term by a symmetric matrix based on the first derivatives. A correction to the approximate Hessian is calculated before each Newton step. If these terms are neglected, the Newton method reduces to the task of calculating the conductivity update to the construction and solution of a set of linear equations.

2.3.4 The Least-Squares Method and EIT

The Least Squares method is commonly used when minimising a function contaminated by noise. In this subsection the Least Squares method is applied to the calculation of the conductivity update.

For each experimental measurement, E_{ij}, we can find a conductivity update such that $E_{ij}(\sigma_m+\Delta\sigma_m) = 0$. If the Taylor expansion for $E_{ij}(\sigma_m+\Delta\sigma_m)$ is truncated after the linear term this can be expressed as:

$$E_{ij}(\sigma_m+\Delta\sigma_m) = E_{ij}(\sigma_m) + \left(\frac{\partial E_{ij}}{\partial \sigma}\right)\Delta\sigma_m + \ldots = 0$$

$$\Rightarrow \left(\frac{\partial E_{ij}}{\partial \sigma}\right)\Delta\sigma_m = -E_{ij}$$

Given that the conductivity field is represented in a basis of continuous functions, $\{B_i\}$, $\left(\frac{\partial E_{ij}}{\partial \sigma}\right)^T$ is used to represent the gradient of the function $E_{ij}(\sigma)$ with respect to the coefficients s_k, $\nabla_s E_{ij}(\sigma)$. The conductivity update, $\Delta\sigma_m$, is also a vector in the basis $\{B_i\}$. Collecting these expressions for each experimental measurement yields:

$$J\,\Delta\sigma_m = E \qquad (2.3a)$$

where $J = \left(\frac{\partial E_1}{\partial \sigma}, \frac{\partial E_2}{\partial \sigma}, \frac{\partial E_3}{\partial \sigma}, \ldots, \frac{\partial E_M}{\partial \sigma}\right)^T$ and $E = -(E_1, E_2, E_3, \ldots, E_M)^T$ for some numbering of the experimental measurements, E_{ij}. The Jacobian matrix $J \in R^{M \times N}$ where N is the number of conductivity parameters.

Typically the system defined in Equation 2.3a may be either underdetermined or overdetermined depending on the number of measurements used for the reconstruction. It may be inconsistent due to the linearization assumption or errors in the experimental measurements. For these reasons a solution in the least squares sense will be calculated using the Moore-Penrose inverse of J, $J^\dagger$. $J^\dagger$ is defined to be the unique matrix which satisfies the four Moore-Penrose conditions:

$$JJ^\dagger J = J \qquad J^\dagger JJ^\dagger = J^\dagger$$
$$(JJ^\dagger)^T = JJ^\dagger \qquad (J^\dagger J)^T = J^\dagger J$$

These conditions amount to the requirement that $JJ^\dagger$ and $J^\dagger J$ be orthogonal projections onto the column spaces of J and $J^\dagger$ respectively, [33]. Thus the conductivity update may be calculated using:

$$\Delta\sigma_m = J^\dagger E = (J^T J)^{-1} J^T E \qquad M \geq N$$
$$\Delta\sigma_m = J^\dagger E = J^T (JJ^T)^{-1} E \qquad M \leq N \qquad (2.3b)$$

The model conductivity field can be adjusted by the addition of the correction calculated using the above equation; $\sigma_m \leftarrow \sigma_m + \Delta\sigma_m$. After this, the process may be repeated by calculating a new correction based on a new set of measurements on the region to be imaged and the model. This process can be repeated until physical and simulated measurements agree to measurement accuracy, known as Morozov's stopping criterion [61].

The first of Equations 2.3b is equivalent to solving the Newton step $H_C \Delta\sigma_m = -\nabla C$ derived in the previous section. The Hessian matrix and the gradient vector are related to the Jacobian matrix as follows:

$$H_C \approx J^T J \qquad -\nabla C = J^T E$$

Thus, Newton's method yields the conductivity update which minimizes, in the least

squares sense, the differences in the voltage measurements made on the region to be imaged and the numerical model.

2.3.5 Regularisation

Breckon, [9], has shown that the methods described in this section are adequate for the reconstruction of overdetermined data which is error and noise free. Experimental experience with synthetically generated data has shown them to converge to a region close to the global minimum of the error function when the conductivity fields are homogeneous with homogeneous anomalies and the first estimate conductivity distribution has been sufficiently close, see Section 7.5.2. Small amounts of noise added to the data reduce the resolution of the image by introducing large anomalies, particularly near the centre of the imaged region, see section 7.5.3. Larger errors can prevent convergence by introducing negative conductivities in the model conductivity distribution.

For real data measured on physical objects the largest accumulation of errors is in the experimental measurement vector E. These errors are a combination of random errors, such as thermal and digitization noise in the electronics used to apply the current patterns and make the voltage measurements, and more serious systematic errors such as electrode misplacement and matching errors in the driving electronics. The ill-posedness of the inverse problem means that small errors in the voltage differences are translated into large errors in the conductivity correction. A small relative error in E, certainly less than 1%, will result in the conductivity corrections generated by Equation 2.3b converging to an unrecognizable image or oscillating wildly.

The problem may be circumvented to some degree by regularising Equation 2.3b. This process replaces the ill-posed problem of 2.3b with a more stable, well-posed one. As a linear step in an iterative method, Levenberg [49] and Marquardt [57] suggest the Tikhonov regularised linear approximation:

$$\Delta\sigma_m = (J^T J + \mu I)^{-1} J^T E \qquad M \geq N \qquad (2.3c)$$

or

$$\Delta\sigma_m = J^T (J J^T + \mu I)^{-1} E \qquad M \leq N \qquad (2.3d)$$

where μ is the Tikhonov factor. This is equivalent to finding the $\Delta\sigma_m$ which minimizes $\|J\Delta\sigma_m - E\|^2 + \mu^2 \|\Delta\sigma_m\|^2$. Thus Tikhonov regularisation limits the magnitude of the conductivity correction by penalizing large corrections in the error

function This is equivalent to the constrained optimisation problem: minimise $\|J\Delta\sigma_m - E\|^2$ subject to $\| \Delta\sigma_m \| < \rho$. Here μ is the Lagrange multiplier when the constraint is active. The ball of radius ρ centred at the current approximation σ_m is called the "trust region" as it represents a region in which we can trust the linear approximation to $R(\sigma_m)$. For $\mu=0$ the method becomes simply the Newton-Kantorovich method but for large μ the direction of the update vector tends to the direction determined by the Steepest Descent method, J^TE. In this way the Levenburg-Marquardt update can be thought of as an interpolation between the slow but sure Steepest Descent method, and the rapid but unreliable Newton-Kantorovich method.

2.4 Error Analysis using Singular Value Decomposition

Each iteration of the reconstruction algorithm requires the solution of the system of Equations 2.3b. An understanding of the effect of varying the regularisation parameter μ can be gained by studying the singular value decomposition, SVD, of the Jacobian matrix, $J \in \mathbf{R}^{M\times N}$. The SVD factorizes the matrix J into a product of orthonormal basis matrices, $U \in \mathbf{R}^{M\times M}$ and $V \in \mathbf{R}^{N\times N}$ and a diagonal matrix of monotonically decreasing positive singular values, λ_i, $\Lambda = \mathrm{Diag}(\lambda_1,\lambda_2,\lambda_3,\ldots\lambda_{\min(M,N)})$ by writing $J = U\Lambda V^T$. The columns of U and V are the right and left singular vectors of the matrix J and lie in the space of voltage differences and conductivity distributions respectively.

$$J\,U_i = \lambda_i\,V_i \qquad\qquad J^*\,V_i = \lambda_i\,U_i$$
$$\lambda_i \geq \lambda_j \geq 0 \;\forall\; i<j \qquad\qquad \langle U_i,U_j\rangle = \langle V_i,V_j\rangle = \delta_{ij}$$

The solution of the regularised Least Squares system, Equation 2.3c, may be written:

$$\Delta\sigma_m = \sum_i \left(\frac{\lambda_i}{\lambda_i^2+\mu^2}\right)(U_i,E)V_i$$

where (X,Y) is the usual dot product, $(X,Y) = \Sigma\, X_iY_i$. It is clear from this formulation that the conductivity update is a weighted sum of the right singular vectors where the weighting is a product of an amplification factor, A_i, and a projection factor, P_i:

$$A_i = \frac{\lambda_i}{\lambda_i^2+\mu^2} \qquad P_i = (U_i,E)$$

While the projection factor is clearly bounded, $(U_i,E)^2 \leq (E,E)$, the amplification factor grows without limit as both λ_i and μ approach zero.

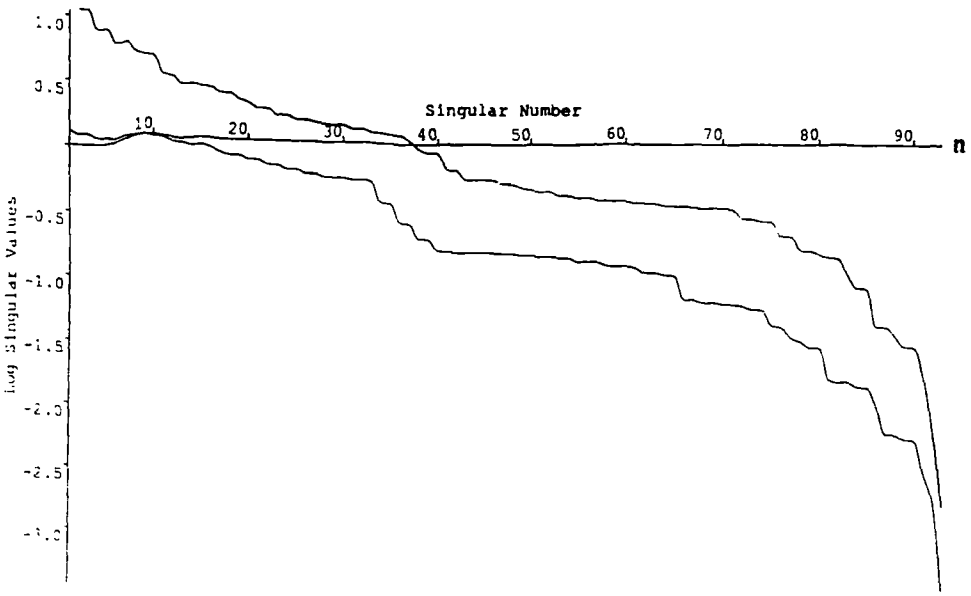


Figure 2.4a *The singular values of the derivative matrix for a uniform disk driven by trigonometric current patterns. The upper curve was calculated using trigonometric measurement patterns and the lower curve from adjacent pair measurement patterns.*

Figure 2.4a shows the singular values of a derivative matrix corresponding to a disk with a uniform conductivity distribution driven by trigonometric current patterns applied to thirty two, equally spaced, electrodes covering 30% of the boundary. Two sets of voltage measurement patterns were tested; voltages measured between adjacent electrodes and trigonometrically weighted voltage measurements. Voltages were measured on point electrodes half-way between the current injection electrodes. Trigonometric measurement patterns result in larger voltage

measurements and so the singular values of the derivative matrix using trigonometric weights are larger. The singular values decay with a similar pattern. Breckon, [9], reported a correlation between steps in the decay curve of the singular values and breaks in the symmetry of the singular functions. Singular values were found to decay faster than e^{-n} where n is the singular number. This is consistent with the results of Breckon, [9] for sixteen electrode systems, who found that the decay of the singular values was of the order $O(e^{-p(n)})$ where p is a polynomial of degree two or more. This very rapid decay in the singular values is indicative of the extreme ill-posedness of the EIT inverse problem.

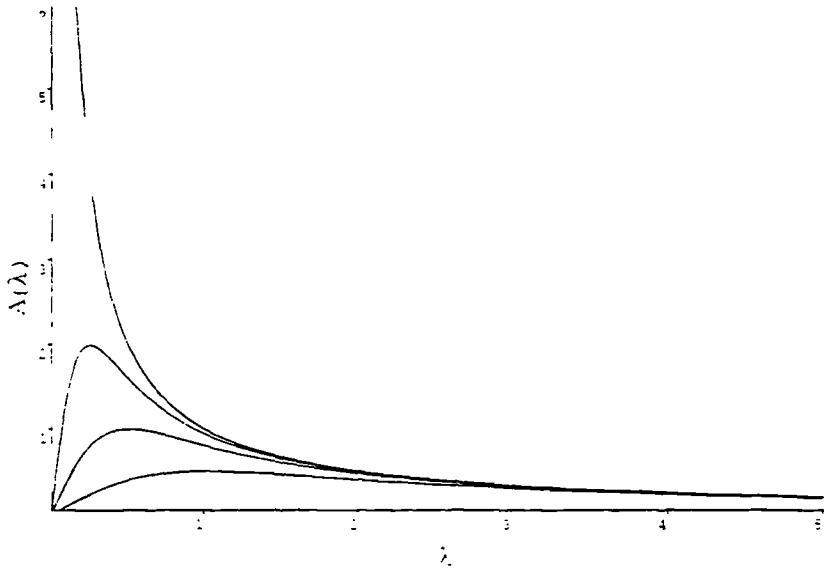


Figure 2.4b *The amplification factor* $A(\lambda) = \frac{\lambda}{\lambda^2+\mu^2}$ *for, from top to bottom;* $\mu=0$, $\mu=0.25$, $\mu=0.5$ *and* $\mu=1.0$.

If the model conductivity distribution is changed by the addition of V_i then the voltage difference measurement would change by $\lambda_i U_i$. Thus the singular conductivity distributions with small singular values are those that, when varied, make the least change in the voltage difference measurements. These are the components of the conductivity image that are least determined by boundary voltage measurements. As the amplification factor, without regularisation, is $1/\lambda_i$ these components of the conductivity field are associated with the largest amplification factor. Adding U_i to the data used for reconstruction changes the conductivity update

by $(1/\lambda_i)V_i$. Typically the voltage difference vector will be degraded by systematic and random noise introduced by the measurement apparatus. Random noise will include components of all the left singular vectors, U_i. The noise in the components with small singular values will be greatly amplified when its contribution to the conductivity update is calculated. Introduction of the Tikhonov factor limits the maximum amplification to $1/2\mu$ which occurs for $\lambda=\mu$ and, as the singular values tend to zero, the amplification factor also decreases to zero so preventing these noise components swamping the calculated conductivity update, see Figure 2.4b.

In general, conductivity perturbations farthest from the boundary result in the smallest voltage changes on the boundary, see Section 2.5. Hence, right singular functions with the variation closest to the centre of the region are those with the smallest singular values. Similarly, the first few singular functions are those with conductivity perturbations near the electrodes as these produce the largest voltage changes. Decreasing singular values are associated with conductivity variation with increasing spatial frequencies, decreasing amounts of symmetry and variation further from the boundary of the imaged region.

Regularisation, therefore, is equivalent to choosing the conductivity update from a space of smooth functions (i.e. functions with low spatial frequencies) with high degrees of symmetry and their principal variation near the boundary. The Tikhonov regularisation factor needs to be chosen to fit the singular noise spectrum of the electrical measurements. If the vector of experimental measurements is written in terms of the left singular vectors, $E=\sum_i (s_i+n_i)U_i$, where s_i and n_i are the singular components of the signal and noise respectively, then the signal to noise ratio in the conductivity update is:

$$SNR = \frac{\sum (A_i s_i)^2}{\sum (A_i n_i)^2}$$

The Tikhonov factor needs to be chosen to maximise the signal to noise ratio in the conductivity update. The spectrum of the noise, the n_i's, is uniformly distributed for most forms of random noise such as electronic noise in the data acquisition system but can be concentrated in a few singular components for systematic errors such as inaccuracies in electrode placement. The singular spectrum of the experimental measurements is typically strongly weighted towards the first singular components with the largest singular values. Thus, large Tikhonov factors include only the most reliable data with the highest signal to noise ratio.

In practice, the singular spectrum of the noise in the experimental measurements is unknown, so the Tikhonov factor is adjusted dynamically based on the reliability of the conductivity update. A conductivity update with large amplitudes near the centre of the region is indicative of too small a Tikhonov factor. Sophisticated schemes for the dynamic adjustment of the Tikhonov factor are described by Hebden, [42]. In practice these were found to be unnecessary. A conservatively large Tikhonov factor kept constant throughout reconstruction was found to work as well as other methods. This minimizes the contamination of the image with conductivity components which produce small voltage measurements on the boundary. Once these are included in the image they are very difficult to remove. The disadvantage of this conservative approach is that resolution is lost and large conductivity gradients are smoothed.

2.5 The Derivative Matrix

In order to calculate the conductivity update it is necessary to calculate the Jacobian matrix, J, whose coefficients are $\partial E_{ij}/\partial s_k$, where E_{ij} is a measurement of the boundary voltage induced by the application of a current pattern and s_k is a parameter of the conductivity, see Section 2.3.3. Approximations to this matrix, sometimes called the *sensitivity matrix*, have been calculated by a number of other workers in the field. Kim [47] and Tarassenko [79] use a perturbation technique to calculate a finite difference approximation to this derivative. The calculation of each coefficient of the matrix involves the construction of a finite element model for the given current pattern and conductivity component. This scheme is impractical for iterative reconstruction techniques that require repeated calculation of the derivative matrix. Yorkey [86] and [87] describes a method of calculating derivative matrices, based on the Compensation Theorem, where the forward model is a resistor network.

A more efficient scheme to calculate the derivative matrix is given by Breckon [9] where it is shown that to a linear approximation:

$$\langle M_i,(R(\sigma_m+\Delta\sigma) - R(\sigma_m))J_j\rangle \approx \left(\frac{\partial E_{ij}}{\partial s_k}\right)\Delta s = \int_\Omega \Delta\sigma \nabla\phi_i.\nabla\psi_j \qquad (2.5a)$$

where ϕ_i is the potential field induced in the region Ω with conductivity σ_m by the application of the current pattern J_i. The conductivity update is expressed as a finite linear combination of independant continuous functions: $\Delta\sigma=\Sigma s_i B_i=\mathbf{s}^T\mathbf{B}$. The lead

field, ψ_j , is the potential field that would be induced if the measurement pattern, M_i, were applied as a current pattern. From Equation 2.5a the following expression for the ij,k^{th} element of the Jacobian matrix may be derived:

$$\frac{\partial E_{ij}}{\partial s_k} = \int_{\Omega} \chi_k \nabla\phi_i . \nabla\psi_j \qquad (2.5b)$$

where χ_k is the Finite Element nodal basis function associated with node k.

The derivative is much easier to calculate in this form as the finite element model of only a single conductivity distribution need be calculated. A further simplification is obtained if the conductivity distribution is defined in the finite element approximation space. Once the potential fields have been calculated the coefficients of the derivative matrix may be readily calculated from the S(ij,k,element_shape) data.

Equation 2.5a explains why boundary voltage measurements are relatively insensitive to interior conductivity perturbations. The sensitivity of voltage measurement E_{ij} to a conductivity change at x is $\nabla\phi_i(x).\nabla\psi_j(x)$. For optimal experimental measurements on a uniform disk, see Section 2.2.6, the sensitivity of the measurement E_{ii} to a conductivity change at $x=(r,\theta)$ is proportional to k^2r^{2k-2} where $M_i=J_i=(1/\pi)\cos k\theta$. Thus the sensitivity drops off dramatically as perturbations are made further from the boundary

2.6 Solving the Newton System

Once the rows of the derivative matrix, $(\partial E_{ij}/\partial s_k) \in R^{M\times N}$, have been calculated it is necessary to construct the system of equations required to solve the regularised Least-Squares problem, Equations 2.3c,d. There are two cases to consider, M>N and M<N.

2.6.1 Overdetermined Systems

If the number of independent, experimental measurements used for reconstruction is larger than the number of parameters used to model the conductivity distribution, then the system $J\Delta\sigma=E$, where J is the Jacobian matrix, is technically overdetermined. However, due to the decay of the singular values compared to the errors in modelling and experimental measurements, the system may be numerically

underdetermined. The regularised Least Squares matrix, $(J^TJ+\mu I)$, is positive definite, symmetric and dense, and so Cholesky factorization is the standard solution method. Without regularisation the Least Squares system may not be positive definite and so Cholesky factorization could fail. By factorizing $(J^TJ+\mu I)=LL^T$ the Least Squares system can be solved by forward and backward substitution. An alternative scheme is suggested by recognizing that the matrix $(J^TJ + \mu I)$ can be written as D^TD where D^T is the augmented matrix $(J^T \mid \mu^{1/2}I)$. If D is written as the product of a orthonormal matrix Q and an upper triangular matrix R then the Least Squares system may be calculated via:

$$(J^TJ + \mu I)\Delta\sigma_m = D^TD\Delta\sigma_m = R^TQ^TQR\ \Delta\sigma_m = R^TR\ \Delta\sigma_m = J^TE.$$

This scheme allows us to factorize the matrix $(J^TJ + \mu I)$ without explicitly forming the product J^TJ. As the Jacobian matrix is ill-conditioned the product matrix, J^TJ, is far more so since $K(J^TJ) = K(J)^2$ where K(A) is the condition number of the matrix A. By not forming this product we expect to obtain better accuracy in the result. To decide which of these two schemes is numerically faster we need to compare the number of operations for each of the steps of these algorithms. The table below lists the number of operations needed for the steps involved in the solution of the Least Squares system as calculated by Golub and Van Loan, [33].

	Operation	Number of Floating Point Operations
A	form the product J^TJ	$N^2M/2$
B	Cholesky factorization of $(J^TJ + \mu I)$	$N^3/6$
C	QR factorization of $[J \ \ \mu^{1/2}I]$	$N^2(M + 2N/3)$
A+B	steps A and B together	$N^2(M/2+N/6)$

Steps A and B together require fewer operations what step C and so the Cholesky factorization method is faster than using QR factorization A comparison of the efficiencies of each of these algorithms in a parallel environment is required to determine which is better, see Sections 6.5.6 and 6.10.

2.6.2 Underdetermined Systems

If the number of independent experimental measurements is less than the number of conductivity parameters, i.e. M<N, the system is underdetermined and the

desired solution is the minimum norm $\Delta\sigma_m$ in the subspace spanned by the columns of J. The regularised system to be solved is:

$$(JJ^T + \mu I)Y = E \qquad \Delta\sigma_m = J^T Y$$

The same analysis can be applied to this set of equations and the Cholesky factorization method produces the result with fewer operation. In the reconstruction algorithms described in Chapter 5 which use underdetermined systems the number of rows of the matrix J is considerably less than the number of columns. In this case solution of the Least Squares system is an insignificant part of the whole reconstruction step.

2.7 Conclusions

In this chapter the general framework for Newton based, iterative reconstruction algorithms has been explored. A new definition for an experimental measurement in EIT has yielded expressions for optimal current and measurement patterns which are applicable to a wide variety of tomographs. These patterns optimise the size of experimental measurements and the distinguishability between conductivity distributions. Their use has lead to the development of an efficient reconstruction algorithm which is described in detail in Section 7.5.2.

Chapter 3

Forward Modelling in EIT

3.1 Introduction

Our ability to predict the current and potential fields in a conductor accurately gives information vital to the design of tomographs. The forward model is also a critical stage of any iterative reconstruction algorithm. Modelling errors are equivalent to data collection errors in their effects on the image produced by an impedance tomograph. The larger of the modelling and data collection errors limits the resolution of practical tomographs. For a forward model to be incorporated into an iterative reconstruction algorithms it must not only be accurate but also fast. Chapter 3 investigates a range of mathematical models used for EIT. A semi-analytic solution of one of the most successful mathematical models is developed and its implications in the design of tomographs is explored.

3.2 Forward Modelling

The solution of the forward modelling problem involves the calculation of the voltages induced by the application of electric current to the surface of a region with known conductivity. It is a vital part of the Newton algorithm described in this thesis and an important check on the results of all other reconstruction algorithms. The problem posed is the solution of Equation 2.2a in the interior of a region(Ω) with boundary conditions imposed on the surface. The assumption that there are no sources or sinks of current in the interior of the region leads to the equation:

$$\nabla.\sigma\nabla\phi = 0 \qquad \text{in } \Omega$$

where σ is the known conductivity distribution and ϕ is the potential field. This is a second order, elliptic, partial differential equation in ϕ. Sufficient boundary conditions need to be specified to make the solution of this equation unique. These may be potentials on the boundary, known as Dirichlet conditions, or current densities crossing the boundary, known as Neumann conditions, or a combination of both. For the calculated potential field to be unique at least one Dirichlet condition needs to be specified. This is commonly achieved by setting the average potential to be zero over a region considered to be earthed.

3.3 Analytic Solutions

Analytic methods for solving these problems are limited to simple conductivity distributions, boundary conditions and domains. In two dimensions, classes of solutions exist which are linked by conformal mappings. The potential inside domains with uniform conductivity and polygonal boundary may be calculated using the method of Schwarz and Christoffel [60]. Circular regions with uniform conductivity, enclosing circular regions with, possibly a different, uniform conductivity may be conformally mapped to known analytic solutions [79]. Where the boundary conditions are complicated a linear combination of these solutions may be required. In three dimensions known analytic solutions are restricted to regular shapes such as concentric spheres or concentric cylinders with uniform conductivity regions. In EIT these analytic methods are only useful for calculating the potential induced by an initial approximation to the conductivity e.g. the NOSER algorithm [16]. If a more elaborate first guess is available or iterative methods are to be used, it is necessary to use numerical techniques.

3.4 Approximate Solutions

The three most readily available numerical methods for solving partial differential equations are the Finite Difference Method (FDM), the Finite Element Method (FEM) and the Boundary Element Method (BEM). In the FDM the potential in the region is approximated by its value at nodes lying on a regular grid. A system of linear equations is generated by replacing the differential operators by difference operators. This reduces the calculus problem to a linear algebra problem requiring the solution of a, possibly large, sparse set of simultaneous equations. In the FEM the region is decomposed into a mesh of irregular polygons or polyhedra (known as finite elements). The potential is approximated by interpolating within elements using a set of element basis functions. These basis functions are polynomial within elements and so the resultant potential approximation space is piece-wise polynomial. A set of simultaneous equations is constructed by applying a weak form of the differential equation over the region. This also results in a set of sparse, linear equations. The conductivity distributions for the FDM and FEM are also represented in the relevant approximation space. The BEM is different in that the conductivity distribution is assumed to be uniform between surfaces. These surfaces are decomposed into meshes of elements of one dimension less than the region being modelled. An integral form of the differential equation is constructed using Green's Theorem leading to a set of equations linking nodal potentials. The major advantage claimed by the BEM is that in many cases the number of nodes required to model a region is smaller than that required by the FDM or the FEM. This advantage is

balanced by the fact that the resulting equations are dense and so require more numerical operations to solve.

For the work described in this thesis it was decided to use the FEM. There is no consensus as to which of the three methods provides solutions of a given accuracy with the minimum computational effort. The FDM has the advantage that the regular grids are easy to generate, the programs are less complex and the resultant regular systems of equations may be solved by highly efficient algorithms. For EIT this advantage is countered by the difficulty in applying boundary conditions and adapting the regular meshes to irregular regions. The irregularly shaped elements of the FEM are fitted to the regions encountered in the application of EIT. The FEM is also more adaptable to the application of complex boundary conditions. Lastly, the size of elements in the mesh can be adjusted to take into account the complexity of the potential field being modelled resulting in greater accuracy and fewer nodes. The BEM was not used because the piece-wise constant approximation to the conductivity would be difficult to adapt to an iterative reconstruction algorithm. Changing the number and position of the interior surfaces during reconstruction would introduce a very large numerical overhead during each iteration of the algorithm.

The use of the FEM has become close to universal in the EIT field. One group in Nijmegen, the Netherlands (Van Oosterom, [59] and [65]) uses the BEM for modelling electrical fields in the body but has not used the method as part of a reconstruction algorithm. Following Tarassenko [79] and Breckon [9] we chose to use the Finite Element Library of Greenough and Robinson [35] now distributed by Numerical Algorithm Group (NAG).

3.5 Modelling electrodes

The unique solution of the problem posed in Section 3.2 depends upon the boundary conditions we impose on the surface of the region. On the physical region to be imaged these boundary conditions must be consistent with the application of current through electrodes in contact with the surface. To accurately predict the physical measurements made on the region it is necessary to model the physical processes occurring throughout the region, and in particular on the boundary near electrodes.

3.5.1 The Continuous Model

Several levels of boundary condition complexity have been investigated by

Cheng *et al* [16] and the results compared with phantom measurements. The first level of approximation ignores the localizing effects of electrodes completely and considers a current pattern which injects current I_i on electrode i: $I_i = \cos k\theta_i$, to be equivalent to the application of a continuous current density $J(\theta) = A_c \cos k\theta$. A_c is a normalizing constant to match the total current crossing the boundary. Voltage measurements on the current delivering electrodes were equated to the potential on the boundary at the middle of electrodes. This formulation, known as the continuous model, was found to be totally inadequate for the prediction of voltage measurements on physical phantoms.

3.5.2 The Gap Model

The continuous model was refined by recognizing that current only crosses the boundary under the electrodes. The current density under electrodes was assumed to be uniform. This leads to the gap model defined by the boundary conditions:

$$\sigma \nabla\phi.\mathbf{n} = 0 \qquad \text{between electrodes} \qquad (3.5a)$$

$$\sigma \nabla\phi.\mathbf{n} = I_i / A_i \qquad \text{beneath electrode i.} \qquad (3.5b)$$

where A_i is the area of electrode i and $\mathbf{n}$ is a vector perpendicular to the boundary. This model has been successfully applied in EIT reconstruction and is the one presently used by the Rensselaer group.

3.5.3 The Complete Model

Further refinement is possible. The gap model assumes that only the current delivered to an electrode from the driving electronics crossed the boundary and that this spreads itself out evenly along the electrode-region interface. Analysis of the two dimensional situation where a finite electrode delivers current into a half space with uniform conductivity tells us to expect a square root singularity in the current density at the edges of electrodes, [73]. Furthermore, there is the problem of contact between the electrode and the region. Many workers, e.g. Pollok [75], Gedes *et al* [29] and Yoshida [88], have reported the existence of a high impedance layer between electrodes and electrolytes. Lui [56] has linked the characteristics of this high impedance layer to a fractal measure of the roughness of the interface. A proportion of the voltage measured on a current carrying electrode is due to current crossing this high impedance layer. The phantom measurements of Cheng *et al* suggest a value for this contact impedance (z) large enough to seriously contaminate EIT voltage

measurements. The Complete model was designed to take these effects into account:

$$\sigma \nabla\phi.\mathbf{n} = 0 \qquad \text{between electrodes.} \qquad (3.5c)$$

$$-\int_{\gamma_i} \sigma \nabla\phi.\mathbf{n}\, d\gamma = I_i \qquad \text{for electrode i.} \qquad (3.5d)$$

$$\phi + z_i \sigma \nabla\phi.\mathbf{n} = V_i \qquad \text{beneath electrode i.} \qquad (3.5e)$$

This model was found to be very successful in the prediction of voltage measurements on phantoms. Its difficulty in practice is the estimation of the contact impedances z_i. Cheng assumed a single constant value beneath all the electrodes, yet in situations other than highly controlled phantom experiments this is unlikely to be the case. In clinical situations, z_i is expected to be a wildly varying function of both position and time. This observation calls into question the practice of making voltage measurements on current carrying electrodes. The current density distribution under electrodes is determined by the contact impedance there. This in turn determines the potential in the region and hence the voltage measurements made on even passive electrodes.

3.6 Semi-Analytic Solutions in Two Dimensions

Analytic solutions to the forward problem provide an important test for the numerical algorithms used for EIT reconstruction. They provide a standard against which the accuracy and speed of convergence of the numerical methods are measured. The best standards are analytic solutions to problems of similar complexity as those routinely solved during EIT reconstruction. Standard results exist for the analytic solution of Laplace's equation for simple regions and boundary conditions. Problems with more complex boundary conditions may be solved by forming linear combinations of these solutions.

3.6.1 The Boundary Fourier Method

The Boundary Fourier Method (BFM), Paulson *et al* [70], calculates linear combinations of solutions of Laplace's equation on a disk which solve problems with complex boundary conditions, such as those proposed by Cheng *et al,* to arbitrary precision. Consider a two dimensional, homogeneous disk of radius R_1 and conductivity σ_1 surrounded by an annulus of outer radius R_0 and conductivity σ_0. By solving Laplace's equation on this configuration, together with the continuity of ϕ and $\sigma(\partial\phi/\partial\mathbf{n})$ on the inner circular boundary, it is easy to show that trigonometric boundary current densities $j_k(\theta)$ are associated with boundary voltages $v_k(\theta)$. Hence

$$j_k(\theta) = \cos k\theta \qquad \Leftrightarrow \qquad v_k(\theta) = \frac{R_0}{k\sigma_0}\Lambda_k j_k(\theta)$$

where

$$\Lambda_k = \frac{1-\mu\Gamma^{2k}}{1+\mu\Gamma^{2k}}, \qquad \Gamma=\frac{R_1}{R_0}, \qquad \mu=\frac{\sigma_1-\sigma_0}{\sigma_1+\sigma_0}.$$

If we restrict ourselves to even functions of θ, then, by Fourier decomposition, all solutions of Laplace's equation on the disk can be written as linear combinations of these basis solutions as follows:

$$J(\theta) = \sum_k a_k j_k(\theta) \quad \Leftrightarrow \qquad V(\theta) = \sum_k a_k v_k(\theta)$$

where k can run from 1 to infinity. An approximate solution to the problem described by the Complete model, Equation 3.5c,d,e, may be found by obtaining the equations linking the Fourier coefficients a_k, $1\le k\le n$. Consider, therefore, a disk driven by currents applied through L=2m, $m\in \mathbf{N}$, identical, symmetrically placed electrodes on the surface, each delivering a current I_i at a voltage V_i. The boundary conditions 3.5c and 3.5e may be rewritten:

$$zJ(\theta)=\begin{cases} V_i - v(\theta) & \text{on electrode i;} \\ 0 & \text{elsewhere.} \end{cases}$$

Linear equations linking the Fourier coefficients may be obtained by multiplying each side of this equation by $\cos(m\theta)$ and integrating around the boundary as follows:

$$z\pi a_m = \sum_i \left(V_i \int_{\gamma_i} \cos(m\theta)\, d\gamma - \frac{R_0}{\sigma_0}\sum_k \frac{a_k}{k}\Lambda_k \int_{\gamma_i} \cos(m\theta)\cos(k\theta)\, d\gamma \right) \quad (3.6a)$$

where the integrals are over γ_i, the segment of the boundary under electrode i. The unknowns in these equations are the n Fourier coefficients a_k and the electrode voltages V_i. There are as many independent, linear equations as Fourier coefficients but the system is underdetermined due to the unknowns V_i. The equations necessary to make a fully determined system may be formed by applying boundary condition 3.5d to each electrode i = 1,2,...L;

$$\int_{\gamma_i} J(\theta)R_0\, d\theta = I_i \quad \Rightarrow \quad \sum_k a_k R_0 \int_{\gamma_i} \cos(k\theta)\, d\theta = I_i \qquad (3.6b)$$

The system of linear equation formed by the two equations 3.6a and 3.6b, may be solved to yield the Fourier coefficients a_k and the electrode voltages V_i.

The BFM can be extended to three dimensional, spherical regions using basis solutions in the form of surface spherical harmonics [76]. These functions, related to Ferrer's polynomials, are eigenfunctions of the three dimensional transfer impedance function and so are the analogue of simple trigonometric functions on two dimensional circular surfaces. Such solutions were found to be inefficient to calculate, due to the large number of basis functions needed to adequately model electrode edge effects and the difficulty of using a standard based on a spherical region. Similarly, solutions based on linear combinations of Bessel functions can be found for cylindrical regions.

3.6.2 The Boundary Fourier Method and EIT

Figure 3.6 displays the boundary current density and potential predicted by the BFM for a homogeneous disk of radius 15 cm driven by 32 symmetrically placed electrodes covering 50% of the surface. The first 1000 non-zero frequencies up to a wave number of 16000 were used. Doubling the number of frequencies used from 500 to 1000 resulted in less than a 0.5% change in the predicted voltages. It is clear that the current density across electrodes is far from uniform. The singularities in the current density field are associated with Neumann conditions which are discontinuous on the boundary and ohmic conductors, [76] and [73]. The BFM shows large current densities near the edges of electrode which are prevented from being singularities by the contact impedance under the electrodes. The current crossing the boundary beneath passive electrodes is significant. As electrodes are modelled as perfect conductors they "short circuit" the part of the boundary they press against. A proportion of the current applied to image the region is instead *shunted* along electrodes, decreasing the amount of useful information provided by the experiment. Total shorting of the boundary under each electrode is prevented by the layer of high contact impedance. Cheng *et al* found the magnitude of the contact impedance to be proportional to the resistivity of a uniform disk, $z \propto \rho$. In this case the shunt currents are independent of the impedance of the disk. For a uniform disk with current being driven between a diagonal pair of four symmetrically placed electrodes, half of the applied current is shunted through the passive electrodes when they cover 80% of the boundary. This proportion is drastically reduced when more electrodes are used.

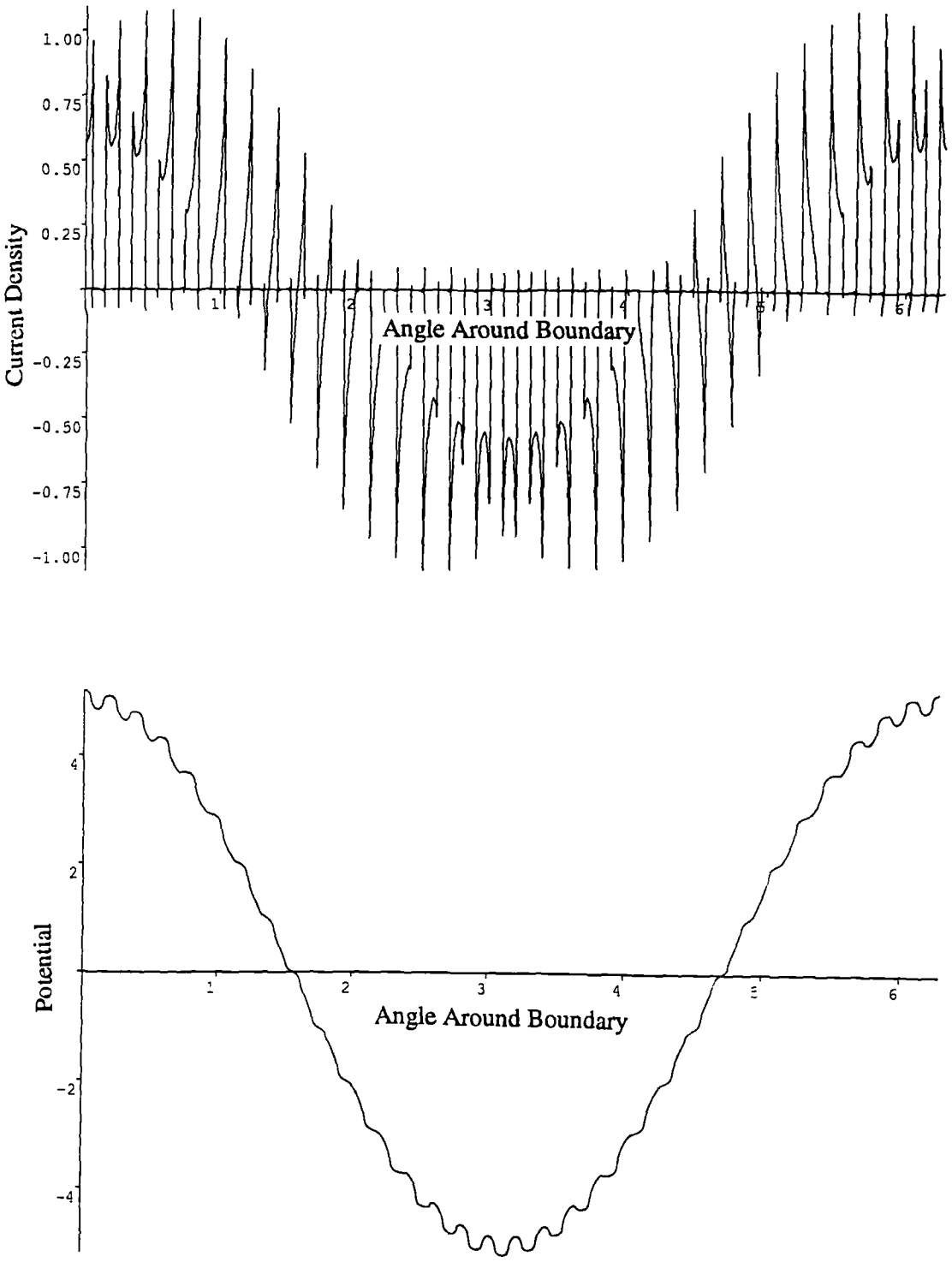


Figure 3.6 *The boundary current density and voltage on a disk of uniform conductivity and radius 15cm, driven by thirty two symmetrically placed electrodes covering 50% of the surface as predicted by the Boundary Fourier Method.*

For systems with a large number of electrodes the loss of information due to shunting is important only when the size of electrodes is very large i.e. when the gaps between electrodes are small.

3.7 Electrode Configurations

The results obtained from the BFM clearly show the difficulties faced by the forward modelling component of an EIT system. The voltages measured on large electrodes are strongly determined by the profile of the current density beneath them. The electrode voltage is very sensitive to the impedance of the region just beneath the edges of electrodes and to variations in the contact impedance there. In a clinical situation this would be very difficult to control and so accurate modelling would require the difficult determination of the contact impedance between each electrode and the boundary.

3.7.1 Separate Current Drive and Voltage Measurement Electrodes

Several alternative electrode configurations are possible. One possibility is to measure voltages on small, passive electrodes interleaved with much larger, current driving electrodes. This *hybrid* system is explored in detail in Paulson [71], see Appendix, and in section 4.11. Voltage measurements half-way between drive electrodes are the least sensitive to variations in the current density beneath electrodes. The insensitivity of voltage measurements made mid-way between electrodes to the current density profile on the electrodes can be demonstrated by comparing the mid-point voltages predicted by the gap model and those calculated by the BFM. The predicted voltages show differences of less than 0.1% for trigonometric current patterns with a spatial frequency less than eight applied to a thirty two electrode system. The current density profile under an electrode is determined by the variation in the contact impedance there. This insensitivity to contact impedance and current density profile is countered to a degree by the loss of voltage amplitudes for patterns with high spatial frequencies. The gap model predicts that the amplitude of the voltage pattern measured at points half-way between drive electrodes is a half of what it would be for measurements on the current driving electrodes. When the current driving spatial frequency equals half the number of electrodes the amplitude of the voltage pattern is zero. In practice the amplitude of the voltage pattern degrades faster than the gap model predicts, see Section 7.2.1. This limits the imaging resolution near the boundary. However, most of the information about objects in the interior of the region is in the low spatial frequency patterns which are relatively unchanged.

3.7.2 Compound Electrodes

Another configuration implemented by Eung, [24], uses compound electrodes with two, electrically isolated, surfaces. A large annular region delivers current to the imaging region while a small electrode within the annular region measures voltage. Van Oosterom, [66], studied directional sensitivity of electrodes of this design for ECG. This configuration does not suffer from the voltage attenuation for high spatial frequencies as the voltages are measured near the points on the boundary where current is applied. For the same reason we would expect higher sensitivity to current density variation across the electrode and to contact impedance. At present it is not clear which of these two configurations has the advantage.

3.7.3 Independent Measurements

The electrode configuration determines the number of independent measurements that can be made with a system and hence the rank of the Jacobian matrix. This limits the number of parameters in the conductivity distribution that can be reconstructed and so determines the resolution. A tomograph with N electrodes used for both current driving and voltage measurement yields N(N–1)/2 independent experimental measurements, [9]. Of the N^2 measurements made by measuring the voltages on N electrodes for a basis of N different current patterns, N(N–1)/2 are dependent due to the Reciprocity Theorem. This theorem states that driving current between electrodes A and B and measuring the voltage between electrodes C and D yields the same result as driving current between electrodes C and D while measuring the voltage between electrodes A and B, Geselowitz [30]. Another N measurements are dependent due to the normalization of each voltage and current pattern to have an average value of zero. Thus, of the N^2 measurements, at most N(N–1)/2 are linearly independent with further redundancies introduced by other symmetries, see [52].

The Reciprocity Theorem is expected to hold approximately for compound electrodes so the system of Hua would be expected to have the same number of independent measurements. For the hybrid arrangement, however, reciprocity does not apply as currents are applied and voltages measured at different places on the boundary. This system would have twice the number of independent measurements for the same number of current driving electrodes. The above analysis tells us the theoretical rank of the Jacobian matrix. However the numerical rank is determined by the noise level and the decay of the Jacobian's singular values. Although the hybrid arrangement has twice the number of independent equations the singular values would be expected to decay more rapidly due to the increasing insensitivity of the

system to current patterns with high spatial frequencies. Both these systems are worthy of investigation and in all probability, both will find their niche.

3.8 Conclusions

In this Chapter a range of mathematical models relevant to EIT have been investigated. The Complete Model is known to give results which agree well with physical measurements and so is the one best suited for use in reconstruction algorithms. A semi-analytic method for the calculation of electrical fields consistent with this model has been developed. Investigations based on the use of this model have suggested a new configuration of electrodes to be used for EIT measurements. This configuration yields experimental measurements which are relatively insensitive to contact impedance.

Chapter 4

The Finite Element Method in EIT

4.1 Introduction

The Finite Element Method is used to perform forward modelling in a large majority of tomographs which use iterative reconstruction algorithms. It is well understood and it is known to converge to exact solutions for many of the models used in EIT. As a forward model needs to be calculated at each iteration of the reconstruction algorithm, the calculations involved must be minimised. A Finite Element Model can be highly optimsed to execute quickly on a digital computer. In this Chapter the mathematics of Finite Element Modelling in both two and three dimensions is investigated. In particular, boundary conditions consistent with the application of current through surface electodes are introduced to Finite Element Models.

4.2 The Finite Element Method

As stated in Section 3.4 the OXPACT system uses the Finite Element Method (FEM) to predict the voltage measurements made on a region of given conductivity after the application of a current pattern to the boundary. This is a problem that needs to be solved repeatedly for different conductivity distributions during an iterative reconstruction.

The region is divided into a mesh of irregular polygons or polyhedra known as elements with nodes lying on their common vertices. Associated with each node is a basis function which has the value one at that node and decays polynomially to zero at all the other nodes in adjacent elements. In all other elements the basis function is zero. The basis function associated with node i is called its characteristic function χ_i Scalar functions of position within the mesh, such as the potential and conductivity, may be approximated by linear combinations of these basis functions:

$$\phi \approx \Sigma \chi_i \phi_i = \chi T . \phi^e$$

where χ is a vector of characteristic functions and ϕ^e a vector of nodal potentials.

The FEM uses the Rayleigh-Ritz-Galerkin method to transform the differential equation 2.2a into a set of simultaneous equations, one for each node, of the form:

$$\int_\Omega \chi_i(\nabla.\sigma\nabla\phi) = 0$$

By applying Green's theorem this integral may be rewritten:

$$-\int_\Omega \sigma\nabla\chi_i.\nabla\phi + \int_{\partial\Omega}\chi_i\sigma\nabla\phi.n = 0$$

$$\Rightarrow -\left(\int_\Omega \sigma\nabla\chi_i.\nabla\chi^T\right).\phi^e = \int_{\partial\Omega}\chi_i\sigma\nabla\phi.n \qquad (4.2)$$

When all these equations are collected they may be expressed concisely in matrix notation as:

$$K\phi^e = \mathbf{F}.$$

Many other equilibrium and potential problems reduce to equations of the same form so, in FEM jargon, K is known as the system stiffness matrix and **F** as the system forcing vector. The original partial differential equation problem has been reduced to the numerical calculation of the system stiffness matrix and forcing vector coefficients and the solution of the matrix equation.

The system forcing vector contains the information in the Neumann boundary conditions. For the forward problem of EIT the forcing vector is a piecewise polynomial approximation of the current crossing the boundary expressed in the mesh basis restricted to the boundary.

4.3 Calculation of The System Stiffness Matrix

The system stiffness matrix is a discrete approximation to the transfer admittance operator, $R^{-1}(\sigma)$. It is symmetric and positive definite if the conductivity obeys the physical restriction of being greater than zero and bounded above. A large part of the computational effort of the FEM in two dimensions occurs during the calculation of the coefficients of the system stiffness matrix. In practice the system stiffness matrix is calculated by summing the contributions from element stiffness matrices, K_e, associated with each element. The coefficients of K_e are the volume integral of Equation 4.2 restricted to the region of a single element Ω_e. The characteristic functions of element nodes, restricted to an element, are known as the shape functions, N_i. Typically the nodes in element e are given an element node

numbering from one to n(e) which is different from the mesh node numbering. The element node numbering is specific to a single element. The i,jth element of the element stiffness matrix can be expressed as:

$$K_e^{ij} = \int_{\Omega e} \sigma \nabla N_i . \nabla N_j$$
$$= \sum_{k=1}^{n(e)} \sigma_k \int_{\Omega e} N_k \nabla N_i . \nabla N_j = \sum_{k=1}^{n(e)} \sigma_k \, S(i,j,k,e) \qquad (4.3)$$

for $1 \leq i,j \leq n$ where n is the number of nodes in that element. If the conductivity is approximated in the FE basis the coefficients of K_e are a weighted sum of the nodal conductivities, σ_k, with the weights given by integrations of functions of the shape functions over the element. Typically the conductivity is parameterised on a coarser mesh than the potential and so the nodal conductivity values need to be found by interpolation. The weights, S(i,j,k,element), are independent of element position and orientation and so computational savings may be made by precalculating and storing a set of weights for each element shape used in the FE model. Typically many elements with the same shape will exist in a mesh. Each set of element weights is symmetrical in i and j and so further computational savings can be made by calculating and storing only the upper triangular part of the S(i,j,k,element) matrix in a S(ij,k,element_shape) data structure.

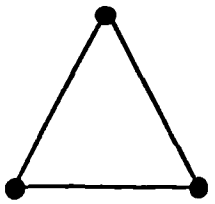
4.4 Finite Element Modelling in Two Dimensions

Each element in the finite element model is a polygon with nodes at each vertex and possibly on the edges or in the interior. Each node is associated with a shape function which is non-zero only within elements which include that node. It is a polynomial function of position which takes the value one at the node in question and zero at all other nodes. Thus, three sets of data are required to define an element; the position of the nodes that form the element, a list of edges defined by the nodes at the ends of each edge, and the shape function associated with each node. A mesh is a collection of non-overlapping elements such that the nodes that lie on edges shared by adjacent elements belong to both.

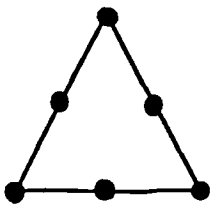
4.5 Triangular Elements

A host of elements are in common use of which the simplest is the Turner or Courant triangle. Historically this was the first element to be studied during the development of the Finite Element method, [18]. The Turner element is triangular

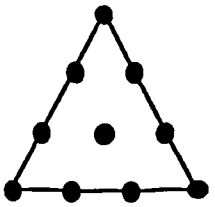
with a node at each vertex and linear interpolation functions. The characteristic functions are pyramid shaped and formed from the shape functions associated with a node at the junction of three or more elements. The associated approximation space contains functions which are continuous across element boundaries and linear within elements. It is a subspace of H^1 as the first derivatives are piecewise constant. A major step in the development of the Finite Element method was the generalisation of elements to higher order approximations. If the element shape functions are quadratic functions of position i.e. $f(x,y)=a+bx+cy+dx^2+exy+fy^2$, then three more degrees of freedom per element are introduced. These extra degrees of freedom can be associated with nodes which, if placed on the mid-points of the edges of the triangle, guarantee continuity of the basis functions across element boundaries. This process can be continued to cubic approximations with the ten noded triangle and beyond, see Figure 4.5.



Turner Triangle.
Three nodes.
Linear interpolation



Six Noded Triangle.
Quadratic interpolation



Ten Noded Triangle.
Cubic interpolation

Figure 4.5 *Triangular finite elements for two dimensional modelling.*

The approximation spaces associated with all these elements belong to the space of continuous functions, C^0. Extending this formulation to approximations spaces that have continuous first derivatives across element edges is surprisingly difficult. The simplest C^1 element is the 21 noded triangle with quintic approximation functions, [78]. Triangles of lesser degree that match first derivatives at the vertices are in use but are not guaranteed to converge to the correct solution as

the mesh is refined.

4.6 Quadrilateral Elements

The quadrilateral cousin of the Turner triangle is the four noded rectangle with a bilinear approximation function, $f(x,y) = a+bx+cy+dxy$. The shape functions are graphically known as pagoda functions and are the product of the piecewise linear roof functions of x and y. Only the isoparametric form of this element is continuous across element boundaries, [78]. Quadrilaterals are potentially more accurate than triangles due to the non-linear, xy, "twisting" term and have the advantage that fewer are needed to fill a region. However, they approximate smooth boundaries badly. This problem can be overcome by mixing isoparametric bilinear quadrilaterals with Turner triangles on the boundary. Quadratic elements can be generalised in the same way as triangles and the use of both biquadratics and bicubics is reported in the literature.

4.7 Convergence of The Finite Element Model

Many more complex elements are reported in the literature to solve problems with special boundary conditions or boundary shapes. However, the majority of finite element problems are solved using triangular or quadrilateral elements, or their three dimensional equivalents. There are several reasons for this: as the complexity of elements increases the task of mesh generation becomes increasingly more difficult. Elements with greater numbers of nodes require considerably more calculation to form the element stiffness matrix. This is not only due to the larger element stiffness matrix but also the requirement to perform numerical integrations over polynomials of greater degree. The resulting system stiffness matrix is less sparse and so the finite element system requires more computational effort to solve.

Ultimately the forward problem in EIT is to predict, for a given conductivity distribution, the boundary voltages to an accuracy predetermined by the resolution demanded from the reconstruction algorithm. This accuracy is governed by the ill-posedness of the inverse problem and the minimum resolution of useful images. The ultimate performance of a tomograph is determined by the greater of the measurement and the modelling errors. Clearly there is no advantage to be gained from exceeding the precision of electrical measurements made on the region to be imaged, including those errors introduced by contact impedance and electrode placement. For iterative reconstruction algorithms the modelling must be both accurate and fast. The accuracy and speed of computation of a finite element solution is limited by the characteristics of the elements, coarseness of the mesh and the complexity of the boundary

conditions. There is no *a priori* way of determining which combination of these will yield the result to the desired accuracy with the minimum of computational effort.

The convergence of the finite element approximation, ϕ^a to the solution, ϕ, of a second order elliptic partial differential equation can be summarised in Equation 4.7 given by Strang, [78]

$$\frac{\| \phi - \phi^a \|_s}{\| \phi \|_{k+1}} \leq C\, h^{k+1-s} \qquad (4.7)$$

where s is non-negative and $\|f\|_s$ is the Sobolev norm of the function f. Here, the radius of an element, h, is defined as the radius of the smallest circle that can contain the element. For a mesh, the radius is the maximum radius of any of its elements. The degree of the interpolation polynomials, k>0, is defined as the maximum degree of polynomial present in all of its terms. For example, the degree of the bilinear rectangle is k=1 as the interpolating polynomial has the term xy but neither of the terms x^2 or y^2. In the case of three noded, linear, Turner triangles the L_2 error of the potential, corresponding to a Sobolev norm with s=0, has convergence of $O(h^2)$.

$$\frac{\| \phi - \phi^a \|_0}{\| \phi \|_{1+1}} \leq C\, h^{1+1-0} = C\, h^2 = O(h^2)$$

This inequality may be applied locally to show that in a region where any of the first k+1 derivatives of ϕ are large, the absolute error $\| \phi - \phi^a \|_s$ will be large. For the finite element solution to have uniform error across the region the element radius, h, needs to be locally tuned to the variation of the potential field. In practice this is impossible as the field inside the region is not determined from the known boundary conditions unless the conductivity is known. This is, of course, what we are trying to find out. For a homogeneous conductivity distribution on a disk and current patterns of $J=\cos(K\theta)$, the element radii would need to vary as $h(r) = Cr^{K-2}$; $K \geq 2$. Therefore, to adequately model the application of current patterns with high spatial frequencies, the meshes must become finer near the boundary of the region. For a given mesh the predicted voltages will become less accurate for fields with higher spatial frequencies.

4.8 Finite Element Modelling for EIT

Due to symmetries in our present experimental apparatus the region to be imaged can be modelled as if it were two dimensional, see Section 7.2. The potential we are modelling is the one induced by the injection of current through finite

electrodes on the boundary. These result in near singularities in the current density on the edges of these drive electrodes and so the Fourier spectrum of the current density on the boundary decays slowly. This implies large potential gradients near the edges of electrodes and hence very fine meshes are required there. The coarseness of the mesh imposes a low pass filter to the spatial frequencies in the finite element solution. For an accurate solution nodes need to be concentrated near the boundary and especially near the edges of electrodes. Limits also exist for the rate at which the element radii can change. For a mesh of triangular elements, no two angles subtended by any edge can sum to more than π or the system stiffness matrix may not be positive definite, [78]. It follows from this that elements should be close to equilateral and their radii must vary smoothly across the region.

For the EIT forward problem some experimentation led to the choice of Turner triangles as the preferred element. Although bi-linear quadrilaterals promise greater accuracy for the same number of nodes they were found to be difficult to combine into meshes with the desired degrees of symmetry. Quadrilaterals near the boundary which were not symmetrical with respect to the electrodes introduced errors into the surface potential that were not present with meshes of triangles. Equation 4.7, describing convergence of finite element solutions, suggests faster convergence for higher degrees of interpolating polynomial. The six noded quadratic triangles would be expected to yield more accurate solutions for the same number of nodes due to their quadratic interpolation functions. They were not used in this investigation due to the difficulties of mesh generation and the application of boundary conditions.

4.9 Mesh Generation

The program RMESH, originally created by Breckon [9], has been written to calculate a finite element mesh of Turner triangles covering a two dimensional disk. The nodes of the mesh lie on concentric circles and the layers of elements form annuli, see Figure 4.9. Nodes on the outer boundary are grouped into 'cells' where a cell describes the relative positions of all the boundary nodes that model the region of an electrode. This allows the user to define meshes where the nodes are concentrated around the edges of electrodes. For rings of nodes in the interior this clustering of nodes is smoothed out until the nodes are uniformly distributed near the centre. RMESH sorts the nodes to minimise the computational effort involved in factorizing the system stiffness matrix. The details of the algorithms used to achieve this are described in Section 6.5.5. RMESH forms lists of nodes on electrodes, and precalculates the S(ij,k,element_shape) data. This data is written to a file called a *mesh file* which includes all the information required to specify and perform calculations on a finite element mesh.

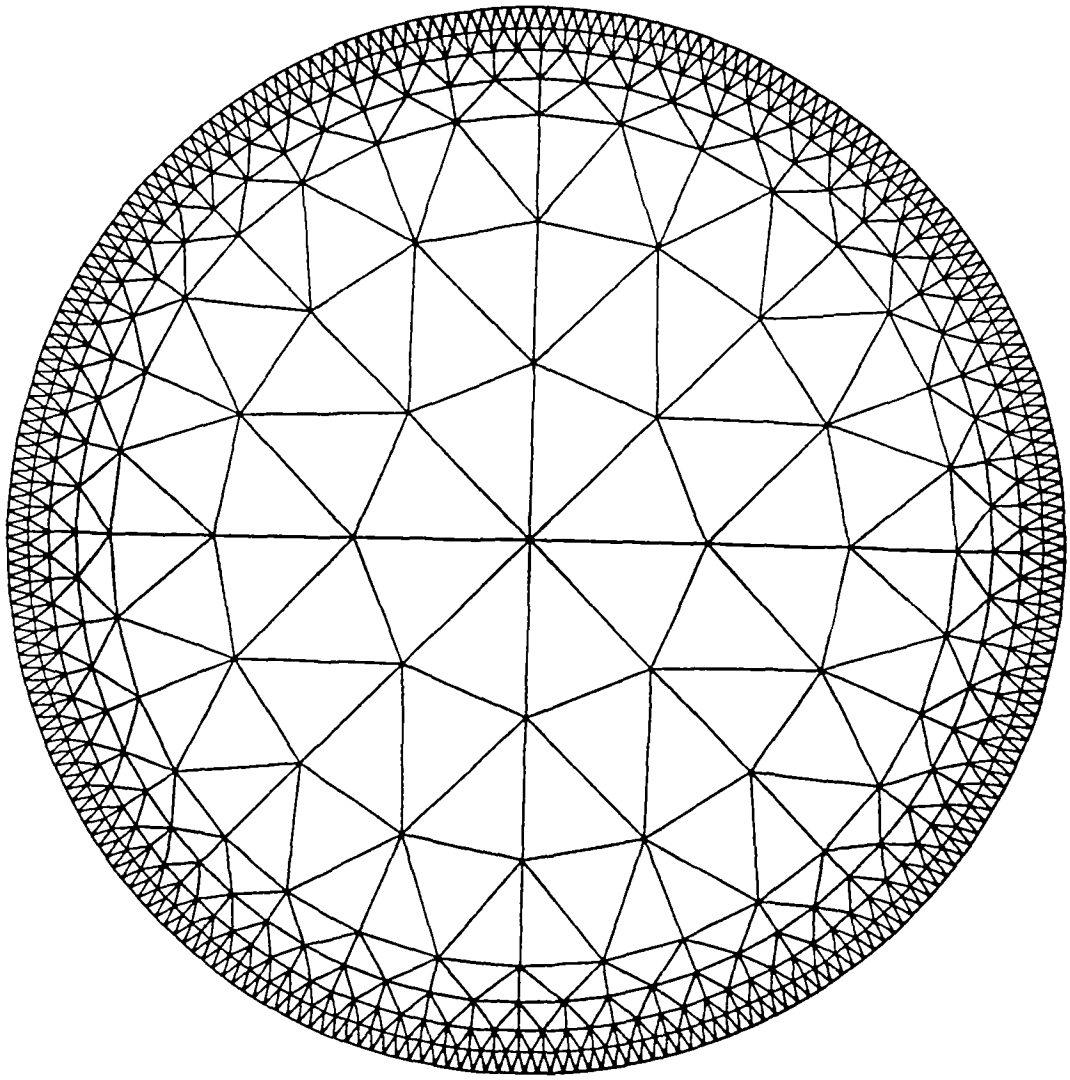


Figure 4.9 *A Finite Element mesh composed of Turner triangles used to model the potential on a two dimensional circular region driven by thirty two electrodes covering 30% of the boundary*

4.10 Finite Element Modelling of Electrodes

The physics of conduction around electrodes that leads to the boundary conditions described in Section 3.5 needs to be implemented in a finite element model used for electrical impedance reconstruction. The constraints involved in doing this are different depending on whether the voltage measurements are made on current-carrying electrodes or not. Systems such as the typical optimal current systems described by Newell, [64], and Ping, [74], measure voltages on large, current carrying electrodes. To model these electrodes to sufficient accuracy requires the sophisticated boundary conditions of Cheng [16]. These boundary conditions, Equations 3.5c-e need to be implemented in the finite element model. Equation 4.2 was the basis of the finite element model used to solve the conduction equation. The right hand side of this equation associates with each boundary node a weighted integral of the current density crossing the boundary. By introducing boundary condition 3.5e to this term for node i beneath electrode l it becomes:

$$\int_{\gamma_l} \chi_i \sigma \nabla \phi . n = \int_{\gamma_l} \chi_i \frac{V_l - \phi}{z} = \frac{V_l}{z} \int_{\gamma_l} \chi_i - \frac{1}{z} \int_{\gamma_l} \chi_i \chi^T \phi^e$$

where γ_l is the section of the boundary beneath electrode l and V_l is the voltage measured on electrode l. Combining these results into Equation 4.2 yields:

$$-\left(\int_{\Omega} \sigma \nabla \chi_i . \nabla \chi^T + \frac{1}{z} \int_{\gamma_l} \chi_i \chi^T \right) \phi^e = \frac{V_l}{z} \int_{\gamma_l} \chi_i \qquad (4.10a)$$

This equation becomes a row of the system stiffness matrix for node i beneath electrode l. If node i is not beneath an electrode both boundary integrals are zero and they need not be calculated. The linear system of equations defined by Equation 4.10a is underdetermined due to the introduction of the unknown electrode voltages, V_l. A further equation for each electrode may be derived from boundary condition 3.5d. These equations constrain the net current crossing the boundary beneath each electrode to be the current delivered by the driving electronics, I_l, while allowing for shunt current along the electrodes.

$$\sum_i \int_{\gamma_l} \chi_i \sigma \nabla \phi . \mathbf{n} = I_l$$

There is one equation for each electrode so all the variables are completely

determined. The resulting system is:

$$\begin{pmatrix} K & A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} \phi \\ V_l \end{pmatrix} = \begin{pmatrix} 0 \\ I_l \end{pmatrix} \qquad (4.10b)$$

where K is the system stiffness matrix defined by the left hand side of Equation 4.10a. The matrix A contains the coefficients on the right hand side of Equation 4.10a. The columns of A are the same as the rows defined by Equation 4.10b so the entire system is symmetric.

The system of Equations 4.10b was solved for a mesh of 761 nodes composed of Turner Triangles. The system modelled a homogeneous disk of radius 15cm driven by thirty two electrodes covering 30% of the boundary. Figure 4.10 displays the boundary current density and voltage during the application of the trigonometric current patterns $I_i=\cos(\theta_i)$. The voltages predicted by this model were compared with the predictions of the Boundary Fourier, see Figure 3.6a. The electrode voltages predicted by the two methods agreed to 1%. It was concluded from Equation 4.7 that of the order of ten times as many nodes would be required for a finite element model of linear triangles to predict the voltages on current-carrying electrodes to 0.1%, even given the unlikely event that the contact impedance is uniform and known. These assumptions are even less likely in clinical situations where the contact impedance is known to be large and vary rapidly in both space and time due to variation in skin condition and wetness, [85].

4.11 Finite Element Modelling of Phantoms

On the basis of these experiments the "hybrid" measurement scheme described in Section 3.7.1 was proposed. It was wished to retain the advantages derived from the use of optimal current patterns and also to avoid measuring voltages on current carrying electrodes. A sixty four electrode system was proposed where current was applied to the region through thirty two electrodes and voltages were measured on thirty two separate electrodes interleaved with the current driving electrodes. Once the function of the two types of electrodes has been separated these electrodes can be customized to their different roles.

Using the Gap Model, Gisser *et al*, [32], have shown that as the size of current driving electrodes is increased the measured signal increases as the square root of the proportion of the boundary covered by electrode. This result was confirmed by Paulson *et al*, [70], for the discrete case with contact impedance.

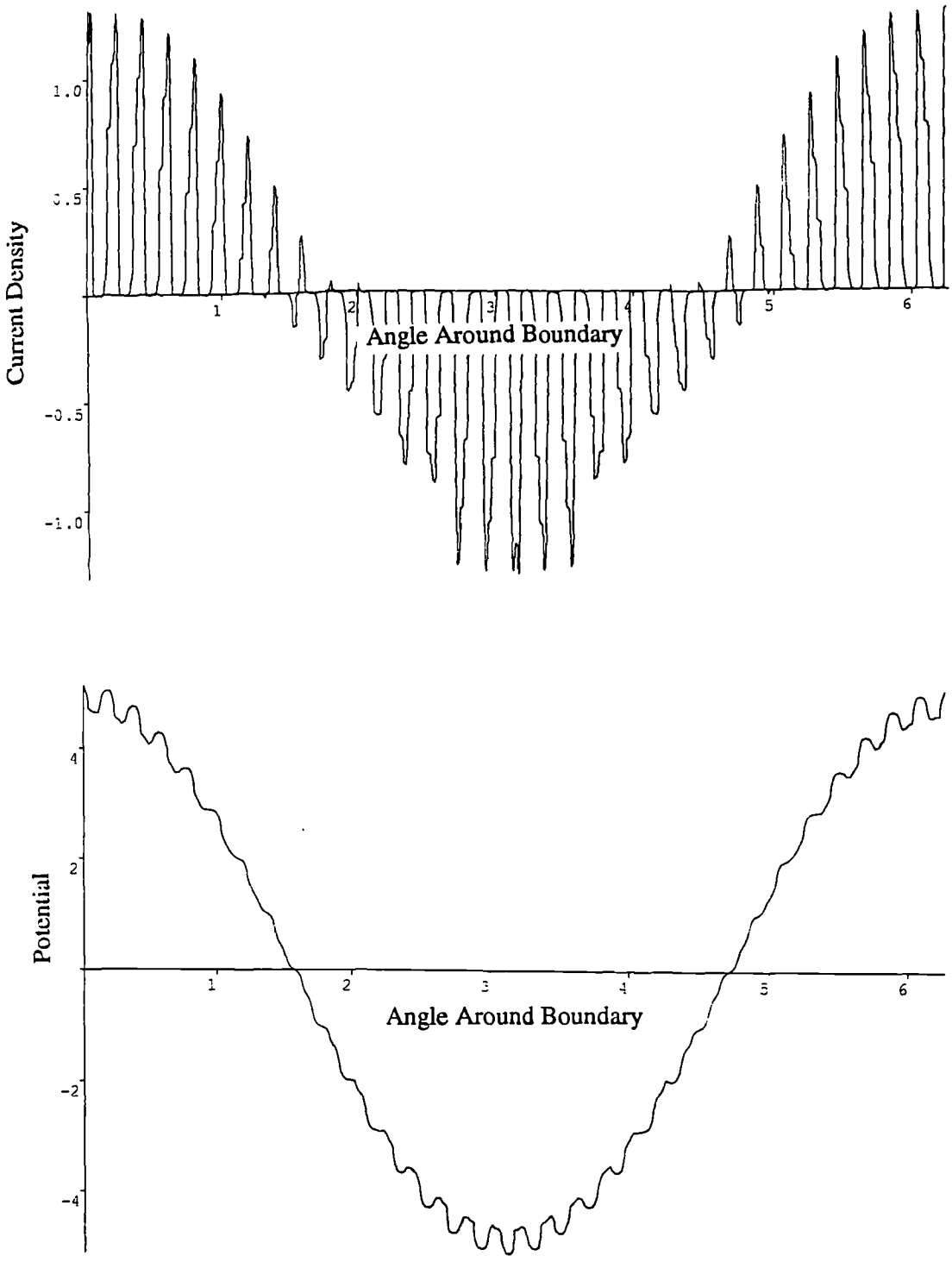


Figure 4.10 *The boundary current density and voltage on a disk of uniform conductivity and radius 15cm, driven by thirty two symmetrically placed electrodes covering 50% of the surface as predicted by the Finite Element Method.*

For this reason, groups applying optimal currents have often used very large electrodes covering up to 98% of the boundary. However, it is extremely difficult to model large electrodes adequately due to the complex fields they induce. This is particularly the case where the edges of electrodes are very close, in which case the shunt current increases dramatically. Large electrodes also cause problems in the construction of phantoms. The electrodes need to be placed very precisely due to the narrow gaps between them. Misplacement of an electrode or the growth of a conduction path bridging the gap can lead to significant errors in voltage measurements.

The opposite constraint is placed on the voltage measuring electrodes. They need carry only the miniscule current necessary to measure the voltage; approximately 10^{-9} amps. To ease the modelling problem we wish to limit the effects of the voltage measuring electrodes on the current and potential fields. As even passive electrodes shunt current this requires them to be as small as possible. There is no reason for them to be larger than needles. Very small electrodes suffer from large thermal noise problems but it was found by experiment that electrodes with a diameter of 0.5mm are small enough to neglect in a finite element model but large enough not to suffer from this problem.

Placing the passive voltage measuring electrodes half-way between the large current driving electrodes limits the effects of contact impedance on the measured voltages. As the voltage measuring electrodes are virtually passive their contact impedance does not effect the voltage they measure. The tomograph electronics delivers the same current to an electrode no matter what its contact impedance. Contact impedance effects the voltage measurements only in the effect it has on the current density distribution under the drive electrodes. Voltage measurements half-way between electrodes are the most insensitive to this distribution. This was tested using the Boundary Fourier Method by comparing voltage measurements made half-way between electrodes on two homogeneous disks. One was driven by electrodes covering 30% of the boundary and the other by electrodes covering 10% of the boundary. The difference in the boundary current density patterns for these two systems is much larger than would be expected due to variation in contact impedance alone. The difference in the voltage measurements between these two systems was 0.25% for the first optimal current pattern and showed a maximum variation of 0.6% for the last optimal current pattern.

This system can be simulated using a standard finite element model. The insensitivity of the measurements to current density distributions under electrodes

means that coarse meshes can adequately model the behaviour of configurations of small electrodes. Larger electrodes inject current current closer to the point of voltage measurement and so are difficult to model. By comparing the predictions of the Boundary Fourier model with those of the finite element method using a reasonable number of nodes, it was found that electrodes covering 30% of the boundary gave the best agreement. The voltage measurements predicted by the Boundary Fourier Model for a homogeneous disk driven by electrodes covering 30% of the boundary and a finite element model of 761 nodes were compared. The predictions agreed to better than 0.1% for the first sixteen trigonometric currents.

4.12 Finite Element Modelling in Three Dimensions

4.12.1 Three Dimensional Elements

To model three dimensional volumes a mesh composed of three dimensional elements is required. At the present stage of EIT development the volume to be imaged is a cylinder with electrodes attached to the curved surface. One way of constructing a mesh filling a three dimensional disk is to translate a two dimensional, circular mesh in the direction perpendicular to the plane and extend the triangles into triangular prisms, known as wedges. Six noded wedges have interpolation functions that have a subset of the terms in a quadratic polynomial in three variables. As the interpolation functions are not fully quadratic the 6 noded wedge counts as a linear element in terms of its speed of convergence, k=1 in Equation 4.7. To make the three dimensional mesh an analogue of the two dimensional one we would prefer it to be composed of four noded tetrahedra with the full linear interpolation function, N=a+bx+cy+dz. As the S(ij,k) data structure has a size proportional to the number of nodes in an element cubed the tetrahedral elements represent a considerable saving in terms of memory requirements without compromising convergence speed. A mesh of tetraheda can be achieved by decomposing each wedge into three tetrahedra as shown in Figure 4.12. The cylindrical mesh can be constructed by stacking these disks of tetraheda on top of each other.

4.12.2 Three Dimensional Mesh Generation

The program RMESH3D has been written to construct three dimensional cylindrical meshes of tetrahedra. The electrodes lie on circles at levels up the sides of the cylinder. A typical configuration would consist of four levels with sixteen equally spaced electrodes at each level. Part of the input to the program is the number of electrodes on each level and the number of levels that make up the cylinder.

Associated with each electrode is a rectangular area on the curved surface of the cylinder called a *cell.* The input to the program includes the relative position of all the surface nodes in a cell including identification as to which form part of an electrode.

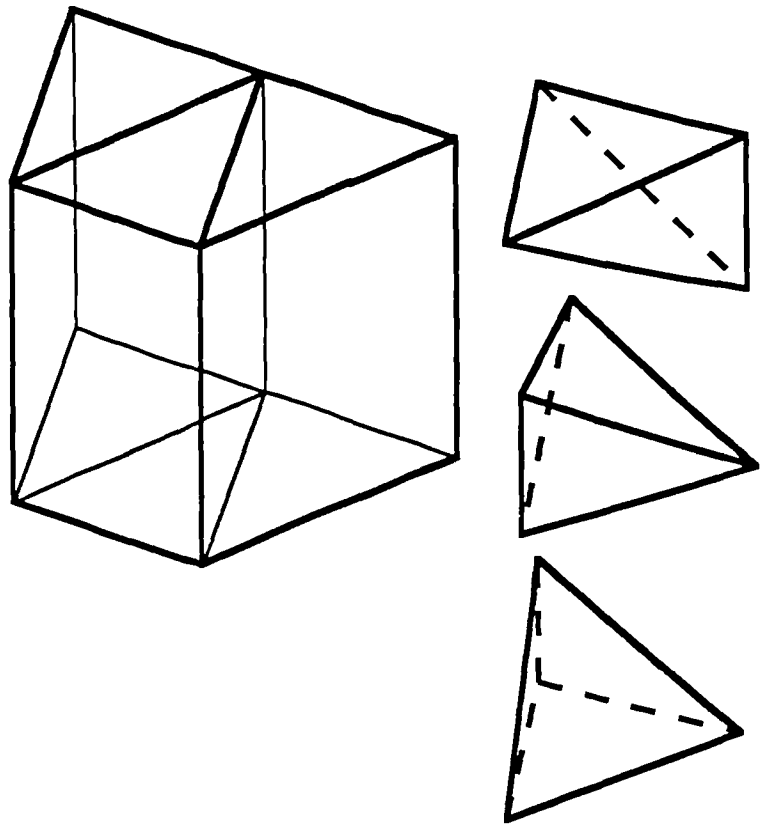


Figure 4.12 *Part of a three dimensional mesh illustrating how a space filled with wedges can be cut into tetrahedrons.*

RMESH3D constructs a two dimensional mesh with one dimensional cells of surface nodes, as described in Section 4.9, and then extends this mesh into a three dimensional disk of tetrahedra. These disks are stacked on top of each other to build a cylindrical mesh with surface nodes at the relative cell locations in the other dimension. As with RMESH, the nodal numbering is chosen to minimise the computational effort involved in factorizing the system stiffness matrix, see Section 4.9. RMESH3D produces a mesh file with the same format and information as RMESH. The suite of finite element programs developed during the course of this project will operate on a mesh file produced by either program. Useful three dimensional meshes tend to be very large and often consist of tens of thousands of nodes.

4.13 Conclusions

In this Chapter numerical models have been developed which agree with analytic and semi-analytic calculations to sufficient accuracy to be used for two dimensional impedance image reconstruction. Similar models for three dimensional imaging have been developed. A novel data structure has been described which greatly increases the speed at which the matrices required by these models can be calculated.

Chapter 5

Parallel Computing

5.1 Introduction

A taxonomy of computers has been described by Flynn [27]. This scheme classifies computers by the number of instruction streams executed concurrently and the number of sets of data executed upon concurrently. At any time the classical Von Neumann machine is executing a single instruction operating on a single set of data. It is classified as a Single Instruction Single Data machine, SISD. Supercomputers are typically vector machines capable of performing operations on vectors of data with single instructions. These operations are typically the level one Basic Linear Algebra Subprograms, BLAS, such as the Linpack GAXPY (Generalised $D_i=aX_i+Y_i$, i=1,N), [22]. The hardware of these machines will often allow parts of this vector calculation to execute concurrently. This is achieved by allocating physically different pieces of hardware to different parts of the operation that can execute concurrently. These computers are described as Single Instruction Multiple Data, SIMD. Fully parallel computers have many instruction streams executing on different, possibly vector, sets of data on independent processors. These Multiple Instruction Multiple Data, MIMD, computers have greatly increased the speed of high performance computing. The MIMD class of computers can be decomposed into two sub-classes depending on whether the data storage is shared by the processors or distributed with them. By offering potentially unlimited scalability of computing power and eliminating contention for data access, distributed memory MIMD computers appear to be the key for high performance computing in the near future. This Chapter explores the issues involved in the choices of parallel hardware and parallel software.

5.2 Computing for EIT

One of the aims of research into EIT is to produce a medical imaging tool for monitoring or imaging patients in hospitals. This puts severe restrictions on the computing hardware that is appropriate. The amount of computation required to reconstruct an impedance image can be of the order of 10^6 floating point operations for a two dimensional image and 10^{10} operations for three dimensional tomography. An image should be calculated within a minute so that medical staff can wait for the image to be produced. If an image can be calculated in a second then EIT can image changes in the chest during an inhalation-exhalation cycle. Real time cardiac imaging

will require images to be produced in a tenth of a second to monitor changes in the heart during a single beat. Faster imaging makes EIT applicable to a greater range of problems. The computation rates needed to calculate an image in less than a second are those associated with present day super-computers. These machines are very large, expensive and completely inappropriate for a hospital environment. The desired computer hardware for EIT has very high, scalable performance and is both compact and cheap. In the last few years a range of parallel computing elements have become available which fit these requirements.

5.3 Parallel Computers

In this thesis only MIMD machines with distributed memory will be considered. The components of a distributed memory MIMD machine are known as *processors* and function like small computers. Each processor has a computation unit connected to its own local memory and a number of *channels* to other processors along which data can be exchanged. The computation unit on a processor runs its own programs, known as *processes*, which can access the memory local to that processor. To use data stored in memory local to another processor, a process on that processor must access the data and pass it along a channel connecting the two. If two processors are not directly connected the data will need to be passed between processes executing on the processors along a connected path between the processor whose local memory contains the data and the processor requiring the data.

A computer of this type has the potential to perform calculations very quickly. Each processor is connected by its own bus to its own memory so there is no competition between processors for buses or memory resources. The speed at which data can be delivered to the computation unit can be tuned to its requirements so that no operation is delayed waiting for operands. As each processor can perform calculations concurrently the theoretical computing power of a machine with P processors is P times the computing power of a single processor. It is this linearly scalable performance that has increased the peak calculation rate advertised for new computers from Megaflops (Mflops) to Gigaflops and, recently, Teraflops.

In the last few years several parallel processors have become available. The first and most well known is the Inmos Transputer family of parallel processors. These processors are very cheap, designed to be connected into parallel computing machines and each processor has a computing capability of 1 to 4 Mflops. Intel produces the i860 chip as a component in a parallel computer. These chips are considerably more expensive than Transputers but have a peak computation rate of 60 Mflops in double precision arithmetic. As the i860 has no communications channels

they are commonly partnered by Transputers which control the inter-processor communications. In 1992 Texas Instruments released the TMS 320C40, capable of peak calculation rates of 50 Mflops and able to handle its own communications.

5.4 The Transputer

The Inmos Transputer family of processor chips were designed to be components in concurrent programming systems. Each Transputer is a VLSI device with processor, memory and communications links (channels by another name) for direct communications with other Transputers. Parallel computers can be constructed from a collection of Transputers operating concurrently and communicating through links.

The Inmos T800 is the chip designed for numerically intensive applications. It consists of a CPU including a hardware scheduler, a floating point unit, four communications links, 4 Kbytes of on chip RAM and an external memory interface; all packaged onto a chip 26 mm square. On the chip with 20MHz clock rate the floating point unit is capable of sustaining 2.25 Mflops on 64 bit, double precision numbers concurrent with the operation of the CPU. Each of the four links can transfer data bi-directionally at up to 2.35 Mbytes per second. The external memory interface can directly access a 32 bit wide, linear, address space of 4 Gbytes and transfers information at a rate of 4 bytes every 100 nanoseconds corresponding to 40 Mbytes/sec, [45].

Typically, Transputers are purchased on a printed circuit board called a TRAM (an abbreviation of TRAnsputer Module) A TRAM will include at least one transputer together with its own local memory, generally in the range 1 to 16 Mbytes, and possibly extra hardware to control communications with the host computer. These TRAMS can either plug directly onto a PC expansion slot or, more commonly, onto a *motherboard* that acts as an interface between the transputer network and the host computer. As only one Transputer need be connected to the host only one set of host interaction hardware is required and this is usually included on the motherboard. TRAMs designed to plug on to a motherboard may therefore be more simple. Inmos have set a standard, public domain, interface to TRAMs from an early stage allowing third party production of TRAMs and other transputer based application boards. This has resulted in fierce competition between TRAM manufacturers and corresponding low prices. The TRAM market is highly volatile with prices varying enormously between companies and over short periods of time. Memory fast enough to effectively service the 35 MHz transputers is expensive. A large proportion of the price of a TRAM is invested in the fast memory and so TRAM prices are linked to the

world RAM market.

The hardware purchased for this EIT project is a TMB04 motherboard with a 20 MHz T800 Transputer and 4 Mbytes of DRAM, [83]. Three TTM17 expansion boards attach to the motherboard each with a T800 Transputer and 4 Mbytes of DRAM. This system cost £1850 in 1990. The transputer system occupies a single expansion slot in a Zenith XT clone.

At the time of writing Inmos had postponed the release of the next generation of Transputer, the T9000, until the fourth quarter of 1992. It is expected to achieve a peak performance of 25 Mflops and maintain a link communications rate of 100 Mbits per second. Apart from increased performance the major innovation expected in the T9000 is hardware control of the routing of communications through a network of Transputers. Typically, messages passed between Transputers not directly connected in a network must be routed through intermediate Transputers. On each of these Transputers a process must run to receive messages and re-transmit them towards their destination Transputer. The implementation of the routing processes is often a major undertaking and their operation in parallel with application processes degrades performance. It is hoped that automatic through-routing of messages under hardware control will greatly ease the use and increase the performance of the T9000. It is known that Inmos aims to introduce *virtual links*. A Transputer cannot execute a set of processes which require more than four links to the rest of the network. The number of links available places restrictions on how processes are allocated to processors in a transputer network. This restriction is commonly circumvented by the use of multiplexer processes that communicate with several processes on a transputer and pass all their inter-processor communications along the available links. With the T9000 it will be unnecessary to implement this in software and any application will map onto any connected, Transputer network with any processes on any processor.

5.5 Models for Parallel Computing

The model primarily used for the design of parallel algorithms is that of Asynchronous Communicating Sequential Processes, (ACSP). This models a parallel program as a collection of processes that communicate with each other along communication channels. Each process may itself be a collection of inter-communicating processes, see Figure 5.5a. At the lowest level a process will be a sequential program that could be written in any standard programming language with extensions to implement communications.

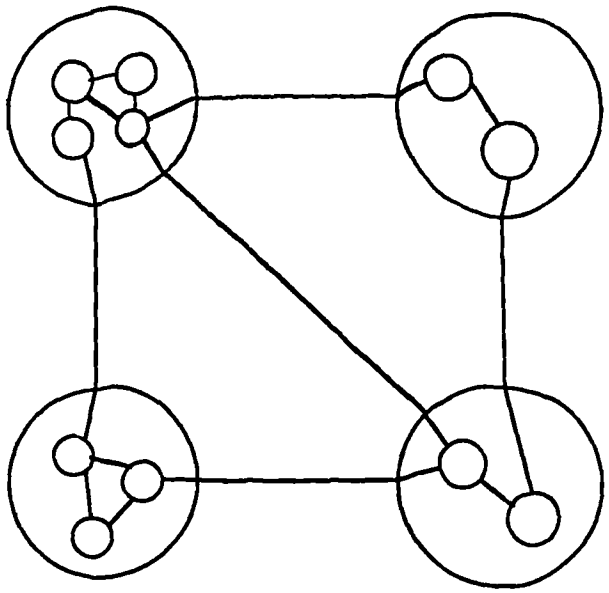


Figure 5.5a *An Occam model of a parallel application which consists of four inter-communicating processes. Each process is indicated by a circle with the communication channels linking them. A process may be composed of a number of sub-processes linked with their own communication channels.*

There are several protocols controlling the passing of messages between processes. The Occam model assumes a *double blocking* protocol, [2], which prevents either process involved in a communication from continuing until the communication is complete. Thus a process wishing to send a message must wait until the destination process is ready to receive it and vice-versa. The double blocking protocol allows processes to be synchronised. This is an advantage as it allows the determination of the order that events will occur in different processes.

The Transputer and the concurrent programming language Occam were developed together to be the hardware and software implementations of the Occam model of concurrent computation. The Transputer's hardware scheduler allows many processes to appear to run concurrently on a single Transputer by quickly and efficiently sharing the available processing time and resources between them. This allows an entire application of many concurrent, asynchronous processes to appear to run in parallel on a single Transputer. Furthermore, a network of transputers can be constructed by connecting them together via their links. A link is a hardware implementation of the Occam model's communication channels. Thus an application based on the Occam model may be configured to run on a network of Transputers as long as the number of channels between processes on different processors is less than

the number of hardware links between them. This ability means that some degree of portability is introduced into parallel applications from the earliest design stage. To the programmer of a process, inter-processor and inter-process communication is identical. No modification to the application source files is needed to reconfigure an application to run on a different network of transputers.

5.6 Parallel Software

5.6.1 Languages For Parallel Programming

At the beginning of this PhD project several parallel languages were commercially available but the most commonly used was Occam. This language was developed specifically for concurrent programming environments, in particular the Transputer. Parallelism is inherent in Occam. Every program statement is specifically labelled to be executed sequentially or concurrently. Occam is still the most natural parallel programming language and compiles into the fastest code to run on Transputers. Unfortunately, Occam for the Transputer was integrated into a folding editor and set of development tools called the Transputer Development System (TDS). TDS has a number of inadequacies which have limited its application. Standard operations such as input/output to files are poorly implemented. For example, Occam does not use named files but accesses areas that are a specified number of "folds" away in the folding directory system. This would be a major handicap when writing a flexible reconstruction package.

The alternatives to Occam were standard sequential languages, such as Fortran and C, with extra intrinsic functions to enable inter-process communications. These were widely available from many software producers in the UK and the Continent.

5.6.2 Constructing a Parallel Application

A parallel application is constructed by designing and writing a set of inter-communicating processes in any suitable language. Both TDS and systems based on sequential languages require further information to map the suite of concurrently executing processes onto the processor network. This is generally achieved by a user written *configuration file* which contains a description of the processor network; the processor types and the links between them, a list of all the processes and allocation of each process to a specific processor. All the channels connecting processes need to be declared in the configuration file. An executable file is then built by combining

these with the information in the configuration file. During configuration the application is checked to see if it can be physically mapped onto the network of processors as described in the configuration file. The executable file, called the *application file*, must load all the processes onto the correct processor by *worming* its way through the network booting one processor at a time, and initiating all the processes simultaneously.

5.6.3 Error Handling on Transputer Networks

Transputer networks are marketed as add-on components to a host computer and all interaction with them is made via the host's operating system. The Transputer network has no operating system or environment of its own. All interaction with the Transputer network is made via user-written processes executing on the single Transputer connected to the host system. As there is no monitoring or error reporting whatsoever the transputer network is a naked and extremely hostile environment for the software developer. Developing and debugging in this environment is a nightmare.

Parallel programs can have many errors that do not exist for sequential programs. These arise from problems with communications between processes. Two processes must cooperate when a message is to be passed between them. If this cooperation does not occur, one or both processes will be left permanently waiting for communication to take place. The situation, called *deadlock*, can develop when two processes are waiting to communicate with each other. Errors which are far more difficult to correct occur when messages get mixed up. As processes are asynchronous, the order in which communications arrive at a process can be indeterminate and can vary between executions of an application. These communication *races* can lead to intermittent errors which are extremely difficult to debug. There is no way that a compiler can detect the circumstances in which this may arise without running the application. These communication errors and all the errors that sequential programming is prone to, occur in the development of parallel programs. No error report will be generated and the most likely result of any of these errors will be a hung system.

Any error trapping and reporting on the Transputer network must be implemented by the user. Creating such a system is a difficult task on its own which relies heavily on network wide communications as error messages often need to be transmitted via intermediate processors to reach the host computer. Any problems on the intermediate processors will prevent the user initiated error messages from reaching the host. Errors such as over-writing the end of an array will often hang a

Transputer before a user-initiated error message can be transmitted. This is particularly true when receiving arrays of data from other processes. Worst of all, user diagnostic messages will often interact with the algorithm's communications to cause new problems.

5.6.4 Development Tools on Parallel Computers

Interactive debuggers are still a recent innovation in the programming of Transputer networks. Tbug, [81], is a debugger produced by 3L Ltd. to be used in conjunction with their range of languages with parallel extensions. It allows the user to single step through independently executing processes and to set break points. As with all such debuggers, it has the disadvantage that applications run differently while under the control of Tbug due to delays introduced by its presence. This is especially the case with TBUG which reconfigures the application so that all the processes execute on a single Transputer. The changes forced on the application to make this possible are likely to introduce errors of their own.

Higher level development tools for parallel programs were, and mostly still are, unavailable. Within the last two years there have been several developments which improve the interface between the network and the user. A distributed operating system known as Helios has become available. This fully distributed, Unix like operating system is, unfortunately, very expensive and so has had little impact on the users of small networks of processors. Atari developed a transputer based workstation, known as the ATW, based on networks of T800's and Helios, but went out of business within six months of its release. Other systems, marketed as environments rather than operating systems, have also become available. CStools extends SunOS, the operating system produced by Sun Microsystems Inc., to transputer networks hosted on Sun workstations. CStools provides communications processes for the user and monitors application processes for errors. It also allows direct communications between the host and Transputers in the network. It allows interactive debuggers to be used on any processes on any processors. A similar system called PARIX has been developed by Parsytec for an European Community project.

5.6.5 Parallel Software for EIT

At the beginning of this PhD project a large suite of sequential software was available. This included a reconstruction program based on the NAG Finite Element Library written by Breckon, [9]. All this software was written in Fortran 77 and resident on a Sun workstation. Considerable effort would have been necessary to

rewrite this software in Occam and it was therefore decided that the benefits incurred would not warrant the time involved. The other option was to port the software with as little modification as possible to the transputer environment. For this reason it was decided to build a parallel reconstruction program in parallel Fortran. A Fortran with parallel extensions produced by 3L Ltd. was chosen, [80]. Towards the end of this PhD project, when it became available, Tbug was purchased to speed up development. It was found to be useful while developing small portions of program but it could not cope with the full reconstruction algorithm as this was far too large to fit on a single Transputer.

Porting a sequential program to a parallel computer is a formidable task. This is particularly the case where an algorithm requires many, varied, complex operations. Enormous changes were introduced to the program to rewrite it as a collection of processes capable of concurrent execution. The task is more difficult than replacing calls to sequential subroutines with calls to parallel ones. As the data used throughout reconstruction needs to be distributed, the parallel subroutines must be designed to use consistent data structures. Besides the main suite of reconstruction processes considerable amounts of programming is required to implement a general, network wide, communications system. Some method to trap and report errors through the communications system must also be designed and implemented. Many other changes in working practices were introduced by the move from Unix to MSDOS. A set of batch files needed to be created to compile, link, and configure parallel applications.

5.7 Parallel Algorithms

5.7.1 When Is a Parallel Program Appropriate?

The parallel programmer faces many more choices than the programmer of a sequential computer wishing to perform the same calculation. S/he must decide upon the algorithm to use to achieve the desired result. The algorithm must be formulated in an Occam model and then the Occam processes must be implemented in an appropriate programming language. For numerical analysis problems the goal of all these design stages is to produce a computer solution to a problem that runs quickly. The extra design effort required to produce a parallel application would be wasted if speed of calculation was not of the essence. For some industrial control applications where the data acquisition systems are physically distributed around a factory, a parallel program executing on a distributed, parallel computer may be the most practical solution. For EIT the reason for substituting a parallel computer for a

sequential one is only to increase the speed of execution. In practically all cases the implementation of a numerical analysis application on a parallel computer is considerably more effort that on a sequential computer. This large, and often massive, investment of human resource is only worthwhile if speed of solution is the primary goal of an implementation project.

5.7.2 Choosing a Sequential Algorithm

When choosing an algorithm to achieve a result on a sequential computer there are generally two major considerations, the number of floating point operations necessary and the amount of data storage required. There is often a trade off between these features; allowing the use of more data storage will often yield a result in fewer operations. However, the choice of algorithm is generally straightforward. The maximum storage requirement is a pre-determined constraint, often set by the hardware, and the algorithm that requires the least number of operations to achieve the result with a storage requirement less than the maximum is chosen. The time required to perform the calculation is proportional to the number of operations. For this reason sequential computers are characterised by their benchmark performance on typical applications. Several standard benchmarks have been designed to simulate common mixes of integer and floating point, scalar and vector operations, [50]. They have been in existence for a decade or more and are commonly quoted as a measure of a computers performance. Often this is all the information necessary to chose an optimal sequential algorithm to calculate a desired result.

5.7.3 Modelling a Parallel Application

It is more difficult to use a parallel computer to its full potential. If used optimally a network of P processors could potentially complete a task P times faster than a single processor. In practice this is never achieved. For a processor to take part in a calculation it must have data communicated to it and it must communicate a result back. These communications introduce an overhead that does not exist when using sequential computers. The larger a network of processors the larger this overhead is. For communications purposes, the size of a network of P processors is often measured by its *diameter*, D(P). The distance between two processors is the minimum number of links required to pass a message between them. If a message needs to be routed along at least three links, through two intermediate processors, to be transmitted between processor A and processor B then they are said to be 3 units apart. The diameter of a network is the maximum distance between any two processors in the network. For all the processors in a network to contribute to a calculation, some data needs to be transmitted at least the diameter of the network and

results transmitted back again. For an application of some fixed size, the time required to perform the calculation, T(P), on a network of P processors is the sum of the communications time, COM , and the computation time, CAL of the longest running process. This is generally the process connected to the host computer as it is involved with the transmission of data to the network and passing the results back.

$$T(P) = COM(D(P)) + CAL(P) \qquad (5.7a)$$

The communications time is an increasing function of the diameter of the network. A network in which every processor is connected to every other processor, known as *completely connected*, has a diameter of unity independent of the number of processors in the network. Completely connected systems are technically difficult to build so existing systems are based on compromise networks. Large networks with the topology of hypercubes, see Figure 5.8, have been built both in Europe, for example at the University of Liverpool, and the US. A hypercube network has a diameter that increases as the logarithm of the number of processors; $D(P) = \log_2(P)$.

The computation time can be bounded above by the time required for a single processor to complete the calculation. At best the computation time decreases as the inverse of the number of processors: $CAL \propto 1/P$. This assumes that the problem can be distributed in such a way that all the processors have useful calculations to perform all the time and at no stage does one processor need to wait to be communicated an operand. It also assumes that no computational overhead is introduced in distributing the application. Attempting to achieve this goal is known as *load balancing*. Optimal load balancing is unachievable for networks with more than one processor as no processor can begin execution until it has had data supplied to it..

Assuming the communication time is proportional to the hypercube diameter, $COM(P)=K_1 \log_2(P)$, and the calculation time is inversely proportional to the number of prossessors, $CAL(P)=K_2/P$, an expression for the run time of a particular application is:

$$T(P) = K_1 \log_2(P) + K_2/P \qquad (5.7b)$$

where K_1 and K_2 are constants. This expression has a minimum at $P= ln(2)\ K_2/K_1$. For any given application there is an optimal number of processors. If more processors than this optimal number are used the benefit gained from sharing the calculation with the extra processors is more than compensated by the increase in communication time. The optimal number of processors increases as the amount of calculation increases or the time required for a communication decreases. If

communications are made faster, K_1 is decreased, and the optimal number of processors increases. Similarly, if the rate that calculations can be performed is increased, K_2 is decreased, and the optimal number of processors is decreased. Thus the ratio, K_2/K_1, is an important design consideration when algorithms are being chosen for a given parallel computer.

5.7.4 Measuring the Efficiency of a Parallel Application

Two measures of the effectiveness of parallel implementations are in common use. The *speedup*, S(P), and *efficiency*, E(P), of an implementation is defined as:

$$S(P) = \frac{T(1)}{T(P)} \qquad E(P) = \frac{T(1)}{P \times T(P)} = \frac{S(P)}{P}.$$

If the communications time is negligible and the application is perfectly load balanced then the optimal speedup is P and the optimal efficiency is 1. More realistically we could aim to keep efficiency bounded away from 0 for increasing P, [5]. There are many different interpretations of S(P) and E(P) depending upon the definition of T(1). A common definition sets T(1) to be the run time of a sequential version of the algorithm on a single processor. This definition leads to a large drop in efficiency as the number of processors is increased from one to two. The single processor version can be written as a single process and needs no processes to implement the data communications. This eliminates the overhead of simulating concurrency on a single processor. It can be argued that this is not a fair comparison of the performance of sequential and parallel machines. The optimal algorithm on a parallel computer is likely to be a function of the number and mixture of processors and the configuration of the network. In particular, the optimal algorithm on a single, sequential processor is almost certainly different from the optimal parallel algorithm. If T(1) is defined as the run time of the optimal sequential algorithm, then the speedup and efficiency are relative to the best achievable on a single processor. This definition makes explicit the benefit of moving to a parallel computer. Another definition of T(1) is the run time of the P processor algorithm configured to run on a single processor simulating concurrent processing. With this definition, the single processor performs exactly the same calculations as the distributed version and consequently the speedup and efficiency reveal multi-processor and communication effects but nothing about the relative merits of the particular algorithm chosen. In this thesis the first definition of T(1) is used.

Given any of these definitions an observation known as Amdahl's law, [1], holds. Suppose that a program consists of two sections, one part that is inherently

sequential and another that can execute fully in parallel. If the inherently sequential section consumes a fraction f of the total computation, then the speedup is limited by:

$$S(P) = \frac{1}{f + \frac{1-f}{P}} \leq \frac{1}{f} \qquad \forall P.$$

As the number of processors is increased, the time required for the parallel part of the task decreases to zero leaving the inherently sequential part. One counter argument to Amdahl's law is that as the size of problems is increased, the proportion that is inherently sequential typically decreases so any speedup is achievable if you can find a problem large enough.

5.7.5 Choosing a Parallel Algorithm

The parallel programmer faces the task of finding the algorithm and configuration of network which minimizes T(P). Generally the programmer will already have a fixed number of processors but will often have freedom to connect them within constraints. A Transputer network may be configured to have any pairs of links on different Transputers wired together, as long as there is one connection to the host computer. Reconfiguration generally requires the physical connection of wires to the transputer boards and so it is impossible to reconfigure the network part way through the execution of a program. Minimization of T(P) over all possible choices of algorithms and configurations is a problem so formidable it has not even been attempted rigorously. Typically, an application involves several stages where each stage involves a choice of algorithm and each could have an different optimal configuration. As it is not practical to reconfigure before each stage a common problem is minimising T(P) given the configuration as a constraint.

The four transputer system purchased as a component of the impedance tomograph is the largest that can be configured as a completely connected network while leaving one link to connect to the host. However, if the implementation of the reconstruction algorithm assumed a completely connected network it would be impossible to port it to a system with more than four transputers. This would put an absolute upper limit of 4 on the achievable speedup. A gain in speed of a factor of 4 is just not worth the effort of moving to a parallel computer. The reconstruction algorithm must be designed to take advantage of any number, up to the optimal number, of processors. A topology needs to be used that can be generalised to much larger numbers of Transputers.

5.7.6 An Example of Parallel Algorithm Design

An understanding of the problem of parallel algorithm design can be gained by looking at a specific problem. For example, forward substitution is a method for the solution of the matrix equation $L\mathbf{x}=\mathbf{b}$ where $L \in \mathbf{R}^{NxN}$ is a lower triangular matrix, **b** is a known vector and x is a vector of unknowns to be calculated. In pseudo-code the algorithm to solve this problem can be expressed:

```
FOR i = 1 TO N
        x_i = (b_i - Σ_{j=1}^{i-1} L_ij x_j)/L_ii
```

Forward substitution can be programed in Fortran in six lines. It is a task so trivial that it is easier to write the program than to look up the appropriate routine in a standard package. To the parallel programmer it presents an immediate difficulty. The calculations that form the algorithm need to be split into parts that can be calculated independently on different processors. Yet the algorithm as stated shows that x_i depends upon all the x_j's such that $1 \leq j < i$. In other words x_i cannot be calculated before x_1, x_2 ... x_{i-1} have been calculated. At first glance, forward substitution appears to be an inherently sequential algorithm. However it may be rewritten:

```
FOR i = 1 , N
        x_i = b_i/L_ii
        FOR j = i+1 TO N
                b_j = b_j - L_ji x_i
```

Written in this form the parallelism is clear. Each iteration of the inside loop can be calculated independently and in parallel once x_i has been calculated. Once the parallelism has been identified it is necessary to write the algorithm as a set of communicating processes which minimise the necessity of inter-process communications. Each process needs to store some data, in this case elements of the matrix L, and perform some calculations. Examination of the algorithm shows that the data required to calculate x_i is the i'th row of the matrix L, b_i and all the previous x_j: $1 \leq j < i$. A reasonable distribution of the data and calculations required for forward substitution would place row i of the matrix L and the vector **b** on process i, Q_i. This algorithm requires N processes, $Q_i : 1 \leq i \leq N$, which can be written:

{Process Q_i}

FOR j = 1 TO i–1
    RECEIVE(x_j)
    $b_i = b_i - L_{ij} x_j$

$x_i = b_i / L_{ii}$

FOR j = i+1 TO N
    SEND(x_i) TO Q_j

Each process is aware of its unique process identification number, i. The description has introduced two functions, SEND and RECEIVE to implement transmission of information from one process to another. Each data transmission is point to point in that messages are passed from a single source process to a single destination process. Every process that has data sent to it must have a corresponding RECEIVE or the algorithm will fail. This design splits the calculation and data between N processes which must be able to communicate with every other process. Its Occam model representation would be a completely connected set of N processes. Process i would perform 2i–1 floating point operations and N–1 communications.

An application written to this design could give different results on each execution. As the processes are asynchronous there is no guarantee that the x_j's received by a process will arrive in the correct order. On Transputer networks delays are often introduced by communications with the host which also needs to handle interrupts from its own system. Small differences in the clock rates of different Transputers can lead to variation in the order that events occur through the network. This problem can be solved by transmitting the data pair (j,x_j) and performing the suitable calculation in the receiving processes. This complication will be ignored in the following development.

The design as it stands has two other drawbacks. The processes do very different amounts of work; Q_1 performs one flop and Q_N performs 2N–1 flops. If process Q_i is allocated to a processor P_i then the system would be very badly load balanced as P_1 would spend most of its time doing nothing. In addition, the communications/computation ratio is high and the communications are diverse requiring a communication channel connecting every pair of processors. As designed it could only be configured to run on a single transputer as to put a processes on another transputer would require N–1 hardware links. These problems could be alleviated by merging processes so that a single new process would perform the

calculations presently performed by several processes. This reduces the total number of communications required but would still require a completely connected network to implement. To adapt this algorithm to run on a network of transputers, communications processes would need to be introduced. These processes would route the communications through intermediate processors, where communication between unconnected processes was required, so allowing the method to be used on any connected network.

Consider a solution involving $P \leq N$ processes placed on P processors. If the processes are numbered Q_1, Q_2 up to Q_P where Q_i performs the calculations of the old processes Q_{i+jP}: $1 \leq i+jP \leq N$; then forward substitution can be implemented as:

```
{Process Qi}
    NextRow = i
    FOR j = 1 to N
        IF ( NextRow .EQ. j ) THEN
            NextRow = NextRow + P
            x(j) = b(j)/Ljj
            FOR k=1 TO P (k≠i)
                SEND x(j)  TO Qk
        ELSE
            RECEIVE x(j)
        ENDIF
        FOR k = NextRow TO N STEP P
            b(k) = b(k) – Lkj x(j)
```

This distribution of the rows of the matrix L, similar to the way cards are dealt to the participants of a card game, is known as *round robin*. It ensures that the computational load is balanced between all the processes. At each stage of the calculation, ie for each j, no process does three floating point operations more than any other. The only calculations that occur sequentially are the final divisions by the diagonal elements L_{jj} and the adjustment of NextRow. As is typical with parallel implementations this method is much more complex and design intensive than the sequential program to perform the same calculation. Distributing the work between processors inevitably introduces the overhead of "book keeping" calculations, such as those involving NextRow, that were not necessary in the sequential implementation. The degrading effect of the imperfect load balancing and the unavoidable sequential part of the algorithm becomes less important as the size of the problem, N, increases.

Process Q_i only requires knowledge of rows i+jP of the matrix L. In this

case it is unnecessary and generally undesirable to store the other rows of L. Large data structures, such as L, may be distributed around the network of processors. In this problem it was assumed that the data was already in place within each process when forward substitution was performed. This is a reasonable assumption as lower triangular matrices are usually generated by the factorization of other matrices. This factorization algorithm may have an optimal data distribution that is different from the one deduced for forward substitution. As factorization typically requires of the order $O(N^3)$ operations and forward substitution required $O(N^2)$ operations, it is reasonable to optimize the load balancing of the factorization stage and devise a forward substitution strategy that operates as well as it can with that data distribution. Load balancing, data distribution and communications overhead are three strongly inter-connected issues when designing parallel algorithms.

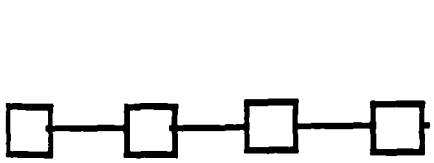
5.7.7 Algorithm Design for EIT

When writing an EIT reconstruction program it is expedient to begin by choosing a network topology. This should be a topology which will approximately minimise the delays introduced by communications in the algorithms which will be executed upon it. It should also be a topology which is possible to generalise to much greater numbers of processors.

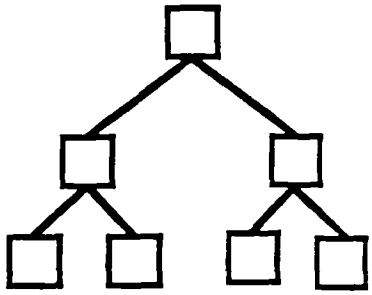
Algorithm design and data distribution are strongly coupled on parallel computers. Much of the data generated by the reconstruction program is used in many of the stages of the reconstruction process. The data structures chosen for these data, such as the potentials generated by the forward modelling stage of reconstruction, determine the algorithms used. Thus the creation of a large program, such as an EIT reconstruction program, must be treated holistically with regard to the largest data structures and those with the largest scope.

5.8 Topologies and Communication Systems

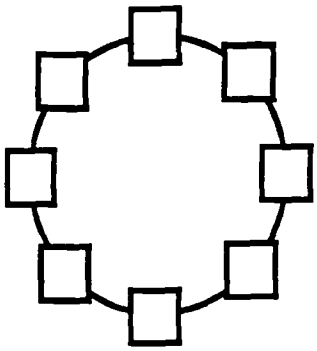
A host of different network topologies are in use. Among the the most common are pipelines, trees, rings, meshes and hypercubes, see figure 5.8. Each topology is suited to different algorithms with different communications requirements. For an algorithm to operate at a high efficiency both the algorithm and the topology need to be matched to the problem



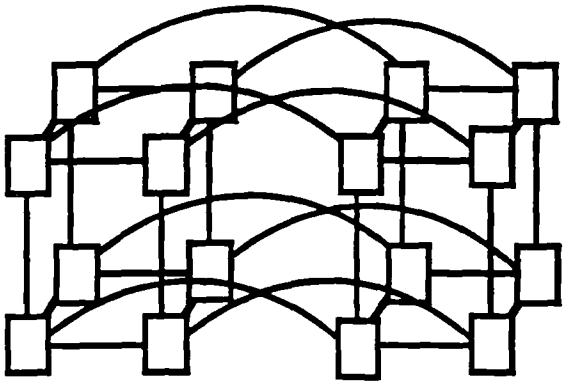
a) Pipeline



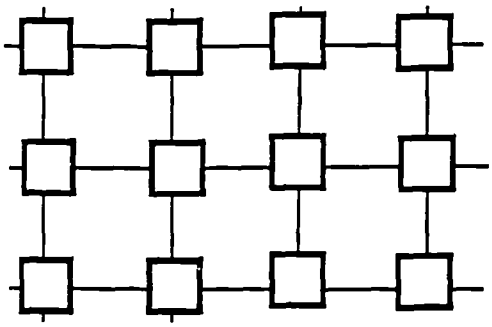
b) Tree



c) Ring



d) Four Dimensional Hypercube



e) Grid

Figure 5.8 *Some popular topologies for networks of concurrent processors.*

5.8.1 Types of Communications

Several classes of communications are commonly used in parallel algorithms. The most basic communication is the point-to-point communication that is the basis of

the Transputers hardware links. Other communications patterns can always be considered as sets of point-to-point communications. Two of the most commonly encountered are known as *scatter* and *gather* corresponding to one-to-many and many-to-one communications respectively. The lines of pseudo-code in the final example of Section 5.7.6 are an example of a scatter:

```
IF (NextRow .EQ. j) THEN
        FOR k=1 TO P (k≠i)
                SEND x(j)  TO Q_k
ELSE
        RECEIVE x(j)
ENDIF
```

A gather would be the same but the functions RECEIVE and SEND would be interchanged. Finally a complete exchange of information held on all processes is accomplished by a *multi-node-scatter-gather* during which all processes communicate, directly or indirectly, with all other processes. The optimal topology for a given algorithm is defined by the mix of these communications protocols.

5.8.2 Message Passing Strategies

Commonly when a parallel application is mapped to a network of computing elements more communications channels are required between processes on different processors than there are hardware links. This problem is commonly solved by the implementation of communications processes which multiplex all the communications required by the processes on the same processor through the available links. The use of these processes has many advantages over direct inter-processes communication. They allow communications between processors not directly connected by routing messages through intermediate processors. This makes applications designed with the Occam model portable between different topologies and allows the design of applications to be independent of the hardware constraints. Finally, separating the calculation and communication into different processes allows the *masking* of communications. The SENDing process can continue to execute useful calculations before the message has reached its destination process, thus masking the delay that would be introduced by double blocking for direct communication.

The routing of messages through a network is often a complex operation. Routing messages along the shortest path commonly leads to "hot spots" where communications bottle-necks delay messages converging along several paths. Random routing, where a message is directed via a random intermediate processor to

its destination, is often used to alleviate this problem. More complex message passing strategies, such as *worm-hole routing*, may be required. This method first allocates all the communication processes along the route connecting source and destination processors to execute the communication. Messages are transferred in small segments chain gang style. This method parallelises the otherwise sequential task of message passing and so messages are transferred faster. The communications processes can be very complex and so commonly one set is available for all applications to use. A trade off exists between routing complexity and speed of transmission as once communications processes become too complex the routing calculations cause more delay than simple message routing.

5.8.3 Communications on Ring Topologies

For this project it was decided to use a very simple ring topology and routing strategy. A ring can be made of any number of transputers and so can be incrementally upgraded at any rate. For comparison a hypercube topology needs the number of processors to double to maintain its symmetry. Although the ring has a large diameter, P/2 compared to $\log_2(P)$ for a hypercube, it can perform a scatter, gather or a multinode-scatter-gather simply and at an optimum rate. A *rotation* can be defined as each processor passing information to the next processor clockwise around the ring and receiving information from the anti-clockwise adjacent processor. One unit of time can be defined as time required to perform a single rotation . A scatter, gather or a multinode-scatter-gather can be performed by P–1 rotations taking P–1 units of time. On any topology these communication protocols require each processor to receive at least P–1 messages. Thus the minimum time possible to complete any of these operations is P–1 units of time. The ring topology can perform these common communications as fast as any other network. Ring topologies are slow for communications between processors placed nearly diagonally on the ring. However if point-to-point communications are only required between near processors they can compete with the more complex topologies. Matrix manipulations with round-robin data distribution, such as the forward substitution described in Section 5.7.6, are rich in nearest neighbour communications, and scatter-gathers. These operations can be performed as fast on a ring as on more complex topologies. The routing strategy is to pass a message around the ring until it reaches the destination processor and then to pass it to the destination process. This simple routing strategy requires minimal calculation and so imposes a relatively small delay between reception and re-transmission of a message.

5.9 Conclusions

In this Chapter the issues in the choices of parallel hardware and software have been investigated. After considering the types of communications that would be expected in the most computationally intense part of a reconstruction algorithm, Transputer networks in the shape of rings have been selected. If communications are between near neighbours or are of the form of scatter-gathers then a ring topology is both simple and rapid. Matrix operations are rich in this form of communications. For small networks of processors the ring topology allows for smooth scalability as more processors are added.

The Occam model of a parallel program has been introduced. A detailed example of the design of a parallel algorithm in terms of inter-communicating processes was given. This example addressed the issues of load balancing and implementation given the constraints of Transputer networks. A number of parallel programming languages were discussed and Parallel Fortran was selected for the implementation of a reconstruction program.

Chapter 6

Parallel Algorithms for EIT

6.1 Introduction

As described in Section 1.4, an iteration of a multi-iteration, Newton type, reconstruction algorithm proceeds in two stages. During the first stage, a numerical model is used to predict the voltage measurements that would be made on a region with a conductivity distribution which is the best estimate, so far, of the conductivity distribution of the region to be imaged. In the reconstruction algorithms described in this thesis the Finite Element method is used for the forward modelling. In the second stage, this best estimate conductivity distribution is adjusted so that the numerical model would better predict the behaviour of the region to be imaged. To achieve this a derivative matrix is calculated. The elements of this matrix are the derivatives of the experimental measurements with respect to the conductivity parameterisation. Using this matrix, a least squares system is solved for the conductivity update.

In this Chapter these two stages of reconstruction are investigated in detail. Different algorithms are compared and parallel implementations are developed. In Sections 6.2 to 6.7 the Finite Element method is investigated. Section 6.3 deals with the construction of the Finite Element system while Sections 6.4 to 6.7 look at a range of algorithms for the solution of this system. In a similar way, Sections 6.9 to 6.10 look at the construction and solution of the least squares, Newton system. Section 6.11 looks at the structure and implementation of a complete parallel, reconstruction program.

6.2 The Finite Element Method in Parallel

In the course of multi-iteration reconstruction algorithms it is necessary to predict the voltages that would be measured on the present *best estimate* of the conductivity distribution. For the system developed at Oxford this is accomplished with a finite element model. Each iteration of the reconstruction algorithm must construct and solve the finite element model for a different conductivity distribution.

The two major stages in the forward modelling segment of a reconstruction algorithm are the calculation of the system stiffness matrix and then the solution of the finite element system, possibly with many right hand sides, to yield the predicted

voltages. The two stages need to be considered together as both involve the system stiffness matrix, K(i,j) which can be very large. For modelling a two dimensional, disk shaped, region, driven by thirty two electrodes, K is typically of the order 1000x1000. A three dimensional, cylindrical region driven by four rings of sixteen electrodes requires a system stiffness matrix of the order 10000x10000. Fortunately not all the matrix needs to be stored. The system stiffness matrices are symmetrical and so only the upper triangular part needs to be calculated and stored. Also most of the elements in the matrix are zero. Element (i,j) in the stiffness matrix will be non-zero only if nodes i and j in the finite element mesh are part of the same element. The matrix system is generally solved by factorizing the system stiffness matrix into the product of two triangular matrices. This process introduces new non-zero elements in each column between the most distant non-zero and the leading diagonal. The half-bandwidth of a matrix, HBW, is defined so that 2HBW−1 = MAX(i−j): K(i,j)≠0. During factorization non-zeros can only occur in the HBW diagonals closest to the leading diagonal. The NAG Finite Element Library includes routines which construct and solve system stiffness matrices storing only this band of the matrix. Even storing the matrix in this form requires large amounts of storage. A three dimensional finite element model of 10,000 nodes with a half bandwidth of 1000 would require 80 Mbytes of storage for the system stiffness matrix. Such a large data structure needs to be distributed around the network of processors. This is not only because a single processor is unlikely to have this much memory but also to balance the amount of calculation performed by each processor.

To determine what data distribution yields the best performance it is necessary to consider the amount of work required by each stage of the model. Building an element stiffness matrix requires $O(n^3)$ floating point operations for each n noded element. For a mesh consisting of elements of only one shape the number of elements is proportional to the number of nodes so the amount of calculation involved in building a system matrix is $O(N)$ where N is the number of nodes. The factorization of a matrix is an $O(N^3)$ operation while forward and backward substitution is an order $O(N^2)$ operation. Given this information, it is reasonable to distribute the matrix so as to optimize the dominating factorization stage.

6.3 Building the System Stiffness Matrix

The system stiffness matrix is constructed by summing the contributions from element stiffness matrices calculated for each element in the finite element mesh. The element stiffness matrices are symmetrical and are calculated by summing the contributions stored in the S(ij,k,element_shape) array with weights calculated from

the conductivity distribution, see Equation 4.3. The calculation of each element stiffness matrix can be performed independently and concurrently in any process that has the mesh data. If the elements are allocated evenly to the processors the element stiffness matrices may be calculated in parallel, with no inter-process communication and very good load balancing.

Some consideration needs to be given to the construction of the system stiffness matrix from the element stiffness matrices. The coefficients in each of the element stiffness matrices need to be added to the correct positions in the system stiffness matrix. As the columns of the system matrix are distributed among the processors to facilitate factorization, this stage requires communications between all the processors. The coefficients of the element matrices need to be transmitted to the processors that store corresponding columns of the system matrix. If this is done as each element stiffness matrix is calculated, the near random communications that are required are difficult to load balance and require complex synchronisation between processors. A simpler and faster approach is to allow each processor to construct its own local system matrix from all the element matrices calculated on that processor. Each processor may sequentially calculate its own element stiffness matrices and sum the contributions to the local system matrix. These calculations may be performed locally with no inter-process communication. The distributed system matrix may then be calculated by synchronised rotations and addition of the columns of the local system matrices. This method concentrates the inter-processor communications into a single, highly regular, period and so results in simpler, readily maintainable programs that execute more quickly. Two data structures are needed to implement this scheme. A system stiffness matrix local to each processor is required with all the columns represented. After the local stiffness matrices have been added together and distributed among the processors a global system stiffness matrix is required with a subset of columns on each processor. The global system stiffness matrix data structure needs to be large to allow for fill-in during factorization. If its columns are distributed round robin fashion this free space allows the two data structured to co-exist in the same physical memory. As the local stiffness matrices are added together they overwrite the local matrix to form the global system stiffness matrix.

6.4 Solution Of The Finite Element System

The finite element system is a set of linear equations which can be expressed in matrix form as $K\phi=\mathbf{F}$, where K is the system stiffness matrix, ϕ is the vectors of unknown potentials and $\mathbf{F}$ is the forcing vector which in our case describes the current crossing the boundary of the region. The matrix K is symmetric, positive

definite and most of its elements are zero. The following sections investigate a range of algorithms, both serial and parallel, which can be used to solve this system. Section 6.5 explores direct methods, in particular Cholesky factorization. Indirect methods are investigated in Section 6.6 and semi-direct methods in Section 6.7.

6.5 Direct Methods

6.5.1 Cholesky Factorization

The Finite Element matrices and the least squares matrix generated during an electrical impedance reconstruction are both symmetric and positive definite. The finite element matrices have the added feature that they are irregularly sparse. The classical method of solving the system $\mathbf{Ax=b}$ where A is a symmetric, positive definite matrix is to factorize it into a product of a lower triangular matrix and its transpose by Cholesky factorization and then calculate $\mathbf{x}$ via forward and backwards substitution:

$A = LL^T$	Cholesky factorization.
$\mathbf{Ly=b}$	Forward substitution.
$L^T\mathbf{x=y}$	Backwards substitution.

The Cholesky factorization algorithm for calculating the lower triangular matrix L is given by Golub and Van Loan, [33]:

FOR k=1 TO N

$$A_{kk} = \left(A_{kk} - \sum_{p=1}^{k-1} A_{kp}^2 \right)^{1/2}$$

FOR i=k+1 TO N

$$A_{ik} = \left(A_{ik} - \sum_{p=1}^{k-1} A_{ip}A_{kp} \right)/A_{kk}$$

This algorithm requires $N^3/6$ floating point operations for a dense matrix, $A \in R^{N \times N}$, and over-writes the elements A_{ij} with the lower triangular matrix L_{ij}. The elements of the finite element system stiffness matrix are mostly zero. Algorithms exist which can factorize these so-called, sparse matrices with fewer operations, by not performing operations which combine zeros to produce a zero element during factorization.

6.5.2 Sparse Matrices

The definition of sparse matrices is rather imprecise but they can be thought of as matrices which have a relatively small number of non-zero elements. Many numerical operations involving sparse matrices are greatly accelerated by using algorithms which allow for the sparseness and do not perform calculations that have no effect on the result. For example, in calculating the dot product of two sparse vectors, $A.B = \Sigma A_i B_i$, the term $A_i B_i$ will be zero if either A_i or B_i are zero. If the sparse structure of A or B is sufficiently regular or sufficiently sparse then the dot product may be calculated more quickly by only considering the terms where both A_i and B_i are both non-zero. There is always a trade off in algorithms designed to use sparse matrices between the time saved by not performing redundant calculations and the overhead of deciding which calculations need to be performed.

6.5.3 Sparse Matrices and The Finite Element Method

Various algorithms exist for sparse matrices whose non-zero elements are clustered in some way. In these cases the matrix may be decomposed into blocks containing only zero elements and blocks that may be treated as dense matrices. The band storage described in Section 6.2, which stores each column as a dense vector of the same length as the half band width of the matrix, fits this description. The system stiffness matrix produced by a finite element model of a cylindrical mesh of tetrahedra will typically have ten to twenty non-zero elements in each column, corresponding to the number of nodes in elements containing a particular node. For the system stiffness matrix $K_{3Dp} \in \mathbf{R}^{10933 \times 10933}$ described later in this Section, the non-zeros constitute 0.1% to 0.2% of the total matrix. Factorization introduces new non-zeros in the matrix in a process known as *fill in*. Typically, after factorization the number of non-zeros will increase to between 1% and 2% of the matrix. These non-zeros will be irregularly scattered throughout the matrix. Algorithms that impose an artificial structure on the irregularly sparse matrix are not as effective as those that recognise the irregular distribution of information throughout the matrix. Algorithms for irregularly sparse matrices store only the non-zeros of the matrix and so are more efficient in terms of data storage and the amount of computation performed, if the matrix is sufficiently sparse.

6.5.4 Data Structures For Sparse Matrices

The most convenient way to specify a sparse matrix is as a set of triplets (K_{ij},i,j). A matrix stored in this fashion requires a real array and two integer arrays. However, factorization of a matrix requires a sequence of row or column operations

and so storage schemes that emphasize rows or columns yield more efficient algorithms. One alternative is to store each column of the matrix as a packed vector. Each column, j, is stored as doublets (K_{ij},i) in a real array and an integer array. A further array is necessary to store the number of non-zeros in each column. There is some benefit in storing the elements in each column in row order but this introduces an overhead in re-ordering the column after the introduction of a new non-zero. This may be alleviated by storing each column as a row ordered linked list. Duff, [23], reviews the characteristics of different sparse matrix storage schemes and their efficiencies in different operations. The choice of storage scheme is dependent upon the sparsity, sparsity pattern, and the computer used. For some computers, vector operations that operate on arrays of data perform as much as an order of magnitude faster than scalar operations on a single pair of operands. Increased performance can be obtained by arranging data so that the computation is *vector-rich* even if this introduces a large number of redundant operations. For a particular class of matrices and a particular computer, experimentation is necessary to determine the best algorithm and data storage to use.

6.5.5 Cholesky Factorization of Sparse, Finite Element Matrices

It has been recognised that sparse algorithms can increase the performance of large matrix operations. NAG and the SERC have produced a version of their Finite Element Library, called PARFEL, with parallel extensions for sparse matrix manipulation, [35]. The library called SPARSPAK, [17] contains software to solve sparse least-squares problems. Factorization of a matrix requires $O(N^3)$ operations for a dense matrix and is often the most computationally intensive matrix manipulation in an application. Both NAGFEL and SPARSPAK devote much of their effort to the efficient factorization of sparse matrices. Cholesky factorization is commonly used to solve the finite element systems encountered in EIT

Table 6.5 compares the run time on a Sun 386i of a sparse Cholesky factorization using two sparse algorithms. The first algorithm, used by NAGFEL, stores the HBW diagonals closest to the leading diagonal as dense vectors. The second algorithms recognises the irregular sparsity of finite element matrices and stores only the non-zeros as packed, ordered columns. Three system stiffness matrices were considered. Matrix $K_{2D} \in \mathbf{R}^{761x761}$ is the finite element system stiffness matrix for a two dimensional disk shaped mesh of triangles. Matrices $K_{3Dc} \in \mathbf{R}^{2405x2405}$ and $K_{3Dp} \in \mathbf{R}^{10933x10933}$ are for cylindrical meshes of tetrahedra. The matrix K_{3Dp} is suitable for modelling the potential on a cylinder driven by four rings of sixteen electrodes K_{3Dc} is a system stiffness matrix calculated using the

2405 noded mesh which is generally used to parameterise the conductivity when imaging a three dimensional cylinder. It is included only as an illustration of Finite Element matrices of its size.

The numbering of nodes in a finite element model is arbitrary. Re-numbering the nodes yields matrices that are similar with respect to symmetric permutation of the rows and columns. Thus the numbering can be adapted to suit the particular solution algorithm used. Two different node numberings are used for each matrix. For the banded storage scheme a numbering is chosen which approximately minimizes the half band width of the matrix. The irregular sparse algorithm is more efficient on matrices that introduce the minimum number of new non-zeros during factorization. A minimum fill-in node numbering is calculated using the Markowitz ordering algorithm described by Tinney and Walker, [82]. Duff [23] gives a review of optimal ordering algorithms. The irregularly sparse algorithm took approximately three times as long to factorize a band optimised matrix compared to the minimum fill-in matrix.

	Banded Sparse	Irregularly Sparse
Matrix K_{2D}	82.12 (2.86)	3.60 (0.21)
Matrix K_{3Dc}	3465 (34.86)	455 (8.26)
Matrix K_{3Dp}	Not Available	6488 (70.0)

Table 6.5 *Run time in seconds for Cholesky factorization and forward and backward substitution (in brackets) on a Sun 386i.*

6.5.6 Parallel, Cholesky Factorization of Dense Matrices.

By inspection of the Cholesky factorization algorithm, see Section 6.5.1, it is clear that most of the computation occurs in the calculation of the dot products of the partial rows; $\sum_{p=1}^{k-1} A_{ip}A_{kp}$. To effectively parallelise the Cholesky algorithm this work needs to distributed between the processors. It is relatively straight forward to parallelise the algorithm by distributing the columns of the matrix A in round robin fashion as in Section 5.7.6. Table 6.5a compares the execution time of the Cholesky factorization of a 208x208 dense, symmetric matrix, implemented on rings of P transputers, [67]. The execution time in the case of P=1 was for an algorithm optimised for a single processor.

P	Execution Time (s)	Speed up	Efficiency
1	11.1	1.0	100%
2	5.9	1.9	94%
3	4.2	2.7	92%
4	3.1	3.6	90%

Table 6.5a *A comparison of the execution times, in seconds, of dense Cholesky factorization on a ring of P transputers.*

6.5.7 Parallel, Cholesky Factorization of Sparse Matrices

Cholesky factorization of sparse matrices is more difficult to implement efficiently on a parallel computer. The data flow through the algorithm is more complex and the irregular amount of calculation at each stage makes load balancing difficult. Heath, [41], gives an excellent review of the data dependencies involved in sparse Cholesky factorization and suggests algorithms for a range of different hardware taxonomies including distributed memory, MIMD machines. If the sparsity pattern of the matrix is known beforehand, or many matrices of the same sparsity are to be factorized, there is benefit in analysing the sparsity to reduce the work of factorization. Heath and Duff, [23], consider permutations Q: $QAQ^T=LL^T$ such that the factorization introduces fewer non-zeros and exhibits higher degrees of parallelism.

The Cholesky algorithm may be written:

```
FOR k=1 TO N
    S(i)=0                          i=k,N
    FOR c=1 TO k–1
        S(i)=S(i)+A_kc A_ic         i=k,N
    A_ik = ( A_ik – S(i))/A_kk      i=k,N
```

This column form of the Cholesky algorithm calculates the columns of the Cholesky factor sequentially. It emphasises operations that can be performed within columns. The partial dot products required to calculate each column of the Cholesky factor occur in the inner loop in the variable c (for column). If the columns of the lower triangular part of the matrix A are distributed among P processors then each can

concurrently calculate its own S vector, S_i $1 \leq i \leq P$. This parallelises the major computational stage of the algorithm. The S_i vectors need to be summed and transmitted to the processor that stores the k^{th} column of A where they can be used in the calculation of the k^{th} column of the L matrix. If the columns of A are shared around a ring of processors with a round robin distribution then the work involved in forming the vectors S_i is nearly equally distributed and reasonable load balancing results. This form of Cholesky factorization also has the advantage that it lends itself well to computations with sparse matrices. The vector operation $S_i(j)=S_i(j)+A_{kc}A_{jc}$ j=k to N, need only be performed if A_{kc} is non-zero.

Once the matrix has been decomposed into its Cholesky factors the solution may be calculated via forward and backward substitution. The forward substitution can be achieved using the algorithm developed in Section 5.7.6 which uses the same data distribution as the Cholesky factorization developed in this Section. The backward substitution performed on this column-wise data distribution is inherently clumsy, even for dense matrices, and this affects the over all efficiency of the substitution stage. However, as backward substitution plays such a small part in the complete calculation, this is a penalty worth paying.

Table 6.5b compares the execution time of the parallel, sparse Cholesky factorization and substitution algorithms developed in this Section for two of the optimal fill-in, finite element system stiffness matrices described in Section 6.5.5. A system with 32 right hand sides was solved. The matrix K_{3Dp} could not be factored on our system due to memory limitations. The parallel programs were tested on a ring of four Transputers.

	Matrix K_{2D}	Matrix K_{3Dc}
CHOFACS	2	106
CHOSUBS	0.1	37

Table 6.5b *A comparison of the execution times, in seconds, of sparse Cholesky factorization (CHOFACS) and forward and backwards substitution (CHOSUBS) on a ring of four Transputers.*

Table 6.5c compares the execution time of the parallel, sparse Cholesky factorization of the finite element system stiffness matrix K_{3Dc} on rings of up to four

Transputers. The efficiency of the sparse algorithm drops much faster with increasing number of processors than the dense algorithm due to the more irregular data. Variations in the number of non-zeros in each column of the matrix and the number of columns effecting the calculation at each stage introduces load imbalances between the processors. These imbalances are less severe for rings with a small number of processors as each processor deals with larger, and hence more consistent, amounts of data. Irregularly sparse algorithms tend to be less efficient due to the irregularity of the distribution of work. The rapidly eroding efficiency of this algorithm limits the optimum number of processors that can effectively be used on the calculation.

P	Execution Time (s)	Speed up	Efficiency
1	342	1.0	100%
2	178	1.9	96%
3	126	2.7	90%
4	106	3.2	81%

Table 6.5c *A comparison of the execution times, in seconds, of sparse Cholesky factorization of the Finite Element matrix K_{3Dc}, on a ring of P Transputers.*

6.6 Indirect Methods

Many of the classical iterative algorithms for the solution of linear systems of the form $A\mathbf{x}=\mathbf{b}$, such as Jacobi, Gauss-Seidel, Over Relaxation methods, Richardson's method and the method of Kaczmarz may be written as:

$$\mathbf{X_{n+1}} = C\,\mathbf{X_n} + \mathbf{D}$$

where C is a constant matrix, **D** is a vector and $\mathbf{X_n}$ are a sequence of vectors that converge to the solution **x**. *In some cases these methods are faster than direct* methods and they generally require less memory. Their simple data dependencies make them highly vectorisable. Matrix-vector multiplications are straight-forward to distribute on a parallel computer with relative little communication and good load balancing if C is dense. If the rows of C and **D** are distributed between processors then at each iteration the corresponding rows of $\mathbf{X_{n+1}}$ may be calculated independently on the different processors. Each iteration would involve a multi-

node-scatter-gather to construct the complete $\mathbf{X_{n+1}}$ vector on each processor. An iteration of these algorithms requires $O(N^2)$ operations so they must converge to the desired accuracy within $O(N)$ iterations if they are to compete with direct methods. For sparse matrices the amount of calculation performed at each iteration is proportional to the number of non-zeros in the matrix, typically $O(N)$, and so indirect methods become much more attractive.

The convergence of all these algorithms depends upon the distribution of the eigenvalues of the matrix C. For Jacobi iteration to converge, the spectral radius of C must be less than one. Strict diagonal dominance of A is a sufficient condition for this to hold, [33]. Convergence becomes increasingly more difficult for larger matrices due to accumulation of errors during calculations. In practice these methods converge only for very special, yet still important, problems.

6.7 Semi-Direct Methods

6.7.1 The Preconditioned Conjugate Gradient Method

In recent years interest has grown in methods with guaranteed convergence, such as the Conjugate Gradient, CG, method. At each iteration this method chooses an α_n which minimizes the residual, $\mathbf{b}-A(\mathbf{X_n}+\alpha_n\mathbf{S_n})$, along some search direction $\mathbf{S_n}$ and the n^{th} iterate is updated by the correction: $\mathbf{X_{n+1}} = \mathbf{X_n}+\alpha_n\mathbf{S_n}$. The search directions, $\mathbf{S_n}$, are chosen to be an orthogonal set. For exact arithmetic the CG method is guaranteed to converge within N iterations. However, for ill-conditioned problems and fixed precision arithmetic, the convergence may be slow or it may not converge at all. Golub, [33] has shown that:

$$\varepsilon_n \leq \varepsilon_0 \left[(1-K^{1/2})/(1+K^{1/2}) \right]^{2n}$$

where K is the condition number of the matrix A, defined as the ratio of the largest to smallest eigenvalue, and ε_n is the error at the n^{th} iteration. The convergence can be accelerated by solving a better conditioned system, with condition number closer to one, produced using a symmetric preconditioner C: $(CAC)(C^{-1}\mathbf{x}) = (C\mathbf{b})$. Efficient algorithms exist for performing precondition conjugate gradient iterations without performing matrix-matrix multiplications or calculating C^{-1} explicitly.

6.7.2 Preconditioning by Incomplete Choleski Factorization

Many methods have been suggested for the calculation of preconditioning matrices for the solution of finite element systems. The least sophisticated preconditioning matrices are the iteration matrices for standard indirect methods such as Jacobi and Gauss-Seidel. Clearly there is a trade-off in the effort used to calculate the preconditioner and the rate of convergence we expect from it. This is particularly the case where the matrix A is distributed on a multi-processor system. The optimum preconditioner is $C = A^{-1}$ which converges in a single iteration. Thus the best preconditioners approximate the inverse of A. Where A has a regular structure an approximate inverse may be calculated assuming a similar sparsity pattern in C as in A. Lipitakis, [53] and [54], for example, considers several algorithms based on this idea and their implementation on parallel computers. However, for irregularly sparse matrices, the similar sparsity pattern assumption does not hold. Methods based on preconditioners calculated by solving the finite element problem on coarse grids were investigated by Wait, [84]. These methods have been shown to be effective yet the calculations involving two finite element meshes are more relevant to SIMD or shared memory machines due to the complexity and irregularity of data flow.

A particularly promising preconditioner is described by Meijerink and van der Vorst, [58]. During preconditioned conjugate gradient iterations a vector **y** needs to be calculated where: **y**=C**d**. Meijerink suggests setting C^{-1} to be the LU factorization of a matrix close to A: $C^{-1}=LU=A+R_\alpha$, where R_α is a matrix whose element are small. Their scheme for calculating L and U neglects terms in the LU factorization of A that are small or do not match some predetermined sparsity pattern. The procedure is known as an *incomplete LU factorization*. When the technique is applied to a symmetric matrix, such as in a finite element system, the symmetric preconditioner is calculated by an incomplete Cholesky factorization. In calculating a conjugate gradient iteration the systems of the form $y=Cd \Leftrightarrow LL^Ty=d$ can be quickly solved by forward and backward substitution. Once again there is a trade-off between the effort exerted in calculating an LL^T factorization close to A and the rapidity of convergence.

Table 6.7a compares the times required to solve the finite element system $K_{3Dc}\mathbf{x}=\mathbf{b}$, as described in Section 6.5.5, using Incomplete Cholesky Factorization Preconditioned Conjugate Gradient, ICFPCG. As K_{3Dc} is irregularly sparse a threshold was used when calculating the incomplete factorization. The coefficients in the matrix $R_\alpha=A-LL^T$ are restricted in absolute value to be less than the threshold α. As α is decreased the time to perform the incomplete factorization approaches that to calculate the full Cholesky factorization and the number of iterations required for

convergence decreases. Figure 6.7a displays a constant rate of convergence for a range of α's showing a stable and reliable algorithm. As α is decreased the factorization produces a less sparse L matrix which requires longer to perform a forward and backward substitution. The convergence gained in each iteration is approximately proportional to the time required for substitution. This test used a first estimate vector, x_0, equal to the zero vector. In practice, a good first guess can be constructed from the potentials calculated during the previous iteration of the reconstruction algorithm. As the reconstruction algorithm converges the conductivity changes between iterations become smaller and a first estimate generated using this method becomes better.

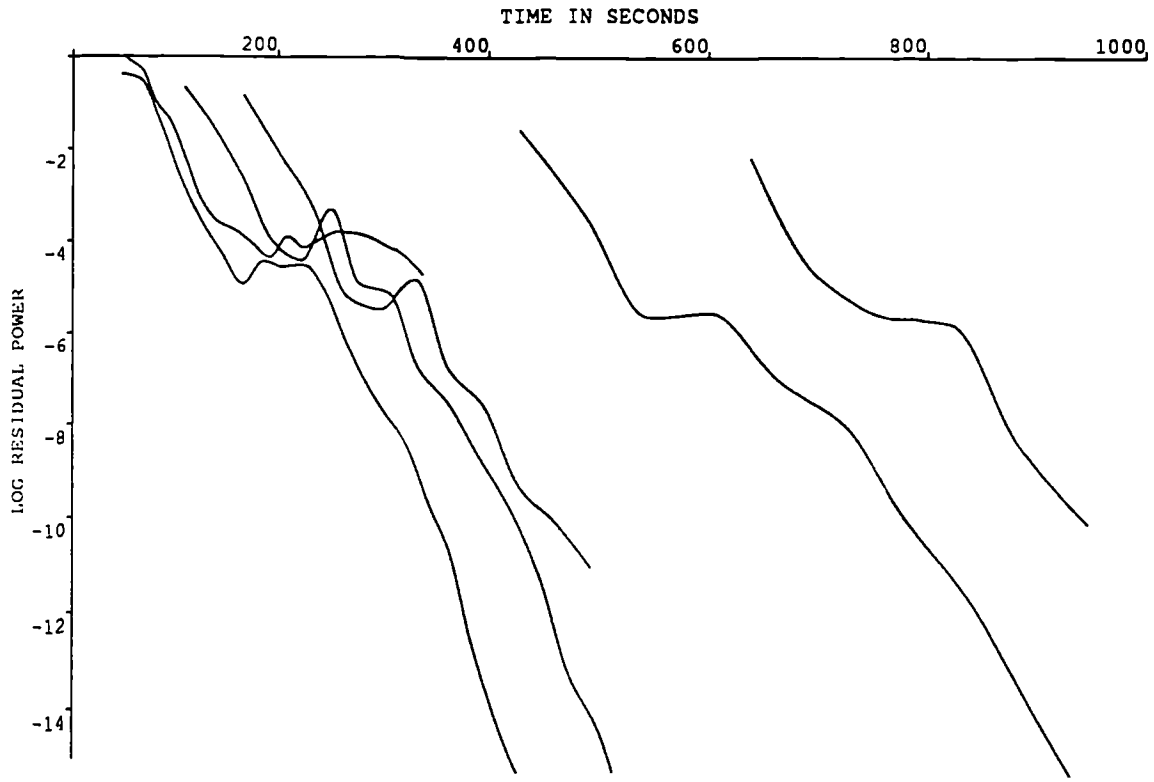


Figure 6.7a *A comparison of the convergence of Incomplete Choleski Factorization Preconditioned Conjugate Gradient Iteration, ICFPCG, used to solve the Finite Element system* $K_{3D_C}x=b$,. *Six preconditioners were calculated using different cut-off values. From left to right on the graph they were 1.0, 0.005, 0.001, 0.0005, 0.0001 and 0.00005. The measure of convergence is the log base 10 of the residual power;* $log(R^TR)$ *where* $R=K_{3D_C}X_n-b$.

A threshold needs to be set to decide when the ICFPCG method has converged to sufficient accuracy. The forward modelling part of the reconstruction algorithm needs to be able to model boundary voltages to at least the precision that electrical measurements can be performed on the region to be imaged, i.e. ≈0.1%. The boundary currents are electronically defined to approximately the same precision. The ICFPCG method calculates the residual power, $\mathbf{R}^T\mathbf{R}$ where $\mathbf{R}=K_{3Dc}\mathbf{x}-\mathbf{b}$. The residual vector is the error in the currents associated with the present best estimate potential in the finite element mesh. To calculate the relationship between residual power and the error in the potential it is necessary to calculate the smallest eigenvalue of the matrix K_{3Dc}. To perform this exactly is an infeasibly intensive calculation although Dongarra, [22], gives a quick method to calculate a numerical approximation. Generally the relative error in the boundary currents is larger than that in the potentials, as the boundary potential is a much smoother function than the potential, so it is reasonable to iterate until the residual power is of the same order as the relative error in the current vector. For the matrix considered this corresponds to a log residual error of $\log_{10}(\mathbf{R}^T\mathbf{R}) = -7$. The optimal choice of $\alpha=0.005$ achieves this in 300 seconds compared with 455 seconds for a full factorization.

The ICFPCG method has the potential to increase the speed of the forward modelling stage of three dimensional reconstruction. More importantly, the method uses a fraction of the memory needed for complete factorization. As yet the method has not been successful when applied to the matrix K_{3Dp} due to accumulation of errors during the calculation of the incomplete factorization. This is a promising area of further work.

6.8 Conclusions I

The forward modelling stage of a reconstruction algorithm occurs in two distinct stages; calculating the system stiffness matrix and solving the finite element system. For models with large numbers of nodes the solution of the finite element system dominates the time required for forward modelling.

The sparsity of the system stiffness matrix can be exploited by some solution algorithms to produce a result after less computational effort. Sparse direct methods were significantly faster than dense algorithms, both on serial and parallel computers. Sparse, semi-direct methods are potentially faster but, as yet, not reliable enough to be incorporated into robust reconstruction algorithms. They were found to be more complex than direct methods and so more difficult to implement in parallel.

6.9 Building the Derivative Matrix

Calculating and solving the Newton system bears many similarities to calculating and solving the finite element system. As with the finite element calculations, a matrix needs to be constructed using the S(ij,k,element_shape) data and then the matrix system needs to be solved.

Each row of the derivative matrix, $J = \partial E_{ij}/\partial s_k \in R^{MxN}$, corresponds to the derivative of a different experimental measurement, as defined in Section 2.2.3, with respect to the conductivity parameterisation. Each row is calculated in two stages. First the derivative with respect to the conductivities parameterised in the potential mesh basis functions is calculated using Equation 2.5b. Then each row is used to calculate the derivative with respect to the conductivities defined on the coarser conductivity mesh. The data required for this calculation are the potentials throughout the region calculated by the forward modelling stage of the reconstruction algorithm, the S(ij,k,element_shape) array, and the mesh correspondence array, MCA. The MCA contains the coefficients necessary to express the coarse, conductivity mesh basis functions as a linear combination of the finer, potential mesh basis functions. Any function approximated in the potential mesh can be approximated in the conductivity mesh by premultiplying by the MCA: $\mathbf{f}_\sigma = \text{MCA}\ \mathbf{f}_p$. Thus the derivative calculated using the S(ij,k) data may be mapped onto the coarser mesh using the MCA.

Each row of the derivative matrix may be calculated independently as no row depends upon any other row. Consequently each row may be calculated in parallel on a different processor assuming that each processor has access to the required data. In practice the number of processors is less than the number of experimental measurements used for reconstruction and so each processor calculates several rows. This distribution of effort results in good load balancing and very little inter-process communication. The drawback is the large amount of data that is needed by each processor. The potential data is available on all processors after the forward modelling stage of the algorithm but the MCA array can be large, even when stored as a sparse matrix. Storing a copy of the MCA on each processor requires a large mount of memory.

6.10 Solving the Newton System

In Section 2.6 two methods for solving the regularised, least squares

system, Equation 2.3c,d, were considered; one based on Cholesky factorization and the other on QR factorization. The method based on Cholesky factorization involves explicitly calculating the matrix $(J^TJ+\mu I)=LL^T$. The conductivity update can then by calculated from $LL^T\Delta\sigma_m=J^T\Delta V$ by forward and backward substitution. The matrix J^TJ may be calculated by summing the outer products of the rows:

$$J^TJ = \sum_{ij} \left(\frac{\partial E_{ij}}{\partial s}\right)\left(\frac{\partial E_{ij}}{\partial s}\right)^T. \qquad (6.10a)$$

where i,j index the experimental measurements used for reconstruction. The contribution from each row to the product matrix may be calculated in parallel. The way this matrix is distributed is determined by the requirements of the Cholesky factorization algorithm. The algorithm developed in Section 6.5.6 requires the data to be distributed by column, round robin fashion. If, after each processor has calculated a row of the derivative matrix, this data is spread around the processors using a multi-node-scatter-gather, each processor can use Equation 6.10a to calculate the necessary columns of J^TJ. The Tikhonov factor, μI, can easily be applied before factorization and substitution.

The second method considered was based on QR factorization. Although this method requires more operations, and hence takes longer to execute on a sequential machine, it could possibly be faster in on a parallel computer and has the potential to yield more accurate results. Two QR factorization algorithms were implemented on the Transputer network. The first used Given's rotations to zero the strictly lower triangular part of the matrix while the second used Householder transformations, [33]. Both algorithms require the same number of operations and the same amount of data is transmitted between processors during the calculations. Table 6.10b compares the performance of these two algorithms on a ring of Transputers, [67]. On a single Transputer the Householder method is faster due to the simpler organisation of the calculations into vector operations which allows extensive use of optimised GAXPY subroutines. The Householder method retains its advantages on multi-Transputer networks by transmitting vectors of data rather than the single rotation angles passed using the Given's method. QR factorization using Householder transformations has comparable efficiency to Cholesky factorization while requiring significantly more operations. Thus the Cholesky factorization method is significantly faster, even on a parallel machine. The reduced accuracy is not important as we are performing a linear step in a non-linear inverse problem. The error introduced by calculating the linear step to sub-optimum accuracy is small compared to the difference in behaviour between the linear approximation and the true non-linear function.

QR Factorization by Given's Rotations.

P	Execution Time (s)	Speed up	Efficiency
1	104	1.0	100%
2	65	1.6	80%
3	47	2.2	73%
4	38	2.7	68%

QR Factorization by Householder Transformations.

P	Execution Time (s)	Speed up	Efficiency
1	81	1.0	100%
2	43	1.8	94%
3	29	2.8	93%
4	22	3.7	92%

Table 6.10b A *comparison of the execution times, in seconds, of two QR factorization algorithms. The test matrix was of size300x208.*

Where the derivative matrix is under-determined, M<N, it is necessary to solve the regularised system $(JJ^T+\mu I)Y=\Delta V$. The (k,l)th element of the matrix JJ^T is the dot product of rows k and l of the derivative matrix. The matrix $(JJ^T+ \mu I)$ is small, symmetric, positive definite and dense so Cholesky factorization is the standard solution algorithm. A round robin distribution of the columns of $(JJ^T+ \mu I)$ is required. This can be accomplished by saving the rows of the derivative matrix as they are calculated on each processor. When all the rows have been calculated the columns of JJ^T can be calculated by rotating the calculated rows around the processors.

6.11 A Parallel Reconstruction Program.

All the stages in the reconstruction of an impedance image have been investigated in this Chapter. For each stage the most efficient serial method has been

sought and in each case this has lead to highly efficient parallel algorithms. These parallel algorithms have been chosen to use consistent data structures so that data is not re-distributed between stages of the reconstruction. The building blocks developed in this Chapter can be brought together to implement a range of reconstruction strategies.

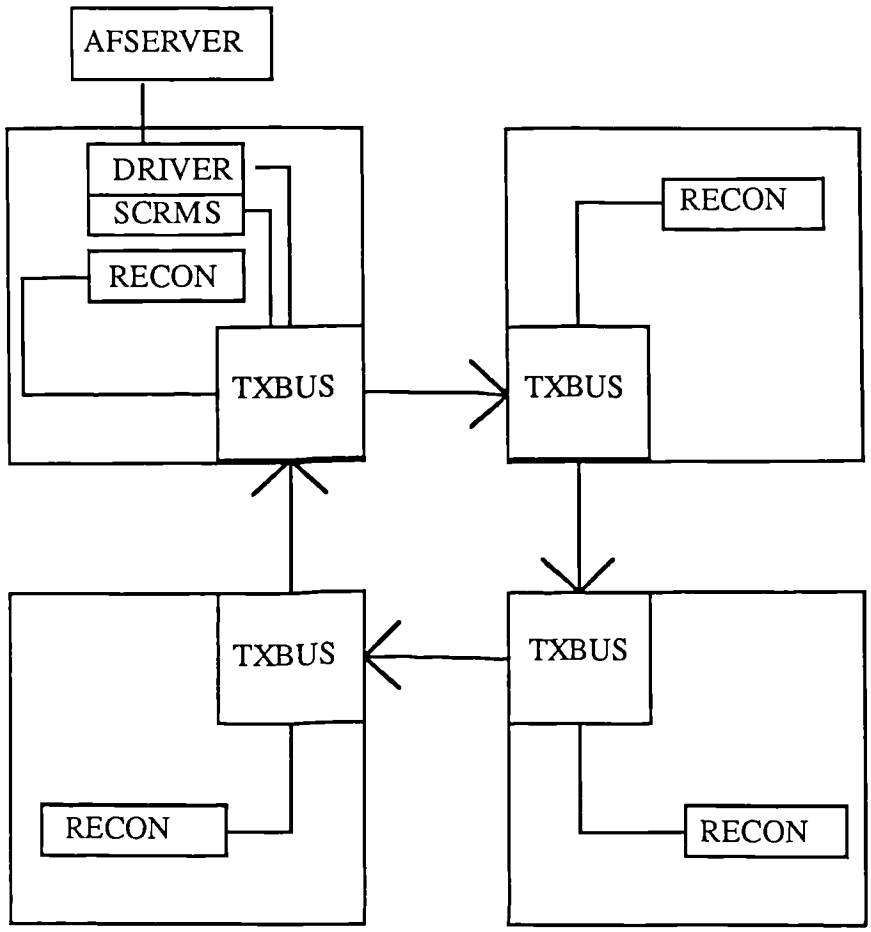


Figure 6.11 *Over all design of the parallel reconstruction application.*

The tasks that make up the parallel reconstruction program RECON and their configuration on a network of four Transputers are shown in Figure 6.11. The four Transputers are indicated by the large boxes. On each Transputer there is a reconstruction process, RECON, and a communications process, TXBUS. In addition the root Transputer has a process called DRIVER which controls all interaction with the host computer. A concurrently executing part of DRIVER called SCRMS intercepts error messages on the communication system and passes them to the host computer. The task AFSERVER, executing on the host computer, controls interactions with the Transputer network.

RECON is a parallel reconstruction program designed to execute on a uni-directional ring of any number of Transputers. It is integrated into the network wide communications system known as TXBUS and error reporting facility. An identical copy of the program RECON executes on each Transputer. The main module of RECON looks very similar to a serial implementation of the reconstruction algorithm. The subroutines called by RECON, for example those that construct the Finite Element system stiffness matrix or the matrix factorization routines, are implemented to perform their tasks in parallel. Global parameters tell each copy of the program, running on its own Transputer, about the configuration of the network, the number of Transputers etc. When the reconstruction program is initiated each copy of RECON is sent a Transputer Identification Number, IDTX, by the process DRIVER which identifies its place in the network.

The communications system, TXBUS, is comprised of an identical process executing on each Transputer. The TXBUS executing on each Transputer can communicate with the TXBUS's on the adjacent Transputers around the ring and to each concurrently executing process on its own Transputer. Messages are passed from TXBUS to TXBUS around the ring until they reach the destination Transputer. The message is then passed to the appropriate process on that processor. In this way messages can be passed from any process on any Transputer to any other process on any Transputer. A TXBUS process can hold a queue of messages waiting to be received by other processes executing on the same Transputer. Messages that are being routed around the ring are not delayed by the stalled messages waiting in this queue. This has been achieved by writing TXBUS as two concurrently executing processes, known as threads. A series of messages communicated between two processes will still always arrive in the order that they were sent.

A process known as DRIVER controls all communications between processes on the Transputer network and the host computer. The DRIVER is connected to a server, AFSERVER, executing concurrently on the host computer. Only a single process executing on the Transputer connected to the host computer, known as the root Transputer, can be connected to AFSERVER. Through AFSERVER the DRIVER can request the reconstruction data from the host file system. This data is then sent to the RECON processes executing throughout the network via the TXBUS communications system. DRIVER initalises all the declared processes by sending them their IDTX to identify their position in the network.

An error reporting subroutine can be called by any process on any Transputer. This subroutine constructs an error message which includes the name of the calling subroutine, the name of the calling process and the Transputer Identification Number.

The error message is passed around the ring until it reaches the root Transputer. SCRNMS, a thread of the process DRIVER, receives the error message and requests AFSERVER to print the error message on the screen of the host computer. This error reporting scheme has been quite successful but fails if there is a problem with a processor on the ring between the source of the error message and the root Transputer.

6.12 Conclusions II

In this Chapter the stages involved in performing a single iteration of a Newton based reconstruction algorithm have been investigated. The most computationally expensive stages in a reconstruction have been identified as the solution of the matrix equations during the forward modelling stage and the calculation of the conductivity update. Both the matrices in these systems of equations are symmetric and positive definite. The finite element matrix is irregularly sparse while the matrix used to calculate the conductivity update is dense. Of the methods investigated, Cholesky factorization was found to be the best solution method for both stages of reconstruction on serial and parallel computers. An irregularly sparse Choleski factorization algorithm was developed to solve the finite element system.

The reconstruction algorithm RECON has been implemented on a ring of Transputers using the best algorithms developed in this Chapter. The performance of this program is described in detail in Chapter 7.

Chapter 7

Performance of The Oxford EIT System

7.1 The OXPACT II System

The Oxford Polytechnic Applied Current Tomograph (OXPACT II) is a complete system for the imaging of conductivity distributions. It is comprised of four major components; a phantom, the Data Acquisition System (DAS), an Interactive Tomograph Controller (ITC) and a suite of reconstruction software. The DAS drives electric currents through a set of electrodes attached to the region to be imaged. Voltage measurements can be made on another set of electrodes interleaved with the current driving electrodes. For the OXPACT II system the electrodes are attached to a purpose built test object in which a range of controlled conductivity distributions can be set. The operation of the Tomograph is controlled by the ITC. It directs the operation of the DAS by specifying the currents to be set and the voltages to be measured. The data collected by the ITC is passed to an image reconstruction program which calculates an image. This image is displayed by the ITC.

7.2 The OXPACT II Phantom

7.2.1 The Design of the Phantom

The phantom is the experimental object upon which electrical measurements are made. The OXPACT II phantom and the rationale of its design are described in detail in Paulson *et al*, [71]. It is a perspex pipe of internal diameter 300 mm and a height of 50 mm with a flat lid and base also made from perspex. Set into the curved surface are 32, rectangular, current driving electrodes measuring 8.8 mm wide by 50 mm high. These electrodes are equally spaced around the curved surface with their edges parallel. This configuration is invariant in the vertical (z) direction and so, except for edge and corner effects at the top and bottom of the electrodes, yields electrical fields which are independent of z. The 32, small, needle shaped, voltage measurement electrodes are placed mid way between the current driving electrodes and half-way from the base to the lid. All the electrodes are gold plated to prevent them from corroding and in an attempt to provide constant and uniform contact impedance.

A design choice was made to use needle shaped point electrodes instead of

narrow, strip electrodes parallel to the current driving electrodes. The point electrodes were chosen so as to be relatively insensitive to the distortions introduced by the top and bottom of the phantom. The point electrodes have little effect on the electric fields in the phantom. Thus their existence can be neglected when modelling the phantom.

The configuration of the current driving and voltage measuring electrodes was chosen so that the phantom could be easily modelled using the finite element method. As the electric fields induced in the phantom are invariant in the z direction they can be simulated with a two dimensional model. It was found by experiment that current driving electrodes covering 30% of the boundary gave the best agreement between the Boundary Fourier method and the Finite Element method for small numbers of nodes, see Section 4.10. The configuration of electrodes described represents a compromise between the size of the signal we can expect to measure and the accuracy with which we can model the behaviour of the phantom (see Section 3.7). Voltage measurements made on large, current carrying electrodes would produce larger signals but would be very sensitive to variations in contact impedance and electrode placing while being difficult to model accurately using the Finite Element method.

The conductivity of the phantom can be set by filling it with saline solution of the appropriate concentration. Regions of near infinite conductivity contrast can be introduced by immersing blocks of metal or wood in the saline solution. Finite conductivity contrasts can be constructed from blocks of agar jelly doped with salt. The OXPACT II phantom allows real electrical measurements to be made on a region similar in size to the human chest with near arbitrary conductivity distribution.

7.2.2 Testing the Phantom

Considerable effort was put into determining the geometry of the phantom. The circularity of the cross-section of the phantom and the placement of the electrodes was set to within 0.2 mm, corresponding to an error of 0.07% in the diameter of the phantom. The largest error in the complete OXPACT II system is in the placement of the voltage measuring electrodes mid-way between adjacent current driving electrodes. The distance between the edges of adjacent current drive electrodes is 20.65±0.28 mm., a variation of 1.4%. When the accuracy of the placement of the voltage measuring electrode is taken into account, this can lead to an error of 3% in the measured voltage when current is driven between adjacent electrodes. This will lead to large voltage measurement errors for current patterns with high spatial frequencies.

As described in Section 3.6.1, a trigonometric current density, $J(\theta)$, applied to the boundary of a homogeneous disk of radius R and conductivity σ induces a boundary voltage $V(\theta)$:

$$J(\theta) = \sin(k\theta)$$

or $$J(\theta) = \cos(k\theta) \quad \Leftrightarrow \quad V(\theta) = \frac{R}{k\sigma} J(\theta)$$

The characteristic resistance, r_c, of a uniform disk can be defined as k times the amplitude of the boundary voltage divided by the amplitude of the boundary current density.

$$r_c = \frac{kR}{k\sigma} = \frac{R}{\sigma}$$

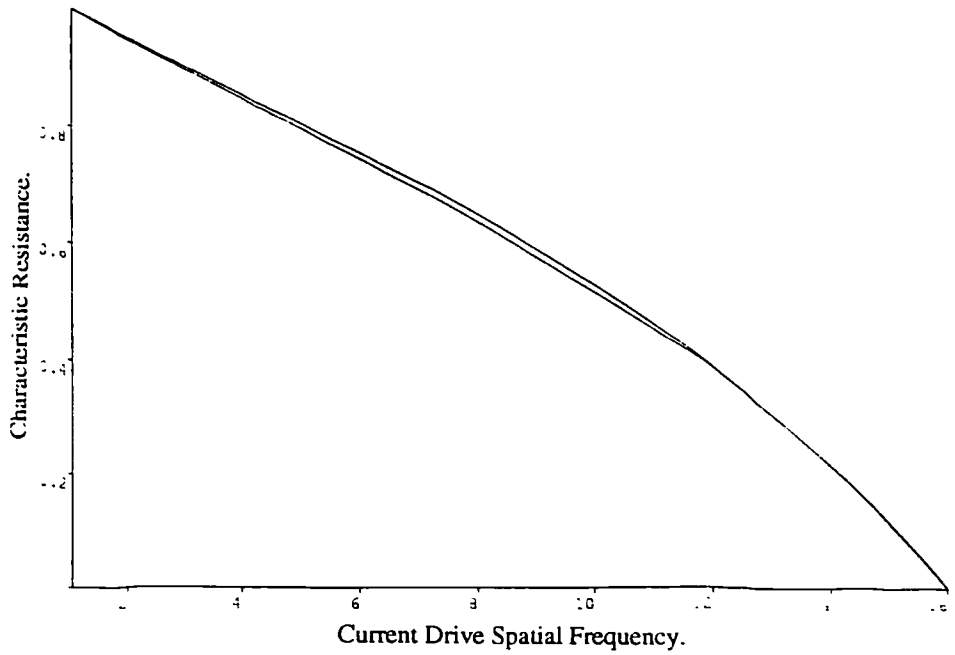


Figure 7.2 *A comparison of the characteristic resistance of the OXPACT II phantom and that predicted by the Boundary Fourier method for the same configuration of electrodes as a function of the spatial frequency of the current drive patterns.*

For a homogeneous disk driven by trigonometric current patterns the characteristic resistance is constant, independent of the driving spatial frequency. Cheng *et al*, [1], have shown that for the complete model of Section 3.5.3 the characteristic resistance is constant to ±10% for the range of applicable current drive spatial frequencies. Figure 7.2 compares the characteristic resistance of the phantom with that calculated using the Boundary Fourier model for the hybrid configuration of electrodes. Its near linear decline to zero with increasing spatial frequency of the driving currents is a

feature of voltage measurement mid-way between finite current driving electrodes, as described in Section 3.7. The deviation between experimental measurement and theoretical prediction is consistent with a systematic electrode placement error within the placement precision. This result shows agreement between model and experiment to within experimental error for a basis of current patterns. Consequently, we can be confident that the numerical model will accurately mimic the behaviour of the phantom for all applied current patterns.

7.3 The OXPACT II Data Acquisition System

Electrical experiments are performed on the phantom by the Data Acquisition System (DAS) driven by an Interactive Tomograph Controller, ITC, resident on a PC clone. The DAS is capable of setting an arbitrary set of currents on the phantom current driving electrodes. It can also measure the voltage between arbitrary pairs of voltage measuring electrodes or between electrodes and earth. A detailed description of the design and function of the DAS can be found in the Ph.D. thesis of Zhu, [90].

7.4 The OXPACT II Interactive Tomograph Controller

The Interactive Tomograph Controller, ITC, tells the DAS what current patterns to set on the phantom and what voltage measurements to make. It allows the user to choose a range of pre-defined current patterns and voltage measurement patterns or arbitrary current patterns can be set up by the user, one electrode at a time. The current or voltage on any electrode can be displayed on the screen as an RMS measurement or graphed as a function of time. The ITC can also display graphs of electrode voltages and currents as a function of electrode number. This provides a useful diagnostic when acquiring data for reconstruction. It has many checks built into it to test for fault conditions on the DAS. Typical faults would be electrode voltages out of the range of the measurement electronics or poor connection between an electrode and the phantom. The ITC also displays the images produced by the reconstruction software. These are displayed on the screen of the PC as colour coded contour plots with interactive user set parameters.

A novel method of current setting by voltage driving is used, see [89]. The ITC uses the DAS to measure the transfer impedance matrix of the phantom. The voltage pattern that would be induced by the desired current pattern is then calculated using the transfer impedance matrix. Electronic voltage drivers apply this voltage pattern to the current driving electrodes so inducing the desired current pattern. Matched voltage sources are much simpler devices to build than matched current sources and so their use has lead to a considerable simplification of the DAS. If the

induced current pattern is not sufficiently close to the desired pattern, an iterative refinement strategy can be used. With refinement this system is capable of setting current patterns with a difference norm of 0.2% of the desired current pattern. As current patterns that are close to optimal current patterns will also be close to optimal this is more than adequate for the purposes of EIT. The degradation of the data through setting currents close to optimal is not significant.

To generate data for impedance imaging the ITC sets a basis of trigonometric current patterns and measures the voltages on all the voltage measuring electrodes relative to earth. Two data files are written to disk; one containing the current patterns actually set and the other containing the voltage measurements. The voltage measurements can either be the raw electrode voltages or weighted sums of electrode voltages using the specified measurement patterns. Alternatively, the voltage difference between pairs of electrodes can be measured directly by the DAS. These data files are used by the suite of reconstruction software to calculate impedance images.

The data from the OXPACT system can be input into three different reconstruction software packages. Two of these packages, using different algorithms, are resident on a Sun 386i work-station and a third runs on a ring of four transputers in a Zenith XT.

The present system is not capable of performing fully adaptive reconstruction. This requires two way communication between the reconstruction package and the ITC, which are resident on different computers. Reconstructions performed by the present system are based on a single set of current and voltage measurements while adaptive reconstruction will use a different set for each iteration of the reconstruction algorithm. Adaptive reconstruction will be under the control of the ITC. It will calculate the optimal current and measurement patterns using the transfer impedance matrices of the phantom and the model used by the reconstruction software. These current patterns will be applied and the electrical measurement data files written to disk. The ITC will request a new best estimate conductivity and transfer impedance matrix to be generated by the reconstruction software using the latest set of experimental data. In a third file the ITC specifies the parameters of each reconstruction step, such as the number of iterations, number of current and voltage patterns to use, and the Tikhonov regularisation factor. This system, which is in a late stage of development, will allow interactive reconstruction of images based on the display of the present best solution.

7.5 Sequential Two Dimensional Reconstruction

Three sequential reconstruction algorithms have been implemented, two for two dimensional reconstruction and one for three dimensional. All three reconstruction programs accept the current pattern and voltage measurement files produced by the ITC as input data. A third file controls the parameters of the reconstruction. Two reconstruction strategies have been used, known as **RECON** and **POMPUS**. Two dimensional reconstruction has been implemented using both strategies while only POMPUS was implemented for three dimensional imaging.

7.5.1 RECON

RECON performs two dimensional reconstruction using pre-defined measurement patterns. The full RECON algorithm can be expressed in pseudo-code as:

WHILE $\| R(\sigma_m) - R(\sigma_e) \| > \varepsilon$ DO

- measure $R(\sigma_m)$ and calculate the first jmax optimal currents, J_j
- make the measurements E_{ij} :i=1,2,3,...,imax; j=1,2,3,...,jmax
- Solve the regularised, Least Squares equation using:

$$\Delta\sigma = \{J^T J + \mu I\}^{-1} J^T E$$

- $\sigma_m \rightarrow \sigma_m + \Delta\sigma$

ENDWHILE.

where J is the Jacobian matrix. Experimental measurements, E_{ij}, were defined in Section 2.2.3.

The measurement patterns used by RECON are either trigonometric or bi-polar. Bi-polar measurement patterns are equivalent to measuring the voltage difference between two electrodes, often adjacent pairs. Either set of measurement patterns form a basis for the space of voltage measurements. Although thirty one measurements are required for a basis, as we have the constraint $\Sigma(V^e_{i+1} - V^e_i) = 0$, thirty two measurements are made to maintain symmetry. These are of the form:

$$\mathbf{M^k} = \mathbf{e_{k+1}} - \mathbf{e_k} \qquad 1 \le k \le 31$$
$$\mathbf{M^{32}} = \mathbf{e_1} - \mathbf{e_{32}}$$

where $\mathbf{M^k}$ is the kth measurement pattern and $\mathbf{e}_i$ is the i^{th} standard basis vector. Thirty one trigonometric measurement patterns are used;

$$M_i^{2k} = \cos\left(\frac{2\pi k}{32}\left(i+\frac{1}{2}\right)\right) \quad : \qquad k=1,15$$

and $$M_i^{2k-1} = \sin\left(\frac{2\pi k}{32}\left(i+\frac{1}{2}\right)\right) \qquad k=1,16$$

where M_i^k is the weighting for the voltage measurement on the i^{th} electrode for the k^{th} measurement pattern.

Up to thirty trigonometric current patterns can be used for reconstruction:

$$I_i^{2k} = \sin\left(\frac{2\pi ki}{32}\right) \quad \text{and} \quad I_i^{2k-1} = \cos\left(\frac{2\pi ki}{32}\right) \qquad 1\leq k\leq 15$$

The sine pattern with k=16 applies no current and the cosine pattern with k=16 gives zero voltage readings on conductivity distributions that are homogeneous near the boundary.

An over-determined Jacobian matrix is constructed with a row for each experimental measurement. This configuration yields 30×31=930 theoretically independent measurements. As these measurements may be numerically dependent due to noise, typically less than 500 conductivity parameters are calculated. The associated regularised, Least Squares problem, Equation 2.3c, is solved using Cholesky factorization as described in Section 2.6.1.

7.5.2 POMPUS

POMPUS also performs two dimensional reconstruction. The full POMPUS algorithm can be expressed as:

WHILE $E_{11} > \varepsilon$ DO

- measure $R(\sigma_m)$ and calculate optimal M_i and J_j
- make the measurements E_{ii} : $E_{ii} > \varepsilon$
- solve the regularised, Least Squares equation using:

$$\Delta\sigma = J^T\{JJ^T+\mu I\}^{-1}E$$

- $\sigma_m \rightarrow \sigma_m + \Delta\sigma$

ENDWHILE.

For each iteration of the WHILE loop the experimental measurements E_{ii} are used which are greater than some noise level within the measurement process. OXPACT

II applies predetermined, trigonometric current patterns and so POMPUS has only partially been implemented. For each applied current pattern a single, optimal voltage measurement is made using the *half optimal* measurement pattern:

$$M_j^k = \frac{V_j^m - V_j^e}{|V^m - V^e|} \qquad \text{where} \qquad |V^m - V^e| = \sqrt{\sum_j \left(V_j^m - V_j^e\right)^2}$$

where M_j^k is the jth component of the kth measurement pattern and V^m and V^e are vectors of electrode voltages measured on the model and on the region respectively. This would be the full, optimal measurement pattern, as described in Section 2.2.5, if the applied current patterns were optimal. The regularised, Least-Squares problem solved by POMPUS is much smaller than that solved in RECON. POMPUS uses only one equation for each current pattern used in reconstruction compared with thirty-one for RECON. In both POMPUS and RECON the calculation of the conductivity update requires the factorization of a dense, symmetric matrix. The size of the matrix factorized by POMPUS is of the order of the number of experimental measurements used in the reconstruction, while in RECON it is of the order of the number of parameters in the conductivity mesh. As factorization is an $O(n^3)$ process, POMPUS runs significantly faster than RECON.

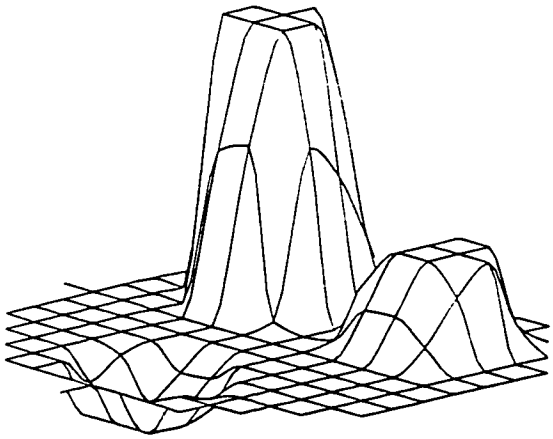
POMPUS uses far fewer constraints when calculating the conductivity update. If n current patterns are applied, the conductivity update used at each iteration of POMPUS is the minimum norm $\Delta\sigma_m$:

$$\left(\frac{\partial E_{ii}}{\partial \sigma}\right)^T \Delta\sigma_m = -E_{ii} \,. \qquad 1 \le i \le n$$

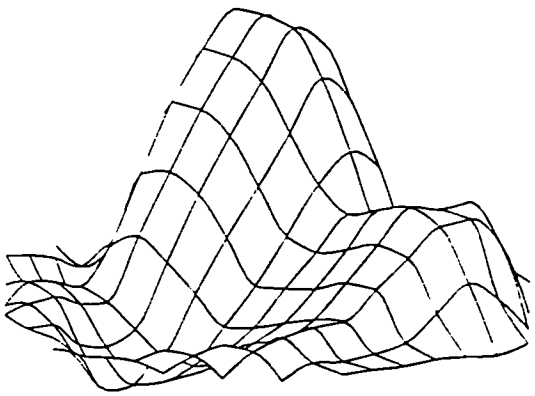
where there is one equation for each current pattern used for reconstruction. The other n(n–1) constraints:

$$\left(\frac{\partial E_{ij}}{\partial \sigma}\right)^T \Delta\sigma_m = 0 \qquad 1 \le i,j \le n;\ i \ne j \qquad (7.5a)$$

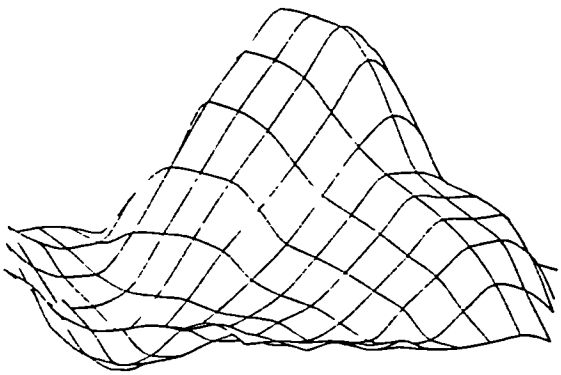
are not used when calculating $\Delta\sigma_m$. Paulson *et al*, [72], have shown that when σ_m and σ_e are both rotationally symmetric distributions on disks *the constraints* in Equation 7.5a are satisfied automatically by the conductivity update calculated using the method of Equations 2.3d. As the error in the experimental measurements, $\Sigma\Sigma E_{ij}^2$, is decreased by this algorithm, it is guaranteed to converge in the same sense as RECON. A convergence proof for POMPUS is given in the next three sections.



a) Target conductivity distribution.



b) RECON



c) POMPUS

Figure 7.5a *The reconstructed image of a uniform, two dimensional, disk of radius 15cm containing three anomalies of radius 4cm and conductivity contrasts of 0.5, 2.0 and 4.0. Image a) shows the target conductivity distribution. Image b) was produced by five iterations of RECON while image c) required seven iterations of POMPUS.*

7.5.3 Convergence of Reconstruction Algorithms

A reconstruction algorithm may be said to be converging if the norm of the difference in the transfer impedance operators, $\|D(\sigma_m)\|$ where $D(\sigma_m)=R(\sigma_m)-R(\sigma_e)$, is decreased at each iteration. An algorithm may converge to a local minimum of the function $\|D(\sigma_m)\|$ or to the global minimum $\|D(\sigma_m)\| = 0$ where no experimental measurements can distinguish the model and experimental conductivity distributions. Typically there will be some noise level, ε, within the measurement process. Conductivity updates calculated using experimental measurements smaller than ε are determined by the noise rather than the signal. Thus, reconstruction algorithms generally terminate when $\|D(\sigma_m)\|<\varepsilon$.

When $D(\sigma_m)$ is expressed in the optimal bases, defined by the matrices $U(\sigma_m)$ and $V(\sigma_m)$, it is a diagonal matrix, $D=Diag(E_{11},E_{22},....E_{pp})$. When current and measurement patterns are expressed in these bases the optimal patterns are the standard basis vectors, $\mathbf{e}_i$. Using these bases to define D allows the experimental measurements to be written $E_{ij}=\mathbf{e}_i^T D\mathbf{e}_j$. Thus as long as D is expressed in the local coordinate system defined by $U(\sigma_m)$ and $V(\sigma_m)$, the Frobenius norm of D is determined by its leading diagonal:

$$\|D(\sigma_m)\|_F^2=\sum_{i=1}^{p} E_{ii}^2(\sigma_m)$$

The steepest descent direction of a function F at the point x is $-\nabla F(x)$. A vector in a direction within 90^o of the steepest descent direction is known as a descent direction. Iterative algorithms for minimising a function, F, by repeatedly adding corrections along descent directions have been studied by Fletcher, [26]. These algorithms are shown to converge given very mild conditions on the smoothness of F and the size of the updates. The gradient of F, ∇F, must be uniformly continuous on the level set $\{x:F(x)<F(x_0)\}$ and the size of the update must satisfy the Wolfe-Powell conditions. The smoothness condition is satisfied by the function:

$$F(\sigma_m) = \|D(\sigma_m)\|_F^2.$$

Once a descent direction has been determined an update can be found which satisfies the Wolfe-Powell conditions in a finite number of steps. Thus, to show that the reconstruction algorithm converges it is sufficient to show that the conductivity update is along a descent direction and to allow the appropriate amount of the update

to be added at each iteration. In the remainder of this section it is shown that the POMPUS update is along a descent direction.

7.5.4 The Steepest Descent Direction.

The direction of steepest descent of the Frobenius norm of D is:

$$\mathbf{Z}=-\nabla_s\|D(\sigma_m)\|_F^2=-2\sum_{i=1}^{p} E_{ii}\nabla_s E_{ii}$$

where ∇_s is the gradient operator with respect to the conductivity parametrisation. When differentiating E_{ii} with respect to changes in the model conductivity distribution, changes in the optimal patterns used to define E_{ii} must be taken into account. The matrix D is diagonal when the currents and voltages are expressed in the singular bases. The diagonal elements of D are its singular values which are equal to the experimental measurements, $D_{ii}=E_{ii}=\lambda_i$, and the singular functions are the standard basis vectors, $\mathbf{e}_i$. The derivative of the singular values of D with respect to changes in its elements allows for the variation in the singular vectors of D. Thus the k'th component of the vector $\nabla_s E_{ii}$ can be written:

$$\left(\nabla_s E_{ii}\right)_k = \frac{\partial E_{ii}}{\partial s_k} = \frac{\partial \lambda_i}{\partial s_k} \qquad (7.5b)$$

An expression for the derivative of a singular value with respect to variations in a matrix can be obtained by a natural extension of the result of Horn and Johnson, [43]:

$$\lambda' = \frac{\mathbf{v}^T D'\mathbf{u}+\mathbf{u}^T D^T{}'\mathbf{v}}{\mathbf{u}^T\mathbf{u}+\mathbf{v}^T\mathbf{v}} = \mathrm{Re}\left(\mathbf{v}^T D'\mathbf{u}\right)=\sum_{i=1}^{p} E_{ii}^2(\sigma_m)$$

where **u** and **v** are the left and right singular vectors associated with the singular value λ. Thus, the derivative in Equation 7.5b may be written:

$$\frac{\partial E_{ii}}{\partial s_k} = \frac{\partial \lambda_i}{\partial s_k} = \mathrm{Re}\left(\mathbf{v}^T\left(\frac{\partial D}{\partial s_k}\right)\mathbf{u}\right).$$

Breckon, [9], has shown that, to a linear approximation, the change in the voltage measurement $v_{ij}=\langle M_i,R(\sigma_m)J_j\rangle$, Δv_{ij}, due to a conductivity change $\Delta\sigma=\Sigma s_i B_i=\mathbf{S}\cdot\mathbf{B}$ is:

$$\Delta v_{ij} = -\int_{\partial\Omega} \left(\sum_k s_k B_k\right) \nabla\phi_i . \nabla\varphi_j = -\sum_k s_k \int_{\partial\Omega} B_k \nabla\phi_i . \nabla\varphi_j$$

where ϕ_i and φ_j are the potential fields induced in the conductivity distribution σ_m by boundary current densities M_i and J_j respectively. As the voltage measurement, v_{ij}, differs from the experimental measurement E_{ii} by a constant additive factor, Equation 7.5b may be written:

$$\left(\nabla_s E_{ii}\right)_k = -\int_{\partial\Omega} B_k \nabla\phi_i . \nabla\varphi_i$$

The Steepest Descent direction, $\mathbf{Z}$, is thus $\mathbf{Z} = -2J^T\mathbf{E}$ where:

$$(J)_{i,k} = \frac{\partial E_{ii}}{\partial s_k} = -\int_{\partial\Omega} B_k \nabla\phi_i . \nabla\varphi_i \qquad \text{and} \qquad E=(E_{11}\ E_{22}\ E_{33}\ \ldots\ E_{pp})^T$$

7.5.5 The POMPUS Direction

The POMPUS direction is defined by the Least Squares system:

$$\Delta s = -J^T(JJ^T+\mu^2 I)^{-1}\mathbf{E}$$

The angle, α, between the unregularised POMPUS update and the steepest descent direction satisfies:

$$\cos(\alpha) = \frac{\Delta s . \mathbf{Z}}{\|\Delta s\|\|\mathbf{Z}\|} \qquad (7.5c)$$

The denominator of this expression is positive as it is the product of the norms of two vectors. When the Tikhonov regularisation factor is zero, $\mu=0$, the numerator can be shown to be positive:

$$\begin{aligned}\Delta s\cdot\mathbf{Z} = \Delta s^T\mathbf{Z} &= (-J^T(JJ^T+\mu^2 I)^{-1}\mathbf{E})^T(-2J^T\mathbf{E}) \\ &= 2\mathbf{E}^T((JJ^T)^{-1})^T JJ^T\mathbf{E} \\ &= 2\mathbf{E}^T\mathbf{E} \\ &= 2\|\mathbf{E}\|^2 \geq 0\end{aligned}$$

The condition that $\cos(\alpha)\geq 0$ implies that the angle between the POMPUS update direction and the Steepest Descent direction $|\alpha|\leq 90^{o}$. Thus the POMPUS update must decrease the Frobenius norm of D if the size of the update is sufficiently small. In practice the conductivity update is $\Delta\mathbf{s}\cdot\mathbf{B}$ unless this results in an increase in the norm of D in which case a smaller step can be made or the Tikhonov factor can be increased.

The denominator of Equation 7.5c can be simplified:

$$\begin{aligned}\|\Delta\mathbf{s}\|^2\|\mathbf{Z}\|^2 &= 4\mathbf{E}^T(JJ^T)^{-1}\mathbf{E}\mathbf{E}^T(JJ^T)\mathbf{E}\\ &\leq 4\|\mathbf{E}\|^4 K(JJ^T)\end{aligned}$$

where K(A) is the condition number of the matrix A. Equation 7.5c may be written:

$$\cos(\alpha) = \frac{2\|\mathbf{E}\|^2}{2\|\mathbf{E}\|^2\sqrt{K(JJ^T)}} \geq \frac{1}{K(J)}$$

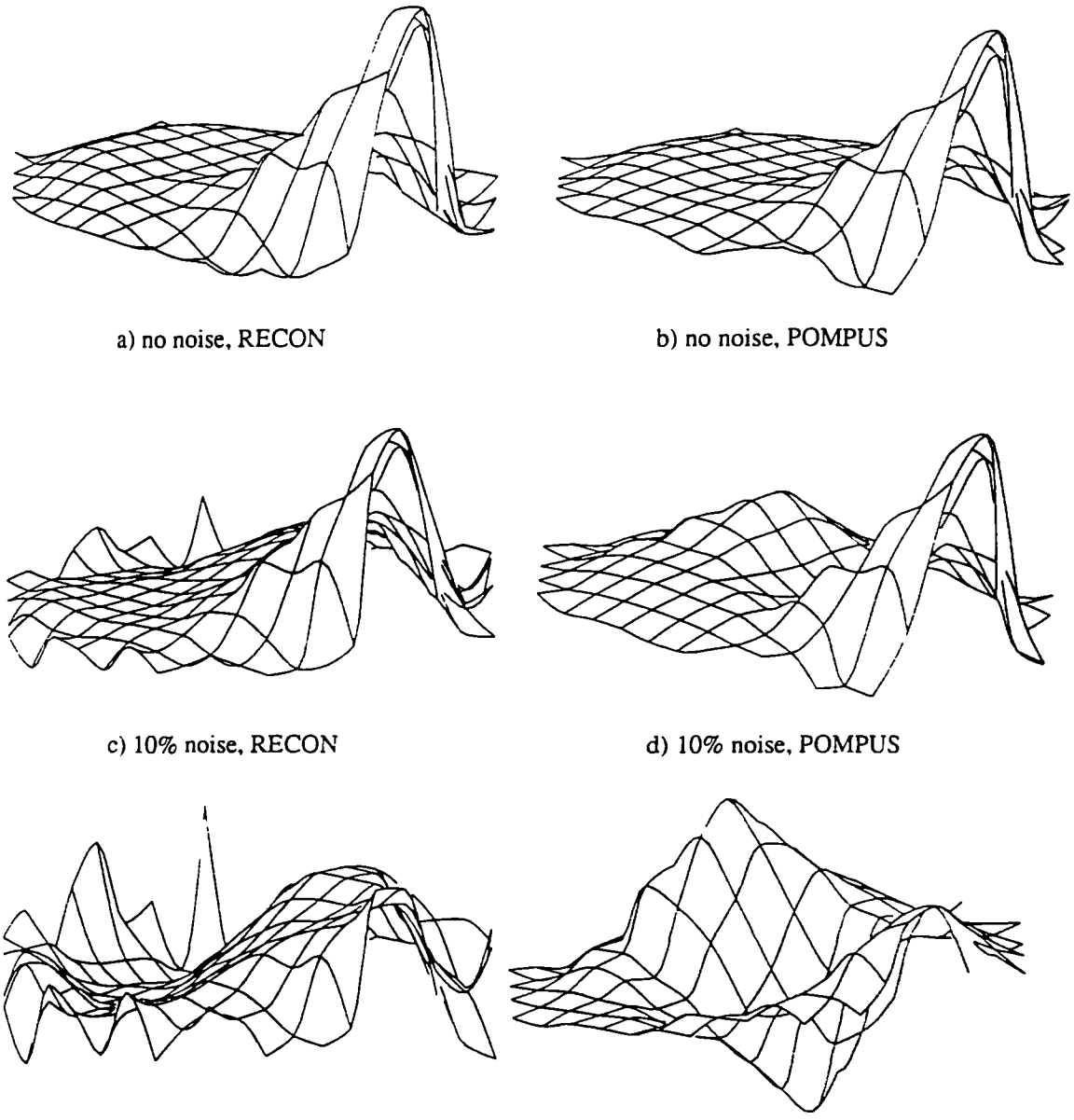
Thus the POMPUS direction diverges from the Steepest Descent direction as the Jacobian matrix becomes more ill conditioned. As the Tikhonov factor is increased the POMPUS direction converges to the Steepest Descent direction. As $\mu\rightarrow\infty$

$$-J^T\left(JJ^T+\mu^2 I\right)^{-1}\mathbf{E} = \frac{1}{\mu^2}J^T\mathbf{E} + O\left(\frac{1}{\mu^4}\right)$$

Thus, the POMPUS update is always in a descent direction and so will converge to a minimum of the function $\|D(\sigma_m)\|_F^2$.

7.5.6 Comparing RECON and POMPUS

The two, two dimensional algorithms were compared by performing reconstruction on synthetically generated data. The finite element method was used to predict the voltages that would be measured on the phantom if the conductivity distribution were uniform with three disjoint areas of different conductivity. These areas had conductivities of 0.5, 2.0 and 4.0 $(\Omega\text{cm})^{-1}$ compared to a background conductivity of 1.0 $(\Omega\text{cm})^{-1}$, see figure 7.5a. To avoid inverse crimes, which lead to spuriously good reconstructions, the synthetic data was generated using a much finer mesh than that used by the reconstruction software. Both programs used the K_{2D} finite element mesh described in Section 6.5.5 for the forward modelling part of the reconstruction. The resulting images are comparable in resolution.



a) no noise, RECON

b) no noise, POMPUS

c) 10% noise, RECON

d) 10% noise, POMPUS

e) 100% noise, RECON

f) 100% noise, POMPUS

Figure 7.5b *The reconstructed image of a uniform, two dimensional, disk of radius 15cm and conductivity σ=1 $(\Omega cm)^{-1}$ containing an anomaly of radius 1.5cm and a center 3cm from the edge with a conductivity σ=10 $(\Omega cm)^{-1}$. Images a,c and e were produced by RECON and b, d and e were produced by POMPUS.*

A second test was performed to determine the effect of noise on the two reconstruction algorithms. Synthetic data was generated for a two dimensional, disk shaped area containing an anomaly near one edge. White Gaussian noise was added to this data to yield noise to signal ratios (noise power divided by signal power) of 0%, 10% and 100%. Images reconstructed using RECON and POMPUS are compared in Figure 7.5b. Once again, RECON and POMPUS produce very similar results with POMPUS reconstructions degrading slightly faster with increasing noise. Comparisons under identical conditions are difficult as the two algorithms require different regularisations.

	RECON	POMPUS
Forward Modelling.		
Building FEM matrix	1.25	1.25
Solving FEM system	14.2	11.3
Calculate Update		
C1MESH		
Build Least-Squares system	212	6.72
Solve Least-Squares system	3.18	0.04
C2MESH		
Build Least-Squares system	565	9.02
Solve Least-Squares system	203	0.04

Figure 7.5c *Comparison of the time (seconds) required for a single iteration of the reconstruction algorithms on a Sun 386i.*

To quantify the relative speeds of reconstruction the time required to perform various stages of reconstruction using RECON and POMPUS are compared in Table 7.5c. For comparison, the conductivity image was calculated on two parameterisation meshes; C1MESH has 93 nodes and C2MESH has 381 nodes. The resulting images are virtually the same using these two conductivity meshes indicating that the resolution was limited by the noise rather than the number of conductivity parameters. Sixteen current patterns were used in each case.

The forward modelling part of the algorithms differ only in the number of right-hand-sides in the finite element system. These times are included for comparison with later stages in the reconstruction. The time required for POMPUS to construct the Newton system is much smaller than RECON due to the fewer

experimental measurements used for reconstruction. The Jacobian matrix built by RECON has thirty one times as many rows as that built by POMPUS. The Least Squares matrix factorized by RECON, using C1MESH, has 93 rows and columns compared to 16 for POMPUS. Modelling the conductivity on the much finer C2MESH, the matrix factorized by RECON has 381 rows and columns compared to 16 for POMPUS. These results show clearly the advantages of using POMPUS, especially where there is a large number of conductivities to calculate. This is always the case in three dimensional imaging where RECON would require the factorization of a matrix many thousands of elements square. This is not practical with the hardware presently available to us and so only POMPUS was implemented in three dimensions.

7.6 Sequential, Three Dimensional Reconstruction

A version of the POMPUS algorithm has been written especially for solving three dimensional problems. POMPUS3D solves the three dimensional forward problem and reconstructs the conductivity field in three dimensions. A numerical phantom has been constructed to simulate the behaviour of a saline tank. The model simulates a conducting cylinder of diameter 30 cm and height 24 cm, approximately the size of a human chest. Four layers of sixteen equally placed electrodes are modelled. They lie at the vertical levels z=–9.0, z=–3.0, z=3.0 and z=9.0 cm, assuming the cylinder is centred on the origin with the axis of the cylinder lying along the z axis.

Three dimensional trigonometric current patterns are used. These are the product of two trigonometric functions, one in the angular coordinate around the cylinder and the other in the vertical coordinate. If electrode (i,j) is the i^{th} electrode on the j^{th} level the applied current patterns are:

$$I_{ij} = \mathrm{HORZ}\left(2\pi k\frac{i}{16}\right) \times \mathrm{VERT}\left(2\pi m\frac{j}{4}\right) \qquad 1\le i\le 16\ ,\ 1\le j\le 4$$

where

HORZ = sin for $1\le k\le 7$ or HORZ = cos for $0\le k\le 8$

VERT = sin for $m=1$ or VERT = cos for $0\le m\le 2$

These 64 current patterns form an orthonormal basis of the space of electrode currents. Only 63 current patterns are used as the pattern I_{ij} = cos(0)cos(0) fails the constraint that the net current crossing the boundary must be zero.

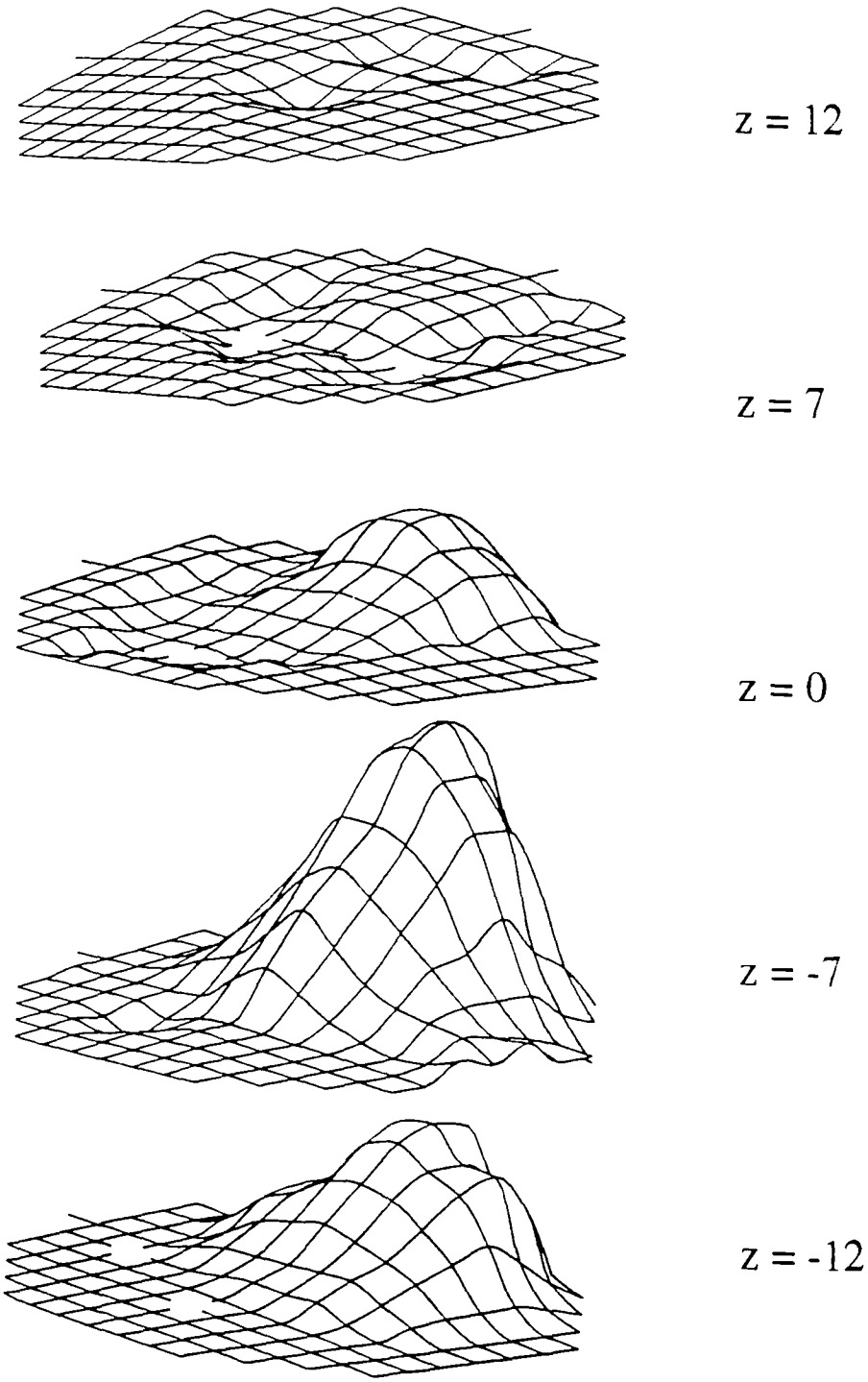


Figure 7.6b *Five horizontal slices through the image produced by POMPUS3D of a uniform cylinder, height 24cm and radius 15cm, with a spherical anomaly centred at (7,0,–7), radius 4cm. The resulting image has a conductivity contrast of 2.5:1 compared to a target contrast of 10:1*

Table 7.6a displays the run times for POMPUS3D. The potential was modelled on the K_{3Dp} mesh of 10933 nodes and 52416 tetrahedral elements while the conductivity was parametrised on the K_{3Dc} mesh of 2405 nodes. Thirty two experimental measurements were used for reconstruction. The data was synthesized using a third mesh with the same number of nodes as K_{3Dp} but the nodes were in different places.

POMPUS3D	
Forward Modelling.	
Building FEM matrix	72
Solving FEM system	6908
Calculate Update	
Build Least-Squares system	872
Solve Least-Squares system	4.5

Table 7.6a *The time (seconds) required for a single iteration of POMPUS3D on a Sun 386i.*

The performance of POMPUS3D has been tested on synthetic data. The phantom was modelled using the finite element method for a uniform conductivity distribution σ=1 $(\Omega cm)^{-1}$ with a homogeneous, spherical anomaly, centred at (7,0,–7) with a radius of 4cm and a conductivity of 10 $(\Omega cm)^{-1}$. After three iterations of POMPUS3D the image in Figure 7.6b was produced. Five horizontal slices through the three dimensional image are shown at levels z=–12, z=–7, z=0, z=7 and z=12 cm. The image has a conductivity contrast of 2.5:1.

7.7 Parallel Reconstruction

An implementation of RECON for two dimensional reconstruction has been written for rings of Transputers. Parallel versions of the algorithms used in the serial RECON are used in the Transputer implementation. The images produced by the two programs are identical to machine precision. Table 7.7a compares the times required for the various stages of a reconstruction iteration on two, three and four Transputers connected into a ring. The reconstruction program is too large to fit on a single Transputer.

Number of Transputers	Two	Three	Four
Forward Modelling.			
Building FEM matrix	<1	<1	<1
Solving FEM system	9	8	8
Calculate Update			
C1MESH			
Build Least-Squares system	50	33	26
Solve Least-Squares system	1	<1	<1
C2MESH			
Build Least-Squares system	166	108	83
Solve Least-Squares system	40	26	20

Table 7.7a *Comparison of the time (seconds) required for a single iteration of RECON on a ring of two, three and four Transputers.*

The forward modelling part of the reconstruction algorithm shows almost no improvement as more Transputers are devoted to the calculation. This is due to the poor efficiencies in the forward and backward substitution stage. We would expect to see better performance with larger or denser Finite Element systems. The second stage of the reconstruction algorithm, calculating the conductivity update, exhibits near linear speed-up as the number of processors is increased. This implies that significant speed-ups could be achieved with greater numbers of processors. The ultimate performance of the reconstruction algorithm as a whole with increasing numbers of processors is limited by the least parallelisable portion of the program. As the number of processors is increased it is expected that the time required for forward modelling would slowly increase. If the time required to calculate the conductivity update continued its hyperbolic decline to zero with increasing numbers of processors, the optimum speed of reconstruction would be approximately 8 seconds. As the execution time of POMPUS is dominated by the forward modelling stage it is expected that a parallel implementation would run at the same optimal rate as RECON.

7.8 Performance of The OXPACT II System

The performance of the OXPACT II system can be measured by its ability to image conductivity distributions. The system needs to be able to resolve disjoint

areas of the same conductivity and to resolve areas of different conductivity. Both the spatial and conductivity resolution are limited by the errors in the data used for reconstruction. These errors include errors in the measurement of current and voltage, errors in our knowledge of the shape of the boundary, the boundary conditions and the position of the electrodes and errors in the model used by the reconstruction programs.

The DAS can measure the current passing through any of the current driving electrodes to an accuracy of 0.2%. Voltage measurement accuracy depends upon the impedance of the imaged region. The system is designed to cope with point-to-point impedances in the range 100Ω to 3000Ω and can measure voltages in the range ±5 Volts. If full scale voltages are induced on the electrodes they can be measured to an accuracy of 0.003%. Smaller voltage measurement are made to less precision down to approximately 1% for very small voltage readings.

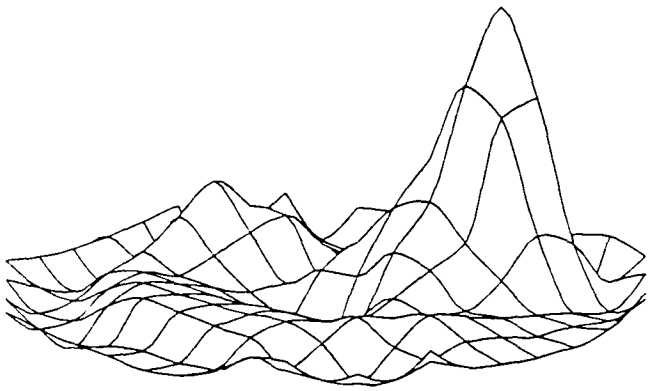


Figure 7.8a *The image produced by RECON from data collected from the OXPACT II phantom filled with isotonic saline. A stainless steel cylinder of radius 1cm was immersed in the saline half-way from the edge of the phantom to the center.*

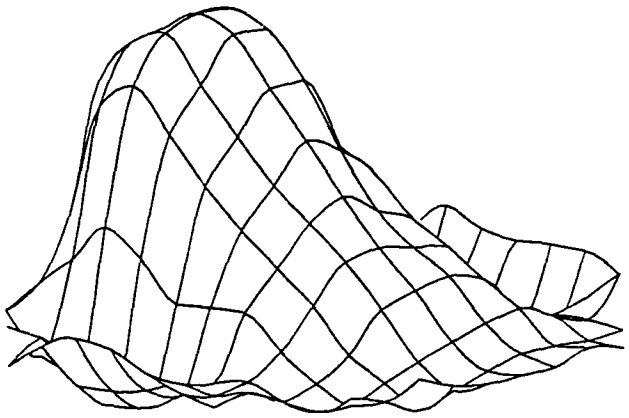
The quality of the images produced by the OXPACT system has been limited by the characteristics of the phantom upon which data was acquired. Knowledge of the boundary shape is essential if the forward modelling is going to be accurate. It is also of utmost importance to know the electrode positions and to be confident of their electrical attachment to the region to be imaged. The error in the gap between electrodes of 2.3% swamps any errors in the DAS electronics. More seriously, this is a consistent error which cannot be averaged away. Variation of the contact impedance between electrodes and the saline in the phantom introduces another large

error. It was found through experience that the phantom needed to be thoroughly cleaned with alcohol to remove any grease from the current drive electrodes or the current fields within the saline would be sufficiently disturbed to change electrode voltage measurements. Slow deterioration of the phantom over its lifetime has been evident. The gold plating on the electrodes has gradually flaked off due to corrosion underneath the gold and physical wear due to constant cleaning. Corrosion of the needle electrodes has also caused problems.

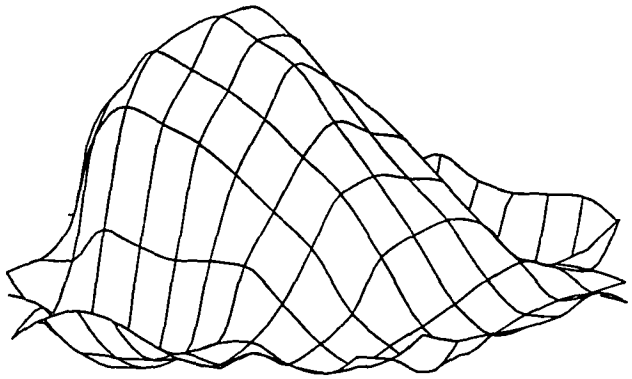
The spatial resolution of the OXPACT II system is demonstrated by the image of Figure 7.8a. A near infinite conductivity contrast was introduced into the phantom by immersing a stainless steel cylinder into the saline solution. The cylinder has a radius of 1cm and it was situated 75 mm from the edge of the phantom. Figure 7.8a shows an object with a conductivity contrast of 3.0:1 situated 72 mm from the edge of the image. The resolution of the system will decline as the target is moved towards the centre of the phantom.

The conductivity resolution of the system was tested by imaging three different anomalies introduced into the tank. Three agar blocks of radius 2.5 cm were produced. They were doped with salt until they had conductivity contrasts with the saline of 2:1, 1.5:1 and 1.25:1. Three sets of measurements were taken with the three different agar blocks immersed in the saline half-way between the edge and the middle of the phantom. Figure 7.8b compares these images. The peak conductivity contrasts in the images were 1.43:1, 1.24:1 and 1.21:1. Clearly some spatial resolution has been lost as the conductivity contrast approaches 1:1. Regularisation smooths the images and spreads the areas of high conductivity into the centre of the image where we have the least information on the conductivity distributions.

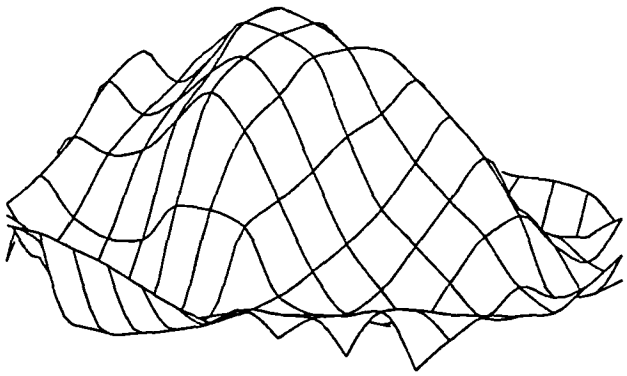
Figure 7.8c shows the image produced from data collected from the phantom when three different anomalies have been introduced. Three agar cylinders of radius 2.5 cm and with conductivity contrasts of 2:1, 1.25:1 and 0.5 :1 have been immersed in the saline at half radius positions and equally spaced around the phantom. The image, produced by POMPUS, clearly shows these three areas.



A) Image of a region with conductivity contrast 2:1.



B) Image of a region with conductivity contrast 1.5:1.



C) Image of a region with conductivity contrast 1.25:1.

Figure 7.8b *Three images produced by RECON from data collected from the phantom filled with isotonic saline. For each image an agar cylinder of radius 2.5cm was immersed in the saline with its centre half-way from the edge to the centre of the phantom. These cylinders were doped with salt to have conductivity contrasts of 2:1, 1.5:1 and 1.25:1 compared with the background saline.*

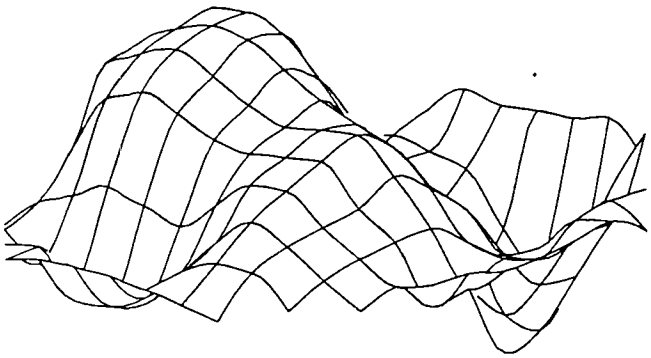


Figure 7.8c *An image produced by POMPUS from data collected from the phantom filled with isotonic saline. Three agar cylinders of radius 2.5cm were immersed in the saline with their centers half-way from the edge to the centre of the phantom. These cylinders were doped with salt to have conductivity contrasts of 2:1, 1.25:1 and 0.5:1 compared with the background saline.*

7.9 The Future of Parallel Computing

This investigation has produced a parallel program for the reconstruction of conductivity images. The program is designed to run on a ring of any number of Transputers. When the program is configured the number of processors needs to be specified. The performance achieved on our small system of four Transputers has been of a similar order to that provided by a work-station. The efficiency of the algorithms used has been high and so it is expected that performance would improve if larger networks of processors was used. However, it is expected that a few tens of processors would deliver optimal speed-up. The best performance we can expect from a network of Transputers would be an order of magnitude better than our present workstation.

The effort required to achieve this modest speed-up has been enormous. This is due partially to historical reasons. The project began when small networks of transputers were just appearing for general use and long before any development tools beyond compilers were available. If the project were started today the job would be considerably easier. However, the redesign work that was necessary to

port the sequential program to a parallel computer would be the same.

The parallel implementation is, by necessity, considerably more complex, and thus less maintainable or adaptable. It is also far less modular. As the data distribution often drives the choice of algorithm within the reconstruction, the dependencies have larger scope. For example, the potential field within the model conductivity region is calculated by the forward modelling stage of the algorithm and used to calculate the Jacobian matrix as a stage in the calculation of the conductivity update. The distribution of the data structure used to store the potentials is affected by the data structure used to store the system stiffness matrix and the algorithm used to solve the finite element system. It also affects the choice of data structure used to store the Jacobian matrix as well as the algorithm used to calculate it. A change in this data structure would require a large part of the program to be rewritten. For these reasons the parallel implementation of the reconstruction algorithm can be viewed as an unwieldy dinosaur. As a development tool it is useless, as any modification has drastic ramifications throughout the program and debugging is such an awesomely difficult task. Better algorithms often have a greater potential for speed increases than better hardware. This will become increasingly true in the future. The difficulty of using Transputers has reduced the number of alternative algorithms investigated and held up the progress of the group. I would recommend that after the completion of this project the parallel program is quickly disposed of.

Until the design of parallel programs becomes much more like the design of sequential programs they will not be worth the investment of human resource. The investment of time and effort into writing obscure and complex software to utilize state of the art computing devices has always been wasted unless the short term advantage has been the primary goal. It is expected that the speed of sequential (or apparently sequential) computers will continue to increase for the foreseeable future. A portable piece of software will eventually out-perform a program specialized to execute on leading edge but dead-end technology, simply by waiting for sequential machines to increase in speed. The latest PC purchased by the group already out performs our network of transputers. It executes a single iteration of RECON in 22 seconds compared to 35 seconds required by the Transputers. Similarly the workstations available today out perform the Sun 386i used in this project by a factor of 400 or more.

This is not to say that Transputers and parallel computing are doomed. Parallel computing is here to stay but considerably more maturity and experience is required in its use. This experience is required in both hardware and software. The development of large, completely connected networks will go some way towards this

goal. Independence of topology and the fast movement of data through the system will liberate the programmer from data distribution driven design. This will also allow data to be re-positioned dynamically during the execution of a program so that the optimal algorithm can be used at each stage. Once this has been achieved then libraries of parallel programs become practical and parallel programming becomes identical to sequential programming. Another route to this objective is via automatic parallelisation. Some progress has been made in this endeavour through graph analysis of program dependencies.

The next generation of OXPACT may perform the reconstructions on a parallel computer. If real time imaging is a goal then images will need to be produced at a rate of one every 40 ms. A succession of images displayed on a screen at this rate appear to be a smoothly moving image to the human eye. Computation at this rate is beyond the range of any available, individual processor. However, if an image can be produced in one second on a single processor, then twenty five processors, working independently, can produce twenty five images in one second; a nominal rate of one every 40 ms. Thus, a real time tomograph can be built from 25 processors, displaying smoothly moving images, if a delay of one second between data acquisition and display is acceptable.

7.10 The Future of OXPACT

7.10.1 Three Dimensional Imaging

There are several directions in which the OXPACT system can develop. The EIT research group has recently been awarded a grant from the Polytechnic Central Funding Committee to construct a three dimensional tomograph. This will require a three dimensional phantom and faster computing equipment than is used in the present generation of OXPACT. It is expected that the new phantom will be a cylinder with electrodes photographically plated onto the inside surface to achieve the desired electrode placement accuracy. The group expects to purchase a sequential work-station capable of sustaining 10-20 Mflops for use in the three dimensional tomograph. We hope to investigate the use of POMPUS on the reconstruction of real data as well as variations of the NOSER algorithm, [14], possibly using derivative matrices calculated using semi-analytic solutions to the first estimate forward model.

7.10.2 Clinical Imaging

Another grant has been awarded to the group, from the Wellcome Trust, to apply the group's experience to an investigation of the medical applications of a two dimensional tomograph. This investigation will be performed in two phases. In the first phase a fully adaptive, real time, medical system will be built. It is proposed that this system will be able to make 25 sets of measurements a second and will be able to display the data soon after acquisition. To achieve this, the system will require a considerably faster Data Acquisition System and reconstruction computer. The reconstruction computer may contain a number of processors working on different reconstructions simultaneously. In the second phase the system will be used to image living subjects in an attempt to measure physiological parameters. A formidable obstacle to this aim is the attachment of electrodes to the subject with sufficient accuracy. The group has commissioned the manufacture of electrode belts from the Northern Ireland Bioengineering Unit at the University of Ulster. These are sheets of strong plastic photographically imprinted with electrodes and cables. In addition the electrodes are covered with a *hydrogel* which improves electrical connection with the skin. These belts can be wrapped around the subject to ensure the electrodes are equally placed around the surface. Further measurement will be required to determine the surface shape. Initial aims of the medical study include determination of lung water and gastric emptying rates.

7.10.3 Multi-Frequency Imaging

A possible further direction of research is multi-frequency impedance imaging. At present the imaged region is assumed to be purely resistive. However a phase lag between the electrode currents and potentials of approximately 2^0 implies a significant capacitive component in the measured impedances. If electrical currents with frequencies higher than the 10 KHz used in the present OXPACT system were applied to tissue then structures on the scale of cells effect the conductivity. At low frequencies, current flows around cells in the extracellular liquid while at higher frequencies the current passes through cells, [85]. Measurements at a range of frequencies would allow full impedance images to be produced showing the conductivity and the permittivity inside the region. By comparing images at two frequencies it may be possible to calculate the ratio of intra and extracellular liquid. Griffiths [36] has produced difference images of conductivity and permittivity from synthetic data and later in [37] in data from a resistor-capacitor network. A full description of the design and development of a complex impedance tomograph can be found in [46].

7.10.4 Forward Modelling

Further acceleration of the reconstruction algorithms is likely to come from improvement in the forward modelling stage. This is particularly the case in three dimensional reconstruction where this stage dominates the execution time of the programs. Much of the *a priori* knowledge about the smoothness and form of the potential fields is not used by the Finite Element method. Some benefit could be gained by looking for potentials in spaces spanned by bases of smooth functions defined over the whole of the imaged region. On a circular, two dimensional region these basis functions, in polar coordinates, could be of the form:

$$B_{mn}(r,\theta) = P_m(r)\ \mathrm{trig}(2\pi n\theta)$$

where $P_m(r)$ is a polynomial in r and trig is either sin or cos. The Galerkin method can be used to calculate the coefficients necessary to express the solution potentials as linear combinations of these basis functions. If a suitable set of basis functions are chosen the linear system to be solved will be dense but much smaller than that produced by the Finite Element method.

7.11 Conclusions

The aim of this PhD project has been to investigate EIT reconstruction algorithms to improve the images that can be produced in both two and three dimensions and to find those algorithms that can execute quickly on both serial and parallel computers. In Chapter 2 a general framework for Newton type, iterative reconstruction algorithms was developed. A novel definition for an experimental measurement in EIT was introduced which leads directly to the concepts of optimal current and measurement patterns. In Chapter 5 a new reconstruction algorithm called POMPUS was described which uses optimal experimental measurements. POMPUS was compared to a standard, Newton type algorithm called RECON and was found to produce images of similar quality with a fraction of the computational effort.

The forward modelling of electric fields through known conductivity distributions was investigated in Chapter 3. Numerical and semi-analytic solutions of the conduction equation, 2.2a, were developed, for boundary conditions consistent with the application of current and voltage measurement through electrodes with contact impedance. An electrode configuration has been developed which is easy to model accurately using the Finite Element method and which yields measurements

which are relatively insensitive to variations in contact impedance. This finite element model has been incorporated into all the reconstruction algorithms investigated in this thesis.

Chapter 4 explored the implementation of reconstruction algorithms on networks of concurrently executing Transputers. A range of factorization and iterative methods for the solution of the matrix equation, $A\mathbf{x}=\mathbf{b}$, were investigated. Those algorithms found to be most relevant to EIT reconstruction were implemented efficiently on the Transputer network. They were incorporated into an parallel implementation of the two dimensional reconstruction algorithm, RECON. Although this implementation was found to have good scalability, the concurrency introduced sufficient complexity for the implementation to be unmaintainable, unmodifiable and unportable.

An impedance tomograph known as OXPACT II has been constructed and has produced absolute images of conductivity distributions from measurements made on a phantom. This system is currently being upgraded to use fully adaptive current patterns to further improve the images the system can produce.

This PhD project started with a sequential reconstruction program, based on the RECON algorithm. It never produced an image from real electrical measurements due to the inaccuracy of the crude, Finite Element forward model, with only 113 nodes, used for reconstruction. This program took fifteen minutes to perform a single iteration of the reconstruction on a Sun 386i workstation. During the course of this project many improvements have been made to the overall reconstruction algorithm and the numerical methods it uses. A much more complex finite element model, which adequately models the behaviour of the OXPACT II phantom, has become standard in our two dimensional reconstruction programs. This model uses 761 nodes and hence a much larger Finite Element system need to be solved, but it allows real electrical measurements to be reconstructed. All the reconstruction algorithms reported in this thesis have benefitted greatly from the use of algorithms which exploit sparse matrices and the introduction of the S(ij,k,element_shape) data structure. The use of optimal experiments and the POMPUS algorithm has massively increased the speed of reconstruction and made three dimensional reconstruction practical. The most recent version of POMPUS can perform an iteration of two dimensional reconstruction in under 3 seconds.

References.

[1] G. Amdahl, The Validity of the Single Processor Approach to Achieving Large Scale Computing Capibilities, AFIPS Conf. Proc. Spring Joint Comput. Conf., No. 30, pp483-485, 1967.

[2] G. R. Andrews, Concurrent Programming Principles and Practice, The Benjamin/Cummings Publishing Company,1991.

[3] D. C. Barber and B. H. Brown, Applied Potential Tomography, J. Physics E, 17, pp723-733, 1984.

[4] D. C. Barber and B. H. Brown, Rescent Developments in Applied Potential Tomography - APT, in Information Processing in Medical Imaging ed. S. L. Bacharach, Martinus Nijhoff, pp106-121, 1986.

[5] D. P. Bertsekas and J. N. Tsitsiklis, Parallel and Distributed Computation Numerical Methods, Prentice-Hall International Inc., 1989.

[6] W. R. Breckon and M. K. Pidcock, Ill-posedness and Non-linearity in Electrical Impedance Tomography, in "Information Processing in Medical Imaging", edited by C.N. de Graaf and M. A. Viergever, Plenum, pp 235-244, 1988.

[7] W. R. Breckon and M. K. Pidcock, Some Mathematical Aspects of Electrical Impedance Tomography, Mathematics and Computer Science in Medical Imaging", edited by M. A. Viergever and Todd-Pokropek, Springer, pp 204-215, 1988.

[8] W. R. Breckon, K. S. Paulson and M. K. Pidcock, Parallelism in EIT Reconstruction, Information Processing in Medical Imaging, Ed. D.A. Ortendahl and J. Llacer, Wiley-Liss, pp187-196, 1989.

[9] W. R. Breckon, Image Reconstruction in Electrical Impedance Tomography, PhD Thesis, Oxford Polytechnic, 1991.

[10] W. R. Breckon, Measurement and Reconstruction in Electrical Impedance Tomography, in "Inverse Problems and Imaging", edited by G. F. Roach, Pitman, Research Notes in Mathematics Series, Vol 245, Longman Scientific and Technical, pp1-19, 1991.

[11] B. H. Brown and A. D. Seagar, The Sheffield Data Collection System, Clin. Phys. Physiol. Meas., 8 Suppl A, pp91-98, 1987.

[12] B. H. Brown, Overview of Clinical Applications, Proc. of EC COMAC-BME Workshop on EIT, Copenhagen, pp 29-35, 1990.

[13] B. H. Brown, A. Leathard, A. Sinton, F. J. McArdle, R. W. M. Smith and D. C. Barber, Blood Flow Imaging Using Electrical Impedance Tomography,Clin. Phys. Physiol. Meas., Vol 13, Supplement A, pp 175-180, 1992.

[14] M. Cheney, D. Issacson, J. C. Newell, S. Simske, J. Goble, NOSER: An Algorithm for Solving the Inverse Conductivity Problem, International Journal of Imaging Systems and Technology, vol 2, pp 66-75, 1990.

[15] M. Cheney, D. Issacson, E. J. Somersalo, E. L. Issacson and E. J. Coffey, A Layer Stripping Reconstruction Algorithm for Impedance Imaging, Proc. IEEE-EMBS conf. (Orlando), No 13, pp 3-4, 1991.

[16] K. Cheng, D. Issacson, J. C. Newell and D. G. Gisser, Electrode Models for Electric Current Computed Tomography, IEEE Transactions on Biomedical Engineering, vol 36, No. 9, Sept, pp 918-924, 1989.

[17] E. Chu, A. George, J. W. H. Lui and E. J. Y. Ng, User's Guide for SPARSPAK-A: Waterloo Sparse Linear Equations Package, Tech. Report CS-84-36, University of Waterloo, Waterloo, Ontario, Nov. 1984.

[18] R. Courant, Variational Methods for the Solution of Problems of Equilibrium and Vibrations, Bull. Amer. Math. Soc. 49, pp 1-23, 1943.

[19] R. Courant and D. Hilbert, Methods of Mathematical Physics, Vol. 1, Wiley, 1966.

[20] A. R. Daniels, I. Basarab-Horwath, F. Dickin, R. G. Green and C. Thornhill, Initial Findings on a Tomographic Imaging System for Sewers, Preprint, 1992.

[21] J. E. Dennis Jr., D. M. Gray, and R. E. Welsch, TOMS, No. 7, pp 348-368, 1981.

[22] J. Dongarra, J. R. Bunch, C. B. Moler and G. W. Stewart, LINPACK Users Guide, SIAM Publications, Philadelphia, 1976.

[23] I. S. Duff, A. M. Erisman and J. K. Ried, Direct Methods for Sparse Matrices, Clarendon Press, Oxford, 1986.

[24] Eung Je Woo, Finite Element Method and Reconstruction Algorithms in Electrical IMpedance Tomography, PhD Thesis, University of Wisconsin-Madison, 1990.

[25] M. Faraj, I. Basarab-Horwath, A Prototype Distributed Preasure Sensor Utilising Electrical Impedance Tomography; Initial Results, Process Tomography A Strategy For Industrial Exploitation, Report from the European Concerted Action on Process Tomography, Manchester, 26-29 March, pp 240-246, 1992.

[26] R. Fletcher, Practical Methods of Optimisation, 2nd edition, Wiley, 1987.

[27] M. J. Flynn, Some Computer Organisations and Their Effectiveness, IEEE Trans. Comput., Vol C-21, pp948-960, 1972.

[28] G. B. Folland, Introduction to Partial Differential Equations, Princeton University Press, New Jersey, 1976.

[29] L. A. Geddes, C. P. DaCosta and G. Wise, The Impedance of Stainless-Steel Electrodes, Med. Biol. Eng., vol 14, pp511-512, 1971.

[30] D. B. Geselowitz, An Application of Electrocardigraphic Lead Theory to Impedance Plethsmography, IEEE Trans. Biomed. Eng., BME-18, pp 38-41, 1971.

[31] D. G. Gisser, D. Isaacson and J. C. Newell, Current Topics in Impedance Imaging, Clin. Phys. Physiol. Meas.,Vol 8, Suppl. A, pp 39-46, 1987.

[32] D. G. Gisser, D. Isaacson and J. C. Newell, Electric Current Computed Tomography and Eigenvalues, SIAM Journal on Applied Mathematics.,Vol 50, No. 6, pp 1623-1634, Dec 1990.

[33] G. H. Golub and C Van Loan, Matrix Computations, North Oxford Academic, 1983.

[34] C. Greenough and K. Robinson, Finite Element Library, Level 1 Documentation, SERC Rutherford-Appleton Lab., 1981.

[35] C. Greenough and C. J. Hunt, PARFEL-An extension of the NAG/SERC Finite Element Library for Multi-Processor Message Passing Systems, SERC Rutherford-Appleton Lab. report RAL-90-070, 1990.

[36] H. Griffiths, The Importance of Phase Measurement in Electrical Impedance Tomography, Phys. Med. Biol.,Vol 32, No 11, pp 1435-1444, 1987.

[37] H. Griffiths, A Phantom for Electrical Impedance Tomography, Clin. Phys. Physiol. Meas.,Vol 9, Suppl. A, pp 15-20, 1988.

[38] H. Griffiths, H. T. L. Leung and R. J. Williams, Imaging the Complex Impedance of the Thorax, Clin. Phys. Physiol. Med. Biol.,Vol 13 Suppl. A, pp 77-81, 1992.

[39] C. W. Groetsch, The Theory of Tikhonov Regularisation for Fredholm Equations of the First Kind, Research Notes in Mathematics, Pitman Advanced Publishing Program, London, 1984.

[40] D. G. Hays, I. A. Gregory and M. S. Beck, Velocity Profile Measurements in Two-Phase Flows, Process Tomography A Strategy For Industrial Exploitation, Report from the European Concerted Action on Process Tomography, Manchester, 26-29 March, pp 319-330, 1992.

[41] M. T. Heath, Esmond Ng and B. W. Peyton, Parallel Algorithms for Sparse Linear Systems, SIAM Review, Vol 33, No 3, pp 420-460, 1991.

[42] M. D. Hebden, An Algorithm for Minimization Using Exact Second Derivatives, Atomic Energy Research Establishment, report TP515, Harwell, 1973.

[43] R. A. Horn and C. R. Johnson, Matrix Analysis, Cambridge University Press, Cambridge, 1985.

[44] D. Isaacson, Distinguishability of Conductivities by Electric Current Computed Tomography, IEEE Transactions on Medical Imaging, Vol MI-5, No. 2, pp 91, 1986.

[45] The Transputer Databook, Inmos document number 72 TRN 203 00, 1988.

[46] H. T. L. Leung, Developement of an Electrical Impedance Tomograph for Complex Impedance Images, Ph.D. Thesis, The Polytechnic of Wales, 1991.

[47] Y. Kim, J. G. Webster and W. J. Tomkins, Electrical Impedance Imaging of the Thorax, J. Microwave Power, Vol 18, pp245-257, 1983.

[48] Y. Kim, H. W. Woo, A Prototype System and Reconstruction Algorithms for Electrical Impedance Technique in Body Imaging, Clin. Phys. Physiol. Meas., Vol 8 Suppl. A, pp63-67, 1987.

[49] K. Levenburg, A Method for the Solution of Certain Non-Linear Problems in Least Squares, Q. J. Appl. Math., 2, pp164-168, 1944.

[50] Lies, Damned Lies and Benchmarks, The Transputer Applications Notebook Systems and Performance, Inmos document number 72 TRN 205 00, pp 258-278, 1989.

[51] E. J. Lindley, B. H. Brown, D. C. Barber, D Grundy, R. Knowles, F. J. McArdle and A. J. Wilson, Monitoring Body Fluid Distribution in Microgravity using Impedance Tomography (APT), Clin. Phys. Physiol. Meas., Vol 13, Supplement A, pp 181-184, 1992.

[52] W. R. B. Lionheart, Complete Non-redundant Sets of Bipolar Measurements for Pair Drive Electrical Impedance Tomography Systems, in press.

[53] E. A. Lipitakis, Solving Eliptic Boundary Value Problems on Parallel Processors by Approximate Inverse Matrix Semi-Direct Methods Based on the Multiple Explicit Jacobi Iteration, Comp. and Maths. with Appls., Vol 10, No. 2, pp171-184, 1984.

[54] E. A. Lipitakis, Explicit Semi-Direct Methods based on Approximate Inverse Matrix Techniques for Solving Boundary Value Problems on Parallel-Processors, Mathematics and Computers in Simulation, 29, North Holland, pp1-17, 1987.

[55] F. A. Lootsma and K. M. Ragsdell, Parallel Nonlinear Optimisation, Parallel Computing, No. 6, pp133-152, 1988.

[56] S. H. Lui and T Kaplan, Theory of AC Responce of Rough Surfaces, in Fractals in Physics, editors L. Pietronero and E. Tosatti, Elsevier Science Publishers B. V., 1986.

[57] D. Marquardt, An Algorithm for the Least Squares Estimation of Nonlinear Parameters, SIAM J. Appl. Math., vol 11, No. 2, pp431-441, 1963.

[58] J. A. Meijerink and H. A. van der Vorst, An Iterative Solution Method for Linear Systems of which the Coefficient Matrix is a Symmetric M-Matrix, Mathematics of Computation, Vol 31, No. 137, pp148-162, January 1977.

[59] J. W. H. Meijs, O.W. Weier, M. J. Peters and A. van Oosterom, On the Numerical Accuracy of the Boundary Element Method, IEEE Transactions on Biomedical Engineering, Vol 36, No. 10, pp1038-1049, 1988.

[60] L. M. Milne-Thomson, Theoretical Hydrodynamics, Macmillon and Co. Ltd, 1962.

[61] V. A. Morozov, Methods for Solving Incorrectly Posed Problems, Springer, Berlin, 1984.

[62] D. Murphy, The Oxford Polytechnic Adaptive Current Tomograph, Department of Computing and Mathematical Sciences Research Report No 14, Oxford Polytechnic, 1988.

[63] A. I. Nachman, Reconstructions From Boundary Measurements, Annals. of Mathematics, Vol 128, pp 539-552, 1988.

[64] J. C. Newell, G. D. Gisser and D. Isaacson, An Electric Current Tomograph,IEEE Transactions on Biomedical Engineering, vol 35, No. 10, October , 1988.

[65] T. F. Oostendorp and A. van Oosterom, The Potential Distribution Generated by Surface Electrodes in Inhomogeneous Volume Conductors of Arbitrary Shape, IEEE Trans. on Biomed. Eng., No. 38, pp 409-417, 1990.

[66] A. van Oosterom and J. Strackee, Computing the Lead Field of Electrodes with Axial Symmetry, Med. and Biol. Eng. and Comput., No. 21, pp 473-481, 1983.

[67] K. S. Paulson, Solving Symetric Matrix Problems on Rings of Transputers, Rutherford Appleton Laboratory Report, 1990.

[68] K. S. Paulson, W. R. Breckon and M. K. Pidcock, The Importance of Electrode Modelling in Electrical Impedance Tomography, Proc. of EC COMAC-BME Workshop on EIT, Copenhagen, pp 84-96, 1990.

[69] K. S. Paulson, W. R. Breckon and M. K. Pidcock, Concurrent EIT Reconstruction, Proc. of EC COMAC-BME Workshop on EIT, Copenhagen, pp136-143, 1990.

[70] K. S. Paulson, W. R. Breckon and M. K. Pidcock, Electrode Modelling in Electrical Impedance Tomography, Siam Journal on Applied Mathematics, Vol 52 Issue 4, pp 1012-1022, August 1991.

[71] K. S. Paulson, W. R. Breckon and M. K. Pidcock, A Hybrid Phantom for EIT, Clinical Physics and Physiological Measurement, Vol 13, Supplement A, pp 155-161, 1992.

[72] K. S. Paulson, W. R. Breckon and M. K. Pidcock, Optimal Experiments in Electrical Impedance Tomopgraphy, Submitted to IEEE Journal on Medical Imaging, 1992.

[73] M. K. Pidcock and K. S. Paulson, Current Density Distributions on Electrodes, submitted to 14th Annual International Conference of the IEEE Engineering in Medicine and Biology, Paris, 1992.

[74] Ping Hua, Modelling and Reconstruction Methods for Electrical Impedance Tomography, Ph.D. thesis, University of Wisconsin-Madison, 1990.

[75] V. Pollok, Computation of the Impedance Characteristic of Metal Electrodes for Biological Investigations, Med. Biol. Eng., vol 12, pp460-464, 1974.

[76] I. N. Sneddon, Mixed Boundary Value Problems in Potential Theory, North-Holland, 1966.

[77] E. Somersalo, M. Cheney, D. Isaacson and E. Isaacson, Layer Stripping: A Direct Numerical Method for Impedance Imaging, Inverse Prob., no. 7, pp 899-926, 1992.

[78] G. Strang and G. J. Fix, An Analysis of the Finite Element Method, Prentice-Hall, Ed. G. Forsythe, 1973.

[79] L. Tarassenko, Electrical Impedance Techniques for the Study of Cerebral Circulation and Cranial Imaging in the New Born, DPhil Thesis, University

of Oxford,1985.

[80] 3L Ltd., Parallel Fortran Users Guide, 3L Ltd., 1990.

[81] 3L Ltd., Tbug Users Guide, 3L Ltd., 1990.

[82] W. F. Tinney and J. W. Walker, Direct Solutions of Sparse Network Equations by Optimally Ordered Triangular Factorisation, Proc. IEEE, 55, pp1801-1809, 1967.

[83] Transtech Devices Ltd, Transtech TMB04 An Expandable Transputer Board for the IBM PC.

[84] R. Wait, Partitioning and Preconditioning of Finite Element Matrices on the DAP., Parallel Computing, 8, pp275-284, 1988.

[85] J. G. Webster, Electrical Impedance Tomography., The Adam Hilger Series on Biomedical Engineering, IOP Publishing Ltd, 1990.

[86] T. J. Yorkey, Comparing Reconstruction Methods for Electrical Impedance Tomography, PhD Dissertation, Dep. Elec. Comput. Eng., Univ. Wisconson, Madison, Aug 1986.

[87] T. J. Yorkey and J. G. Webster, A Comparison of Impedance Tomographic Reconstruction Algorithms, Clin. Phys. Pysiol. Meas., Vol 9, Suppl. A, pp55-62, 1988.

[88] Yoshida H. and Takahashi K., Measurement of Contact Resistance with Microampere Currents, IEEE Transactions on Instrumentation and Measurement, Vol 39, No 5, pp711-714, 1990.

[89] Q. S. Zhu, W. R. Breckon, F. J. Lidgey, C. N. McLeod, K. S. Paulson and M. K. Pidcock, An Adaptive Current Tomograph Using Voltage Sources, IEEE Trans. Biomed. Eng., *in press*, 1992.

[90] Q. S. Zhu, Precision EIT Instrumentation, Ph.D. thesis, Oxford Polytechnic, July 1992.

Appendix

Published Papers

W. R. Breckon, K. S. Paulson and M. K. Pidcock, Parallelism in EIT Reconstruction, Information Processing in Medical Imaging, Ed. D.A. Ortendahl and J. Llacer, Wiley-Liss, pp187-196, 1989.

K. S. Paulson, Solving Symetric Matrix Problems on Rings of Transputers, Rutherford Appleton Laboratory Report, 1990.

K. S. Paulson, W. R. Breckon and M. K. Pidcock, Concurrent EIT Reconstruction, Proc. of EC COMAC-BME Workshop on EIT, Copenhagen, pp136-143, 1990.

K. S. Paulson, W. R. Breckon and M. K. Pidcock, The Importance of Electrode Modelling in Electrical Impedance Tomography, Proc. of EC COMAC-BME Workshop on EIT, Copenhagen, pp 84-96, 1990.

K. S. Paulson, W. R. Breckon and M. K. Pidcock, Electrode Modelling in Electrical Impedance Tomography, Siam Journal on Applied Mathematics, Vol 52 Issue 4, pp 1012-1022, August 1991.

K. S. Paulson, W. R. Breckon and M. K. Pidcock, A Hybrid Phantom for EIT, Clinical Physics and Physiological Measurement, Vol 13, Supplement A, pp 155-161, 1992.

W. R. Breckon, K. S. Paulson and M. K. Pidcock, Iterative Algorithms in Electrical Impedance Tomography, Proceedings of the IEE Conference on Electrical Impedance Tomography, 1992.

K. S. Paulson, W. R. Breckon and M. K. Pidcock, Optimal Measurements in Electrical Impedance Tomography, Proceedings of the 14th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Paris, pp 1730-1731, 1992.

M. K. Pidcock and K. S. Paulson, Current Density Distributions on Electrodes, Proceedings of the 14th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Paris, pp 2386-2387, 1992.

Q. S. Zhu, W. R. Breckon, F. J. Lidgey, C. N. McLeod, K. S. Paulson and M. K. Pidcock, An Adaptive Current Tomograph Using Voltage Sources, accepted for publishing in IEEE Transactions on Biomedical Engineering, February, 1993.

Information Processing in Medical Imaging, pages 187–196

PARALLELISM IN ELECTRICAL IMPEDANCE TOMOGRAPHY.

William R. Breckon, Kevin S. Paulson,
Michael K. Pidcock

Department of Computing and Mathematical Sciences,
Oxford Polytechnic, Oxford OX3 0BP, U.K.

INTRODUCTION

Electrical Impedance Tomography (EIT) is potentially a fast and cheap medical imaging technique. The tomograph consists of an array of electrodes which are attached to the patient, an electronic instrument which is capable of applying a variety of current patterns and measuring the resulting voltages, and a computer capable of calculating the impedance distribution from the measurements. The measurement apparatus requires no moving parts nor expensive magnets and can be made for a few thousand dollars The cost of building our own 32-channel prototype system OXPACT (see Murphy, 1988 and Murphy et al 1989) was around $4000 US). Data collection rates of 24 frames per second (where a frame consists of 104 measurements) have been reported by Brown and Seagar (1987).

It has been shown that the reconstruction problem for EIT is essentially non-linear Breckon and Pidcock(1988)). It is also known that even the linearized problem is not equivalent to the inversion of any generalized Radon transform. This dictates that an accurate solution of the linearised inverse problem requires the repeated solution of a dense system of linear equations. In addition, all reconstruction algorithms require the solution of the forward problem. An iterative algorithm requires this to be done repeatedly. Since currents cannot be constrained to a single plane, the forward problem solver must account for the three dimensional nature of the body. These factors make accurate reconstruction of impedance images computationally expensive.

Recent advances in single-chip microcomputers have brought

near supercomputer performance for the price of a typical desk top computer. These chips such as the Inmos Transputer can be connected together in various topologies to make powerful computing engines for solving numerical problems. Finding parallelism in EIT reconstruction algorithms facilitates dramatic speed increases from using multi-processor machines. This brings the possibility of a low cost, accurate, real time EIT system within the bounds of possibility in the near future.

MATHEMATICAL FORMULATION

Details of the mathematical formulation of the EIT reconstruction problem can be found in Breckon and Pidcock (1988). In this section we briefly summarise only those features relevant to this paper. The *forward problem* consists of solving the partial differential equation

$$\begin{aligned} \nabla\cdot(\sigma\nabla\Phi) &= 0 && \text{in the body} \\ -n\cdot\sigma\nabla\Phi &= j && \text{at the electrodes} \end{aligned}$$

where σ is the conductivity distribution, Φ is the potential and j is the current density. This gives the voltages which we would measure on the electrodes when the current pattern j is applied if the conductivity were σ. This problem can be solved numerically using the finite element method. The *inverse problem* consists of finding a conductivity distribution consistent with a collection of voltage measurements for various applied current patterns. If we collect all the voltage measurements from all applied current patterns into one vector

$$V(\sigma)=(V_1(\sigma),V_2(\sigma),\ldots,V_M(\sigma))^T$$

then we can think of V as the forward mapping. Let σ_0 denote our initial guess for the conductivity and σ_a the actual conductivity. The discrepancy we measure between the actual measurement and our prediction from the finite element solver is then $v = V(\sigma_0)-V(\sigma_a)$. If A denotes the matrix of partial derivatives of V with respect to nodal conductivities ($A=dV(\sigma_0)/d\sigma$) then a first order correction s to σ_0 can be obtained by solving

$$As = v.$$

The matrix A can readily be calculated from the interior potentials Φ (Breckon and Pidcock(1988)).

This linearisation procedure can overcome the non-linearity of the problem. If a crude but fast image is required, σ_0+s can be used. If more accuracy is required the process can be iterated, a new matrix $A = dV(\sigma_1)/d\sigma$ having been calculated at $\sigma_1=\sigma_0+s$. The difficulty remains that the linearised problem is ill-posed and any attempt to solve this system directly will fail. This can be overcome by using regularisation.

REGULARISATION

There are numerous standard techniques for regularisation. It is convenient to divide these into two categories: iterative and direct. In iterative techniques the regularisation is achieved by stopping a standard iterative matrix solver when it has converged to within the accuracy of the measurements (Morozov's stopping criterion (1966)). In direct techniques a modified, well conditioned, version of the matrix is inverted. The modification depends on a parameter called the regularisation parameter whereas in iterative techniques the number of iterations acts as the regularisation parameter.

The most widely known direct technique is Tikhonov regularisation where the system

$$(A^TA + \mu I)s = b$$

is solved. Here $b = A^Tv$ and μ is a suitably chosen regularisation parameter. This new system of equations is well conditioned and can be solved by standard techniques.

Other direct techniques can be derived from the singular value decomposition

$$A=V\Lambda U^T$$

where U and V are the matrices of left and right singular vectors. If the matrix were well conditioned we could solve As=v using the expression

$$s = U\Lambda^{-1}V^Tv.$$

However the essence of the illposedness is that the singular values decrease to zero. A regularised inverse can be obtained by replacing Λ^{-1} by Λ_k^{-1} which has the first k diagonal elem-

ents the same as Λ^{-1} and the remainder replaced by zeros. In this case k acts as a regularisation parameter.

Calculating the singular value decomposition is an expensive process. However if the crude image formed by the first linear step is adequate, a singular value decomposition of $V'(\sigma_0)$ can be stored in advance. This would result in an extremely fast reconstruction technique.

Three well known iterative techniques with interesting regularisation properties are successive approximation, steepest descent and conjugate gradient. In each of these methods, a sequence s_i of successive approximations to the generalised solution is calculated. Define the error r_i by

$$r_i = A^TAs_i - v.$$

The iteration scheme for successive approximation is then

$$s_{i-1} = s_i - \tau r_i$$

where τ is a fixed relaxation parameter. Similarly, the steepest descent algorithm is given by

$$s_{i-1} = s_i - \tau_i r_i$$

but in this case τ_i is given by

$$\tau_i = ||r_i||^2/||Ar_i||^2.$$

The conjugate gradient method is slightly more complicated, being given by

$$s_{i-1} = s_i - \eta_i p_i$$

where

$$p_0 = r_0 = -A^Tv, \quad p_n = r_n + \sigma_{i-1}p_{i-1}$$

$$\eta_{i-1} = \langle r_i, p_i\rangle/||Ap_i||^2, \quad \sigma_{i-1} = -\langle Ar_i, Ap_{i-1}\rangle/||Ap_{i-1}||^2$$

The regularisation properties of these iteration schemes can be understood in terms of singular values (Talenti (1986)).

APPLICATION TO EIT

A variety of these algorithms have been applied to the EIT reconstruction problem. The method of Kim et al (1983) is similar to steepest descent. The methods of Breckon and Pidcock(1988) and of Yorkey (1986) use Tikhonov regularisation followed by a matrix solver such as Choleski factorisation. Murai and Kagawa (1985) used a truncated singular value decomposition.

For execution speed we advocate a filtered singular value decomposition for the first step. Since the singular value decomposition can be stored in advance for an initial guess of the conductivity, the time taken to calculate this is irrelevant. It also has the advantage that a variety of regularisation schemes can be implemented from this data.

For subsequent iterations the forward problem must also be solved again. This also involves the solution of a system of linear equations. Since a very good first guess for the potential Φ is available, iterative techniques provide an extremely fast method. However since solving the linearised problem involves calculating only the conductivity update σ rather than the absolute conductivity, we have no good *a priori* approximation for σ other than zero. In this case direct techniques have a computational advantage. We have yet to determine which combination of techniques yields the best images but we shall see later that both iterative and direct techniques can be efficiently implemented on parallel machines.

CONCURRENT ALGORITHMS

The classical Von Neumann machine, the model on which sequential computers are based, consists of a processing unit together with a list of instructions and a data store. It can only execute one instruction at a time and therefore its speed is limited by the rate at which instructions can be executed. If additional processors are added, possibly each with their own list of instructions, separate instruction streams can be executed concurrently. As additional processors are added the speed is limited, ultimately, only by the rate at which data can be delivered to the processors. This final bound to execution speed can be transcended by giving each processor its own dedicated data store and bus. Flynn (1966) classifies these machines as Multiple Instruction Multiple Data (MIMD).

Concurrent algorithms for MIMD machines can be conceptualised in the framework of the occam model. In this model inde-

pendent, asynchronous processes communicate only via bi-directional channels. Each process can, in turn, be composed of inter-communicating subprocesses. To implement a job on a MIMD machine we must express it in terms of processes according the occam model. This places different constraints on the algorithm design to the Von Neumann model. The latest sequential algorithm is unlikely to be optimal when implemented on a MIMD machine.

If a processor is dedicated to each of N processes then the job could be expected to execute N times faster. This is an ideal which can only be approached. A measure of merit known as efficiency is defined as

$$\text{Efficiency} = T(1)/(N\,T(N))$$

where T(i) is the time taken on i processors. Dependencies between concurrently executing units always exist so time must be spent passing messages between processors. This introduces a delay, not only during the transmission of a message but, more significantly, for the period that a processors is idle waiting for synchronisation. Both these considerations, minimising the amount of data exchanged between processors and balancing the work load to reduce idle periods, are of immense importance when designing concurrent algorithms.

THE INMOS TRANSPUTER

The Inmos Transputer is a hardware realisation of the occam model. It is a single chip computer which includes 4KBytes of fast RAM and an interface to 4 GBytes of external memory. The T800 transputer has a floating point unit capable of 2.5 megaflops. To implement the occam model the transputer has four bidirectional links which can be connected directly to other transputers. These are serial links communicating at 20 MBaud. Concurrently executing sub-processes on the same transputer is mimicked by a hardware scheduler.

Transputers are readily available on extension boards to fit in commonly available micro-computers and work-stations. The host computer provides the support for peripherals and provides an interface with the user. The transputers can be configured (in hardware or software) into arbitrary connection topologies given the limitation of four links per processor.

SOFTWARE TOOLS

The transputer was designed to run programs written in occam, the first concurrent language. Since its design, other languages such as Pascal, C and Fortran have been implemented with parallel extensions to facilitate concurrence. These stand-alone software tools have been integrated into common operating systems on host computers. More recently parallel operating systems have been developed. Due to the immaturity of the area, many basic tools such as concurrent debuggers, do not exist. Those tools which are available are in early stages of development and suffer from many teething problems.

In our own work we have used a Quintek Fast Four which is a standard AT compatible expansion card fitted with four T800 transputers each with 1 MByte of RAM. As a host we use a Sun 80386 based work-station. For reasons of software inertia we chose a parallel Fortran compiler supplied by 3L Ltd.

CONCURRENT EIT RECONSTRUCTION

Some understanding of the complexity of designing reconstruction algorithms to run on concurrent machines can be gained by considering the matrix solution technique of forward substitution. Linear systems of the form $Lx=b$, where L is a triangular matrix, are generated by factorisation techniques such as QR and Choleski decomposition. If L is a lower triangular matrix, $L_{ij}=0$, $\forall\ i,j: i<j$, the system may be solved by the forward substitution formula:

$$x_i = (b_i - \sum_{j=1}^{i-1} L_{ij}x_j)/L_{ii}$$

The recursive nature of this algorithm, each x_i depending upon all the x_i's preceding it, would seem to preclude its division into independent work units. However, once x_i has been calculated, its contribution to all the subsequent x_j's, i.e. $L_{ji}x_i$, $i-1 \leq j \leq n$, may be calculated independently. If the matrix L and the vectors x and b are distributed by row among the available processors then each will be able to calculate a subset of the $L_{ji}x_i$ factors concurrently. To be able to do this each processor would need to receive a message containing the value of the latest x_i calculated. The data transmitted is small compared with the amount of computation so if the affected rows are distributed evenly among all the processors a proport-

linear speed up can be expected. There are many ways of distributing the data, each with its own data flow characteristics and overall efficiency. The simplest distribution, sending contiguous rows to each processor, results in the minimum of message passing and yet the maximum idle time, and an efficiency of only 50%. The optimum distribution is accomplished by "dealing" the rows out to the processors in the same fashion that a card player distributes cards among the players. For each row, the subsequent rows are distributed as evenly as possible among the processors. The processors are idle for, at most, a single floating point operation before synchronisation. This demonstrates the common trade off between the complexity of data storage and the efficiency of concurrent programs. Considerations that are irrelevant in the optimisation of sequential programs become paramount in a parallel environment.

Forward and backward substitution are important steps in the solution of systems of linear equations after factorisation. We have implemented a variety of matrix solution techniques on a ring of four transputers, including QR factorisation and the iterative, conjugate gradient method. QR factorisation is an example of a method that uses highly parallel but out-of-date algorithms rather than the more serially efficient algorithms in current use. In sequential machines QR factorisation using Given's rotations has been largely superceded by the computationally more efficient Householder transformations. However, a concurrent machine can execute independent Given's rotations in parallel and this method becomes preferable once more. These two techniques are quite different in data distribution and the flow of messages during execution but both have been implemented with acceptable efficiency (see Table 1). The optimum algorithm depends upon the characteristics of the matrices being solved. Either of these techniques could be used for ECT reconstruction.

number of processors	QR factorisation Speed-up	Efficiency	Conjugate Gradient Speed-up	Efficiency
1	1.0	100%	1.0	100%
2	1.6	80%	1.9	94%
3	2.2	73%	2.7	91%
4	2.7	68%	3.6	90%

Table 1. QR factorisation versus Conjugate Gradient Method.

CONCLUSIONS

Matrix solution algorithms are an integral part of EIT image reconstruction. The single stage method reconstructs an image from a set of measurements by the application of a single linear transformation. This transformation may be factorised into a diagonal matrix of filtered singular values and two orthonormal matrices. The solution of such a system is easily and efficiently parallelisable with a minimum of communication and good load balancing. Under these conditions we can expect almost linear increases in the speed of reconstruction as the number of processors is increased. Real time reconstruction, of the order of 25 frames per second, is certainly possible with images of resolution currently being used. Two or three Inmos T800 transputers would be sufficient for the inversion of 100 measurements, in real time. With more higher resolution images could be formed. With sufficiently large networks, nonlinear, iterative, reconstruction techniques could be used to produce accurate, high resolution images at acceptable rates.

ACKNOWLEDGMENTS

We would like to thank the SERC/DTI Transputer Initiative for the loan of the equipment used in this work.

REFERENCES

Breckon WR, Pidcock MK (1988). Ill-posedness and non-linearity in Electrical Impedance Tomography. In de Graaf CN and Viergever MA (eds): "Information Processing in Medical Imaging" New York: Plenum, pp 235-244.

Brown BH, Seagar AD (1987) The Sheffield data collection system. Clin Phys Physiol Meas.8 Suppl.A:91-97.

Flynn MJ (1966). Very high speed computing systems. Proc. IEEE. 54:1901-1909

Kim Y, Webster JG, Tompkins WJ (1983). Electrical Impedance Imaging of the Thorax. J. Microwave Power vol 18:245-257.

Morozov VA (1966). On the solution of functional equations by the method of regularisation. Soviet Math. Dokl. 7:414

Murai,T K.and Kagawa, Y.,(1985) Electrical Impedance Computed Tomography based on a finite element model. IEEE Trans. Biomed. Eng., vol BME-32,177-184

Murphy D (1988). Research report, Department of Computing and Mathematical Sciences, Oxford Polytechnic, Oxford OX3 0BP.

Murphy D, Lidgey J, Davey-Winter TGE, Breckon WR, McLeod.C (1989). A multiple programable current source impedance tomograph. To be published in proceedings of the. 2nd IFMBE Pan-Pacific Symposium, Melbourne.

Talenti G (1987). (ed) Inverse Problems. Lecture notes in Mathematics, vol 1225, Springer, Berlin.

Yorkey TJ (1986) Comparing Reconstruction Methods for Electrical Impedance Tomography PhD Thesis, University of Wisconsin, Madison,

Solving Symmetric Matrix Problems on Rings of Transputers.

Kevin Paulson.

1/12/89.

Introduction.

The matrix equation. A.x = b. often requires transformation to yield a unique solution vector x. If A is the coefficient matrix of an over-determined. but not necessarily consistent, set of simultaneous equations then :

$$x = (A^T.A)^{-1}.A^T.b$$

yields the least squares solution. An under-determined set of equations has an infinite number of solutions but the equation:

$$x = A^T.(A.A^T)^{-1}.b$$

finds the solution with the minimum L_2 norm. Both these two special cases of the use of the Moore-Penrose inverse of the matrix A require the solution of a symmetric, and often dense, set of simultaneous equations. This paper will assume the former case yet the results are equally applicable to the latter.

Many techniques exist to solve such a system and they can be classified into direct or iterative methods. Direct methods produce a result in a finite number of steps while iterative methods produce an infinite series of vectors which approach the exact solution to arbitrary precision. Generally, direct methods require $O(N^3)$ operations while iterative methods require $O(N^2)$ operations per iteration. Clearly an iterative method must converge before some fraction of N iterations for it to be competitive. The optimum method to use on a sequential, single processor machine, is determined by the ill-posedness of the problem, the accuracy required in the result and the sparseness of the matrix. In general direct methods require fewer operations to solve dense matrices while iterative methods can produce results faster with ordered, sparse matrices.

On a multi-processor, distributed memory, machine the situation is more complex. The execution speed of an algorithm is less strongly determined by the number of algebraic operations required than by the communication overhead moving data between units of the distributed memory. Much has been written on optimal algorithms for hypothetical machines with unlimited numbers of processors and shared memory, or small communications overhead; see [1&2]. These results have little relevance to the small networks of transputers becoming

available. Transputer boards, typically with less than twenty processors, and relatively slow communications force quite different constraints on the user. Designing optimal algorithms on these networks is considerably more complex than the similar task on a single processor, or a massively parallel machine. This is compounded by the present lack of standard library software and experience of parallel algorithms.

Transputer Based Systems.

Transputer based systems are developing in several directions demanding software design with different constraints. The high profile end of the spectrum is large and expanding arrays such as the Edinburgh hypercube with hundreds or thousands of processors. On large networks, communication costs are of paramount importance and complex topologies and routing algorithms need to be used. Such a machine will be used on very large problems, with long run times, justifying the redistribution of data and code in between algorithm stages. These are the design considerations leading to the NAG parallel library. At the other end of the spectrum is the parallel work station which may have only a handful of transputers. The success of these machines hinge on the development of automatic parallelising compilers and hardware routing of communications. Lying between these two extremes are instrumentation and embedded industrial control equipment. These systems use transputers for their speed and the ability to handle large amounts of data in the distributed memory. Such systems, generally with tens of transputers, will never aim to be general purpose computers and will often run a single program during most of their lifetime. The small diameter of such networks and the specific communications demands of the algorithms that run on them may simplify the communications problem considerably.

Linear algebra algorithms, in particular, can usually be distributed so that a large proportion of the communications occurs between directly connected neighbours. In this situation the simple, unidirectional ring configuration is not handicapped by the relatively large diameter of this topology. It also avoids a lot of the overhead inherent in more complex routing schemes. Not only is there minimal routing logic but intermediary processors are not interrupted to route messages not intended for them. The unidirectionality of the ring removes many of the dead lock problems caused by races between messages to the same destination processor. Messages are received in the same order as they are sent and so synchronisation is simplified and message

header length is minimised. Further more a ring is easily mapped onto other topologies so increasing the portability of algorithms designed for rings. It retains the advantages of topologies with high symmetry, such as binary trees and cubes, without the restriction on the number of processors.

It is relatively simple to implement a ring communication system consisting of a task running on each transputer connected by channels linking the transputers. Message headers include the destination transputer identifier and messages are passed around the ring until the correct transputer is reached. This system allows communications for more complex than nearest-neighbour while adding the minimum demand on transputer resources. Direct communication between numerical calculation tasks is undesirable as it forces synchronisation invariably causing a descheduling of one process while the other catches up. If further useful work can be accomplished after data transmission then the buffer communication process allows calculations to continue.

Solution Techniques.

The solution technique employed on the system:

$$(A^T.A).x = b$$

is determined, in part, by the starting point. The matrix multiplication, $A^T.A$, is computationally expensive and so should not be performed as a step in the solution, but in some circumstances only this product is available. The classical method of solution of symmetric, positive definite, matrices is via Choleski factorisation into the form $L.L^T$ where L is a lower triangular matrix. The system can then be solved via forward and backward substitution in the two steps:

$$L.y = b$$
$$L^T.x = y.$$

If, however, the matrix A is the starting point it may be factorised into an ortho-normal matrix Q and an upper triangular matrix R. The system can then be expressed as the product of two triangular matrices as before and solved the same way.

$$A^T.A = (Q.R)^T.Q.R = R^T.Q^T.Q.R = R^T.R.$$

These are the best, direct, algorithms for dense matrices, in terms of operation count, and so will also be optimal on a parallel system if they can be implemented efficiently.

Iterative methods can be classified into stationary and non-stationary processes. Stationary processes, such as Jacobi, Gauss-Seidel and SOR [2], can be expressed by the iterative formula:

$$x_{n+1} = C.x_n + D$$

where C and D are a constant matrix and vector dependent upon the coefficient matrix. Each method has its own convergence criteria based on the singular values of the coefficient matrix. Jacobi, Gauss-Seidel and SOR all require the absolute value of the singular values to be bounded by a known constant. For a general matrix this is an impractical test to perform and so convergence is not guaranteed. Some non-stationary methods, such as the method of Kaczmarz [5] and steepest descent, are guaranteed to converge for consistent systems. The rate of convergence is determined by the condition number of the coefficient matrix, defined as the ratio of the largest and smallest singular values. The convergence of highly ill-posed systems is slow and bounded by the sensitivity to errors in numerical calculations.

The method of conjugate gradients is classified as an iterative method although, for calculations without error on full rank, positive definite matrices, it converges within N iterations. The solution of highly ill-posed problems meets the same bound as other numerical procedures and satisfactory convergence may require many more iterations. The situation can be improved by pre-conditioning the equation with the pre-multiplication of each side by an approximate inverse matrix, so decreasing the effective condition number. Algorithms exist for calculating the conjugate gradient iterations, without performing the operationally expensive matrix multiplication. The guaranteed and relatively swift convergence of this method has made it the standard for many applications.

Implementation.

The steps in all these methods are basic linear algebra operations (BLA's) such as matrix-vector multiplications and scalar products. Methods of automated parallelisation, such as the farmer-worker model, pass data packets to processors to perform BLA's and return a result to a driver process. Despite the attractiveness of these methods they are inappropriate to Transputers due to the small computation to data ratio for these BLA's. It is faster to perform the operations on a single Transputer than to pass the data out to the network and receive a result back. Fortunately, for higher level BLAs many matrix-

vector multiplications are performed with the same matrix so these data need to be distributed only once. A matrix-vector multiplication can be performed by distributing rows of the matrix among the processors along with a copy of the complete vector. Each processor has the data available to calculate row elements of the result vector which may then need to be collected. If further matrix multiplications are to be performed on the result vector then a copy needs to be made on each processor. In a ring configuration, a rotation of the data involves each processor passing a message containing vector row elements to the processor to its left, and receiving similar information from the processor to its right. The complete result vector can be built by each processor after n-1 rotations, where n is the number of processors in the ring. If data cannot be transmitted and received concurrently by a processor then a ring configuration can achieve this complete exchange of data in the same time as a completely connected system. In parallel Fortran it is difficult to implement concurrent data transmission and for short messages the overhead of initiating a concurrent thread would erode any advantage gained.

The conjugate gradient method, QR and Choleski factorisation have been implemented using the ring buffer system described above. The number of processors in the ring can be incorporated as a global parameter and so only a single declaration needs to be changed to produce source for a ring of arbitrary size. The program running on each transputer is identical although it includes branching dependent upon the transputer's position in the ring and the data stored on it. Only one program is written and copies are placed on each transputer. Standard matrix-matrix and matrix-vector routines can then be used within the program to act upon the subset of data stored on each transputer. Subroutines need to be written for rotations of scalar and vector data to perform sumations after dot products or to rebuild result vectors after distributed matrix-vector products. The conjugate gradient algorithm, consisting only of two simple linear algebra operations and data rotations, can be written elegantly in terms of these subroutines.

QR factorisation is more complex. The upper triangular matrix R is constructed from the coefficient matrix A by zeroing elements below the leading diagonal. The ortho-normal matrix, Q, does not need to be constructed as it is eliminated from the equations. Two QR factorisation algorithms are in common use; Given's rotations zero single elements by performing rotations

between rows of the matrix and Householder transformations that zero entire columns below the leading diagonal by operations between columns. The floating point operation count and the amount of data that needs to be transmitted for common data distributions, is similar for these two algorithms yet Householder transformations prove superior due to the economies of passing fewer but longer messages. Both QR and Choleski factorisation require operations between the column being considered and all columns to the right of it. To distribute the calculations evenly and to minimise data transmission, the coefficient matrix needs to be distributed by column wrapping. If a matrix of N columns is to be distributed among p processors, numbered 0 to p-1, then column m is stored on processor MOD(m,p). This ensures that the work involved with each stage is spread as evenly as possible among all the processors. Romine and Ortega [4] describe implementations of forward and backward substitution on column wrapped data.

The test.

The following tests were conducted on a Quintek Fast Four Transputer board mounted in a Sun 386i work station. The board has four Inmos T800 transputers each with a megabyte of memory. 3L parallel Fortran using double precision arithmetic was used for all the software. A test matrix of size 300*208 was constructed with a condition number of 112 along with a consistent right hand side. The singular values fall off exponentially from 1 to 0.0089. The matrix is quite ill-conditioned and hence sensitive to numerical errors during calculations. Similar singular value distributions are often encountered in ill-posed inverse problems. The symmetric system, $A^T.A.x = b$, was solved using QR factorisation, Choleski factorisation and the conjugate gradient method. The QR factorisation works directly on the matrix A while the conjugate gradient method and Choleski factorisation start with the matrix $A^T.A$. The times given below are only for the actual calculation and not the initial data loading as this time is highly host system dependent and, where these routines are part of a more complex algorithm, irrelevant.

QR Factorisation

no of processors	execution time (s)	speed up	efficiency
1	81	1.0	100%
2	43	1.8	94%
3	29	2.8	93%
4	22	3.7	92%

Choleski Factorisation

no of processors	execution time (s)	speed up	efficiency
1	11.1	1.0	100%
2	5.9	1.9	94%
3	4.2	2.7	92%
4	3.1	3.6	90%

Conjugate Gradient Iteration.

no of processors	no of iterations	execution time (s)	speed up	efficiency
1	1000	700	1.0	100%
2	1000	372	1.9	94%
3	1000	256	2.7	91%
4	1000	193	3.6	90%

In these tests the execution time on a single processor is for an optimised, serial version of the same algorithm implemented in a single task. As soon as more than one transputer is used extra tasks controlling the communications between processors need to be added. It is the overhead associated with these added tasks that is responsible for most of the drop in efficiency between using one and two processors. The similarity of the efficiencies of these quite different algorithms is striking and due to the preponderance of nearest-neighbour communications in their implementations.

With this test matrix QR and Choleski factorisation yield results of similar precision. Using the identity as a preconditioner the conjugate gradient method required 440

iterations and 90 seconds to gain the same precision. An approximate preconditioner was constructed with singular values increasing linearly from 1 to 112 as an approximation to the true inverse with singular values increasing exponentially over the same range. Only 60 iterations were required to reach the desired precision using this preconditioner. Clearly the availability and effectiveness of a preconditioner is a vital consideration if the conjugate gradient method is to be considered on problems of this sort.

Conclusions.

Many linear algebra algorithms can be distributed in such a fashion that most communications are between nearest-neighbours. In this situation a ring configuration has the advantages of fast execution, due to low routing overhead, and rapid implementation due to the simplicity of communications preventing dead-locks and races. The greatest disadvantage of a ring topology, its diameter, is irrelevant as very few communications span the network. Finding optimal algorithms for a ring topology is simplified by the knowledge that diverse algorithms can be implemented with similar efficiencies so optimal serial algorithms are also optimal parallel algorithms.

References.

[1] Parallel Algorithms and Matrix Computation. J. J. Modi, Clarendon Press, Oxford, 1988.
[2] Parallel and Distributed Computing. D. P. Bertsekas and J. N. Tsitsiklis, Prentice-Hall Int., 1989
[3] Transputer Reference Manual. Inmos, Prentice-Hall Int., 1988
[4] Complexity of a Dense-Linear-System Solution on a Multi-processor Ring. I. C. F. Ipsen, Y. Saad and M.H. Schultz; Linear Algebra and its Applications, 77:205-239, 1986
[5] The Mathematics of C.T. F. Naffever, Wiley, 1986.

SOME PARTS EXCLUDED UNDER INSTRUCTION FROM THE UNIVERSITY