A GEOMETRIC APPROACH TO THREE-DIMENSIONAL DISCRETE ELECTRICAL IMPEDANCE TOMOGRAPHY

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Engineering and Physical Sciences

2015

Russell Miller
School of Mathematics

Contents

Word count 24638

List of Tables

List of Figures

The University of Manchester

Russell Miller
Doctor of Philosophy
A Geometric Approach to Three-Dimensional Discrete Electrical Impedance Tomography
February 2, 2015

Electrical impedance tomography (EIT) is an imaging modality with many possible practical applications. It is mainly used for geophysical applications, for which it is called electrical resistivity tomography. There have also been many proposed medical applications such as respiratory monitoring and breast tumour screening.

Although there have been many uniqueness and stability results published over the last few decades, most of the results are in the context of the theoretical continuous problem. In practice however, we almost always have to solve a discretised problem for which very few theoretical results exist. In this thesis we aim to bridge the gap between the continuous and discrete problems.

The first problem we solve is the three-dimensional triangulation problem of uniquely embedding a tetrahedral mesh in $\mathbb{R}^3$. We parameterise the problem in terms of dihedral angles and we provide a constructive procedure for identifying the independent angles and the independent set of constraints that the dependent angles must satisfy. We then use the implicit function theorem to prove that the embedding is locally unique. We also present a numerical example to illustrate that the result works in practice. Without the understanding of the geometric constraints involved in embedding a three-dimensional triangulation, we cannot solve more complex problems involving embeddings of finite element meshes.

We next investigate the discrete EIT problem for anisotropic conductivity. It is well known that the entries of the finite element system matrix for piecewise linear potential and piecewise constant conductivity are equivalent to conductance values of resistors defined on the edges of the finite element mesh. We attempt to tackle the problem of embedding a finite element mesh in $\mathbb{R}^3$, such that it is consistent with some known edge conductance values.

It is a well known result that for the anisotropic conductivity problem, the boundary data is invariant under diffeomorphisms that fix the boundary. Before investigating this effect on the discrete case, we define the linear map from conductivities to edge conductances and investigate the injectivity of this map for a simplistic example. This provides an illustrative example of how a poor choice of finite element mesh can result in a non-unique solution to the discrete inverse problem of EIT. We then extend the investigation to finding interior vertex positions and conductivity distributions that are consistent with the known edge conductances. The results show that if the total number of interior vertex coordinates and anisotropic conductivity variables is larger than the number of edges in the mesh, then there exist discrete diffeomorphisms that perturb the vertices and conductivities such that no change in the edge conductances is observed. We also show that the non-uniqueness caused by the non-injectivity of the linear map has a larger effect than the non-uniqueness caused by diffeomorphism invariance.

Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made **only** in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487), in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University's Policy on Presentation of Theses.

Acknowledgements

Firstly, I would like to thank my supervisor, Bill Lionheart, for introducing me to the problem of EIT and for being a constant source of interesting, new ideas. His guidance and support throughout the project is much appreciated. I would also like to thank EPSRC for providing the funding for my doctoral training programme. Thanks also to my office colleagues and other members of the inverse problems group: Francis, Kyriakos, Sophia, Will, Nicola, Naeem, Henry and Taufiq. I would like to give special thanks to Mike Crabb for providing useful comments and being there to discuss some of the more difficult concepts I have encountered throughout the project.

I would like to thank my Mum and Dad for always having questions about my work and supporting my decision to come back to university. I must give a special thank you to Diane and Ian for welcoming me into their home, and to Toffee the Golden Retriever for being a constant source of cheer. Our lunchtime walks resulted in many of my better ideas. Last but certainly not least, I would like to thank my fiancée Nicola for firstly encouraging me to return to university and for being a constant source of love and support throughout.

Chapter 1

Introduction

1.1 An introduction to EIT

Electrical impedance tomography (EIT) is an imaging modality in which electrical current is applied to the boundary of an object and the resulting voltage is measured on the boundary via surface electrodes. From this information we attempt to infer the electrical conductivity inside the object. Since different materials have different electrical properties, the conductivity distribution can help to determine the distribution of materials in the interior of an object without requiring physical access to the interior.

The idea of using electrical measurements to determine the conductivity distribution within an object was first conceived by Conrad Schlumberger in 1912, when he attempted the first electrical field survey [29]. Since then, there have been many publications on the use of electrical imaging for geophysical applications, see for example, [40], [16], [76] and [96]. When applied to geophysical problems the method is known as electrical resistivity tomography (ERT) since the electrical resistivity, the reciprocal of conductivity, is reconstructed, although the mathematical problem is the same as in EIT.

The idea was introduced to the medical community by Barber and Brown in 1982, [23]. Since the introduction by Barber and Brown, attempts have been made to apply EIT to medical problems such as brain imaging for stroke patients [84], respiratory monitoring of intensive care patients [3], breast tumour screening [32], [45], and prostate screening [95].

The mathematical problem of EIT (see section 2.1 for details) is ill-posed in the sense that the interior conductivity does not depend continuously on the electrical data measured, which means that large changes in the conductivity result in small, sometimes indistinguishable, changes in the resulting voltage data [56]. When combined with the fact that only finitely many electrodes can be applied to the object, the ill-posed nature of EIT leads to poor spatial resolution. Due to the difference in conductivity of different biological tissues [15] and rock types [38], EIT could lead to high contrast images, which may be of greater importance than spatial resolution in some practical applications. Although EIT has poor spatial resolution, it has the ability to be a cheap alternative to other imaging modalities such as X-ray CT and MRI when spatial resolution is not of primary importance.

We can divide the mathematical problem into two sub-problems: the so called *forward problem*, which is the problem of computing the electrical potential given the conductivity distribution; and the *inverse problem*, which is the problem of determining the conductivity given the boundary measurements of the potential. The forward problem is governed by a partial differential equation which will be introduced in section 2.1, which must be solved for a given conductivity. The inverse problem is then usually tackled as an optimisation problem in which we attempt to fit the modelled boundary data to that given by the real measurements by perturbing the conductivity in our model.

1.2 Thesis Objectives

Much of the mathematical literature for EIT is devoted to the uniqueness and stability of the continuous problem, which is the problem of finding a conductivity distribution in a continuous medium with infinitely many measurements, (see section 2.1 for a review of results that have been proved in this area). In practice however, we must find a numerical approximation to both the forward and inverse problems. The most common technique for solving the forward problem is to use the finite element method (see section 2.2), which involves partitioning the domain into polyhedra, then solving the problem for function values at fixed positions, which are then interpolated on the rest of the domain. The finite element formulation results in a linear system of

equations that must be solved for the approximate potential. It is well known that for a piecewise linear approximation of potential and piecewise constant approximation of conductivity, the entries of the finite element system matrix are equivalent to conductances of resistors in a resistor network (see section 4.2). In this thesis we will consider the problem of finding a three-dimensional finite element discretisation and choice of conductivity that is consistent with the conductance values of resistors in a given resistor network. It is important that we gain an understanding of this problem in order to establish the possible conductivity distributions we can hope to reconstruct in real situations, where a numerical approximation to the full problem is necessary.

1.3 Thesis Organisation

In chapter 2 we first define the mathematical problem of EIT and review the main results that have been proved about the uniqueness and stability of the problem. We then introduce the forward problem and the finite element method. Next we give an overview of some theory that we will use from the field of differential geometry. Although our work will not explicitly use many of the definitions in this section, many of the important definitions are dependent on earlier ones so we try to give a self-contained treatment of the theory we need. In addition, an understanding of the theory in this section is important for the problem of recovering anisotropic conductivity and also for understanding some of the tools we will use in a discrete setting. We then give an introduction to discrete differential geometry and some tools from algebraic topology that we require. We introduce some basic graph theory next, which we link in with the previous topic when possible. This will be important for understanding resistor networks and for the geometric problem we will solve in chapter 3. The final section of chapter 2 gives an overview of the singular value decomposition, which we will use to gain an insight into the rank deficiency of systems of equations that arise in our problems.

In chapter 3 we define the geometric constraints that the dihedral angles of a tetrahedral mesh must satisfy in order for it to be embedded in three-dimensional Euclidean space. We prove that given a subset of fixed dihedral angles and a set of independent geometric constraints, the full set of dihedral angles that satisfy the

constraints uniquely determine the vertex positions of the tetrahedra up to similarity. We also provide a constructive procedure for finding the independent constraints and for finding the vertex positions given the dihedral angles that satisfy the constraints. We provide a numerical example to illustrate the procedure. This problem is important because without knowledge of the geometric constraints that the angles, and therefore the vertex positions satisfy, we cannot attempt the problem of finding a finite element mesh that is both embeddable in $\mathbb{R}^3$ and consistent with a resistor network.

In chapter 4 we first give an introduction to resistor networks and review previous work that has attempted to solve the EIT problem in the context of resistor networks. We then describe the linear mapping that maps conductivities to edge conductances for various types of anisotropic conductivities, given a fixed finite element mesh. We give results of numerical experiments that aim to tackle the problem of finding a finite element mesh and anisotropic conductivity that is consistent with conductance values of a resistor network. The reason for tackling this problem is that in the finite element formulation, the potential only depends on the conductivity through the edge conductance values. This means that the inverse problem of finding a conductivity distribution is actually a two-step process of first finding edge conductances that are consistent with the data, and then finding a consistent embedding of a simplicial complex in $\mathbb{R}^3$, along with a conductivity distribution defined on the mesh that is consistent with the conductance values. Our work assumes that we have already solved the first step by finding the conductances that fit some given boundary measurements in the hope that we can define constraints on the conductivity by utilising a priori information, to ensure that the problem of recovering the vertex positions and conductivities has a (locally) unique solution.

Chapter 2

Mathematical background

In this chapter we will describe the mathematical formulation of the EIT problem and review some important results regarding uniqueness and stability of the problem. We then introduce the finite element method in the context of EIT which we will use to numerically solve the forward problem, which we will define in section 2.2. Following this we give an introduction to some theory from differential geometry, algebraic topology, graph theory and linear algebra, which we will use to set up and solve problems related to EIT in later chapters.

We note that none of the work in this chapter is novel and is included to inform the reader of the relevant background mathematics that is necessary in order to understand the work of later chapters.

2.1 The inverse conductivity problem

2.1.1 Problem overview

In this section we describe the mathematical problem underlying practical EIT problems. We then review the main results regarding uniqueness and stability of the problem. The problem was introduced to the mathematical community by Alberto Calderón in his 1980 seminal paper "On an inverse boundary value problem" [27]. Since then the problem has received wide attention and the following description of the mathematical problem of EIT can be found in many publications, for example [92], [4].

Before proceeding with the problem definition we would like to introduce the *Einstein summation convention* since it is very helpful in ensuring that correct indices are used when we come to more complicated concepts involving vectors and tensors. It is applied by summing over any indices that appear as both upper and lower indices in a single expression. As an example, we can write the EIT operator - which we will formally introduce in a moment - for anisotropic conductivity σ^{ij} as

$$L_\sigma = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial}{\partial x^i}\sigma^{ij}\frac{\partial u}{\partial x^j} = \frac{\partial}{\partial x^i}\sigma^{ij}\frac{\partial}{\partial x^j},$$

since both indices i and j appear as upper indices (in the conductivity tensor) and lower indices (as partial derivatives with respect to each coordinate direction). The upper indexing notation used for σ will become clear later in this chapter. From now on if an index appears as both upper and lower indices then we assume the Einstein convention is applied and the expression represents a sum over this index.

Let $\Omega \subset \mathbb{R}^n$ be a bounded domain with $n \geq 2$ and let the anisotropic conductivity be represented by the symmetric positive definite matrix-valued function $\sigma = (\sigma^{ij})$. Let $H^k(\Omega)$ be the Sobolev space for integer $k \geq 0$, defined in [90], as the space of functions whose derivatives up to and including k are square integrable over Ω. That is,

$$H^k(\Omega) = \left\{u \in L^2(\Omega) : D^\alpha u \in L^2(\Omega) \text{ for } |\alpha| \leq k\right\}.$$

For non-integer $s \in \mathbb{R}$, let ξ be the Fourier-space variable in the Fourier transform of u. Then the Sobolev space for non-integer s is

$$H^s(\Omega) = \left\{u \in L^2(\Omega) : \left(1 + |\xi|^2\right)^{\frac{s}{2}} \hat{u} \in L^2(\Omega)\right\},$$

where $\hat{u}$ is the Fourier transform of u.

By assuming direct current and no internal sources of current, the electrostatic potential $u(x) \in H^1(\Omega)$ is governed by

$$L_\sigma u := \nabla \cdot (\sigma \nabla u) = \frac{\partial}{\partial x^i}\sigma^{ij}\frac{\partial u}{\partial x^j} = 0 \quad \text{in } \Omega. \tag{2.1}$$

The space to which the potential belongs is $H^1(\Omega)$ since this implies u and its first derivatives are finitely square integrable, which is physically equivalent to having finite total power. The voltage on the boundary is given by the following Dirichlet boundary condition:

$$u|_{\partial\Omega} = \phi, \quad \phi \in H^{1/2}(\partial\Omega), \tag{2.2}$$

and the boundary current is given by the following Neumann boundary condition:

$$\sigma \frac{\partial u}{\partial \nu}\bigg|_{\partial\Omega} = \sigma \nabla u \cdot \nu|_{\partial\Omega} = j, \quad j \in H^{-1/2}(\partial\Omega), \tag{2.3}$$

where ν is the outer normal to Ω. In order to ensure conservation of charge the following condition must also be satisfied on the boundary:

$$\int_{\partial\Omega} j = 0. \tag{2.4}$$

Solving (2.1) with either of the boundary conditions (2.2) or (2.3) for a given conductivity is the *forward problem*, which we will look at in more detail in section 2.2.

We define the Dirichlet-to-Neumann (D-N) map

$$\Lambda_\sigma : H^{1/2}(\partial\Omega) \to H^{-1/2}(\partial\Omega), \tag{2.5}$$

as the mapping between the boundary voltages and currents, i.e.

$$\Lambda_\sigma \phi = \sigma \left.\frac{\partial u}{\partial \nu}\right|_{\partial\Omega}. \tag{2.6}$$

The *inverse problem* is to find σ given Λ_σ. There are two key concepts we must consider when solving this problem: Does the boundary data uniquely determine the conductivity? And if so, does the solution depend continuously on the data? In section 2.1.3, we will review the main results that have worked towards answering these two questions.

2.1.2 A differential geometry viewpoint

Before reviewing these results, we now pose the inverse conductivity problem in an alternative setting. We refer the reader to section 2.3 for definitions of many of the concepts used in the following section. The following formulation was first considered by Lee and Uhlmann in 1989 [71].

They illustrate how the electrostatic problem of determining the anisotropic conductivity from its D-N map is related to that of determining a Riemannian metric from its D-N map for harmonic functions. They investigate the following geometric problem. Let M be the interior of a smooth n-manifold $\overline{M}$ with boundary and g a smooth Riemannian metric on $\overline{M}$ such that u is harmonic and has Dirichlet boundary data ϕ, that is,

$$\Delta_g u = 0 \quad \text{in } M, \quad u|_{\partial M} = \phi, \tag{2.7}$$

where Δ is the Laplace-Beltrami operator of g given by

$$\Delta_g u = \frac{1}{\sqrt{\det(g_{kl})}} \frac{\partial}{\partial x^i} \left(\sqrt{\det(g_{kl})} g^{ij} \frac{\partial u}{\partial x^j} \right). \tag{2.8}$$

The Neumann data is given by

$$\Lambda_g \phi = i^* \left(\star \mathrm{d}u \right), \tag{2.9}$$

where $\star$ is the Hodge star operator and i is the inclusion map, such that the pull-back of a form ω under i, denoted $i^*\omega$, restricts ω to the boundary ∂M. We give formal definitions of these concepts in section 2.3.

It is then observed that for $n \geq 3$ and $M = \Omega$, the electrostatic problem and the geometric problem are equivalent if

$$\sigma^{ij} = \sqrt{\det(g_{kl})}(g^{ij}), \tag{2.10}$$

where $(g^{ij}) = (g_{ij})^{-1}$ is the inverse metric tensor. We will revisit this problem in the next section when we review past work that has considered the problem of uniqueness of solution for the anisotropic inverse conductivity problem.

2.1.3 Uniqueness and stability of the inverse problem

We first review some results for the isotropic conductivity problem before going on to review the anisotropic problem.

Isotropic conductivity

For the isotropic problem, the conductivity tensor is of the form $\sigma^{ij} = \alpha \delta^{ij}$, for some scalar function α. Then we just denote the isotropic conductivity by the scalar function σ. The problem of uniqueness was first considered by Calderón [27]. He considered the quadratic form Q_σ associated with Λ_σ defined as

$$Q_\sigma(\phi) = \int_\Omega \sigma \left| \nabla u \right|^2 \, \mathrm{d}x, \tag{2.11}$$

where u solves (2.1) with Dirichlet condition (2.2). Q_σ represents the power necessary to maintain a boundary potential ϕ on $\partial\Omega$. By Green's first identity we have

$$Q_\sigma(\phi) = \int_{\partial\Omega} u \sigma \frac{\partial u}{\partial \nu} \, \mathrm{d}s = \langle \phi, \Lambda_\sigma \phi \rangle . \tag{2.12}$$

Therefore knowledge of Q_σ is equivalent to knowledge of Λ_σ. Calderón posed the questions: Is σ uniquely determined by Q_σ? If so, can we calculate σ from Q_σ? Let

$$\Phi : \sigma \mapsto Q_\sigma \tag{2.13}$$

be the mapping from conductivity to the quadratic form. Then Φ is bounded for real and positive $\sigma \in L^\infty(\Omega)$ and $u \in H^1(\Omega)$. By considering a linearisation of L_σ about $\sigma = 1$, it was shown that the map from conductivity σ to potential u has convergent Taylor series about $\sigma = 1$, hence Φ is analytic since a mapping from conductivity to potential is equivalent to the mapping Φ. It was also proved that the Fréchet derivative $\mathrm{d}\Phi$ is injective when σ is constant. By considering a linearisation of L_σ about $\sigma_0 = 1$, that is $\sigma = 1 + \delta$ for a small perturbation δ, it was also shown that the error in computing σ from knowledge of Q_σ is bounded by

$$\|\delta\|_{L^\infty} = \inf\left\{K \geq 0 : |\delta(x)| \leq K \text{ for almost all } x \in \Omega\right\}.$$

In 1984 Kohn and Vogelius [63] proved that Q_σ uniquely determines σ and all its normal derivatives at the boundary, provided σ is smooth near the boundary, i.e. $\sigma \in C^\infty(B)$ where B is some neighbourhood of a point $x_0 \in \partial\Omega$. As a direct consequence, if σ is real-analytic throughout Ω, we can use a Taylor series expansion about the point $x_0 \in \partial\Omega$ to find $\sigma(x)$ for any point $x \in \Omega$. The following year, Kohn and Vogelius [65] then proved uniqueness for the less restrictive case of piecewise real-analytic conductivity.

Since [65], uniqueness for the two-dimensional (2D) and three-dimensional (3D) cases have been treated separately. We review the 2D case first. Global uniqueness for a domain with smooth boundary ($\partial\Omega \in C^\infty$) and near constant conductivity $\sigma \in C^\infty(\Omega)$ was proved by Sylvester and Uhlmann [87]. In 1996, Nachman [81] proved global uniqueness for a domain with Lipschitz boundary and conductivity $\sigma \in W^{2,p}(\Omega)$, for $p > 1$. This requirement on σ states that the conductivity and all its first and second derivatives are in $L^p(\Omega)$. The following year, Brown and Uhlmann [25] relaxed the restriction on the conductivity for global uniqueness to $\sigma \in W^{1,p}(\Omega)$, $p > 2$. The global uniqueness problem for two dimensions was finally solved by Astala and Päivärinta in 2006 [12], where they proved that no regularity conditions need be assumed on the boundary of Ω and a conductivity $\sigma \in L^\infty(\Omega)$ is uniquely determined by Λ_σ.

In 1987, Sylvester and Uhlmann [88] proved global uniqueness for three-dimensional domains with smooth boundary and smooth conductivity $\sigma \in C^{\infty}(\Omega)$. Global uniqueness for the case of a known background conductivity and unknown discontinuity was proved by Isakov [58]. The restriction on σ from [88] was relaxed by Brown [24] to $\sigma \in C^{3/2+\epsilon}(\Omega)$ for $0 < \epsilon < 1/2$. Panchenko et al. [82], proved uniqueness for $\sigma \in W^{3/2,\infty}(\Omega)$. The least restrictive assumption on σ for which uniqueness has been proved for the three-dimensional case is by Haberman and Tataru [50], in which uniqueness is proved for bounded conductivities of the form $\sigma \in W^{1,\infty}(\Omega)$ or $\sigma \in C^{1}(\Omega)$. The general case of proving uniqueness for $\sigma \in L^{\infty}(\Omega)$ in 3D is still an open problem, however it is suggested by the authors of [50] that the assumption made in their paper, that is, the conductivity is Lipschitz, is an optimal assumption.

For the particular case of locating an inclusion in an otherwise homogeneous domain, it has been proved that only one boundary measurement is required ([60] and [10]).

The issue of stability has not been addressed as widely as uniqueness, although some stability measures have been proved. As well as giving an alternative proof of the main result in [63], Sylvester and Uhlmann [89] proved continuous dependence of the conductivity at the boundary on the D-N map for two and three dimensions. Alessandrini [6], proved that the inverse of the mapping $\sigma \mapsto \Lambda_{\sigma}$ has modulus of continuity of logarithmic type. For a detailed review of the issue of stability and problems yet to be solved the reader is referred to [8].

While these results help us to understand what assumptions we must make on the boundary and the conductivity, they are all proven in the limiting case of boundary data defined at infinitely many points. Realistically however, we can only take a finite number of measurements on a subset of the boundary. There are some results of uniqueness and stability for the case of partial boundary data in three dimensions. Bukhgeim and Uhlmann [93] proved uniqueness for $\sigma \in C^{2}(\Omega)$ given knowledge of the D-N map on a possibly large subset of the boundary. Kenig et al. [62] improved the result in [93] by proving the same result for knowledge of D-N on possibly smaller subsets of the boundary. Isakov [59] proved uniqueness for partial data on a half-space for the closely related Schrödinger equation. Heck and Wang [53] proved a log-log type stability estimate for the case of partial boundary data.

Anisotropic conductivity

Before giving an overview of the main results for this problem we refer the reader to [4] and [92] for excellent reviews of work in this area.

In [71] it is commented that Luc Tartar observed that if $\psi : \overline{\Omega} \to \overline{\Omega}$ is a smooth diffeomorphism such that $\psi|_{\partial\Omega} = I$, then the conductivities σ and

$$\tilde{\sigma} = \frac{(D\psi)\,\sigma\,(D\psi)^T}{\det(D\psi)} \circ \psi^{-1}, \tag{2.14}$$

have the same D-N maps, that is, $\Lambda_{\tilde{\sigma}} = \Lambda_\sigma$. In the equivalent geometric problem this means that for any diffeomorphism $\psi : \overline{M} \to \overline{M}$, such that $\psi(x) = x$, for $x \in \partial M$, $\Lambda_{\psi^* g} = \Lambda_g$, where $\psi^* g$ denotes the pullback of g under ψ.

Much of the work from this point forward has been towards proving that this observation is the only obstruction to uniqueness of solution.

For the two-dimensional case it was conjectured in [71] that if $\Lambda_{\tilde{g}} = \Lambda_g$, then there exists a diffeomorphism ψ as defined above such that $\psi^* \tilde{g}$ is a conformal multiple of g, that is, there exists some smooth function ϕ such that

$$\psi^* \tilde{g} = \phi g. \tag{2.15}$$

In two dimensions, uniqueness up to diffeomorphism has been proved in [13] for $\sigma \in L^\infty$ by reducing the anisotropic problem to an isotropic one using isothermal coordinates.

For the three-dimensional case, it has been shown in [68] that the D-N map uniquely defines the metric g up to diffeomorphism for compact, real-analytic manifolds with boundary in which the data is only measured on part of the boundary that is real-analytic. This result was extended in [67] to non-compact real-analytic manifolds with boundary.

Anisotropic conductivity partially known a priori

In 1983, Kohn and Vogelius [64] proved that in some neighbourhood of a point on the boundary, if all but one eigenvector and eigenvalue of the unknown conductivity are known in this neighbourhood, and if the eigenvector corresponding to the unknown eigenvalue is not tangent to the boundary at this point then the unknown eigenvalue and its derivatives up to the dimension n can be uniquely recovered at this point.

In 1990, Alessandrini [7] considered conductivities of the form $\sigma = A(a(x))$, where $A(t)$ is a given matrix-valued function and $a(x)$ is an unknown scalar function to be determined by the data. Uniqueness was proved for piecewise analytic perturbations of a. This result was extended in [9] to conductivities of the form $\sigma = A(x, a(x))$, where again $A(x, t)$ is a known matrix-valued function.

In 1997, Lionheart [73] considered the conformal inverse problem of recovering the scalar function α such that $\sigma = \alpha\sigma_0$, given that σ_0 is known a priori. By definition, two metrics g and g_0 are conformally related if $g = \lambda g_0$ for some function λ. It is observed in [73] that since conformally related metrics result in conductivities $\sigma(g) = \lambda^{n/2-1}\sigma(g_0)$, $\sigma(g) = \sigma(g_0)$ for $n = 2$, which suggests why this dimension is a special case. For $n \geq 3$, it is shown that the map from the space of scalar functions in the conformal inverse problem to corresponding D-N map is injective. It is also shown that if ψ is a diffeomorphism that fixes the boundary such that the pullback of g under ψ is conformally related to g then the scalar function α is uniquely determined. Uniqueness is also proved for piecewise analytic scalar functions α for the same problem.

Abascal et al. [1] presented numerical results that suggest all three eigenvalues of the unknown conductivity tensor are uniquely determined if all the eigenvectors are known a priori. These results were presented in the context of a piecewise linear approximation of the eigenvalues in the finite element method (see section 2.2 for more details on the finite element method).

2.2 Forward problem and finite element method

In the previous section we introduced the problem of determining the conductivity from known boundary measurements. Most methods used for solving the inverse problem in practice attempt to fit data simulated from some chosen conductivity to the real measured data and perturb the conductivity in the simulation until the data matches to a given tolerance [74]. Some direct methods, for example the factorisation method [26], do not rely on fitting the data but rely on the difference between the real data and data simulated from a homogeneous conductivity distribution in order to find inhomogeneities within the domain. In both cases we are required to simulate the boundary data for a given conductivity distribution. This is known as

the *forward problem* and involves solving the partial differential equation in (2.1) with either boundary condition (2.2) or (2.3). In some special cases, such as a homogeneous half-space domain, analytic solutions exist but in most practical problems the problem domain and possible conductivity distributions are not simple enough for an analytic solution to exist, so we are required to use numerical methods to simulate data.

A method that is well suited to complicated domains is the finite element method (FEM). There are many books on the method, for example [86], [44], [80], which discuss important issues such as convergence rates and error approximation. The book by Bossavit [21] introduces the role that algebraic topology plays in the context of FEM. This method involves discretising the domain into triangles in two dimensions or tetrahedra in three dimensions, which are called *elements*. Since we live in a three-dimensional world, we will focus on tetrahedral discretisation although much of the finite element theory presented here is general and applies to two or three dimensions. Once the domain is discretised we approximate the solution to the partial differential equation (PDE) by piecewise polynomial functions of chosen degree. The following subsections describe the process for reducing the PDE and boundary conditions to a linear system of equations in terms of specific values of the interpolation polynomial.

2.2.1 Weak formulation

In order to formulate the problem in terms of FEM we start with the continuous case, that is, with a function $u \in H^1(\Omega)$, where Ω is the problem domain. Let $v \in H_0^1(\Omega)$ be some trial function where $H_0^1(\Omega) = \{v \in H^1(\Omega) : \operatorname{supp} v = \overline{\Omega}\}$ is the space of square integrable functions with square integrable derivatives that are non-zero only in the interior of Ω. We shall keep things general by assuming that the conductivity is anisotropic so that σ is a tensor field. We refer the reader to section 2.3 for the formal definition of a symmetric tensor, but for the purposes of this analysis we can think of it as being an $n \times n$ symmetric matrix-valued function, where n is the problem dimension.

Multiplying (2.1) by v and integrating over Ω gives

$$\int_\Omega v \nabla \cdot (\sigma \nabla u) \, \mathrm{d}V = 0. \tag{2.16}$$

We have the following vector identity for the divergence of a scalar function ϕ and a

vector field F,

$$\nabla \cdot (\phi F) = \phi \nabla \cdot F + F \cdot \nabla \phi. \tag{2.17}$$

Let $\phi = v$ and $F = \sigma \nabla u$, then combining (2.17) and (2.16) gives

$$\int_\Omega \nabla \cdot (v\sigma \nabla u)\, \mathrm{d}V = \int_\Omega \nabla v \cdot (\sigma \nabla u)\, \mathrm{d}V. \tag{2.18}$$

Then applying the divergence theorem to the left hand side of (2.18) gives

$$\int_\Omega \nabla v \cdot (\sigma \nabla u)\, \mathrm{d}V = \int_{\partial\Omega} v\sigma \nabla u \cdot \nu\, \mathrm{d}S, \quad \forall v \in H_0^1(\Omega) \tag{2.19}$$

This is known as the *weak form* of the problem since (2.16) is clearly a weaker condition on u than (2.1). The right hand side of (2.19) is just the Neumann boundary condition in (2.3) multiplied by v and integrated over the boundary $\partial\Omega$. In many practical problems more realistic boundary conditions that model current applied on electrodes are required. For a description of how these models are incorporated into the weak form we refer the reader to [56].

2.2.2 Discretised Problem

In order to solve the weak form numerically we let $V^h \subset H^1(\Omega)$ and $V_0^h \subset H_0^1(\Omega)$ be finite dimensional subspaces of $H^1(\Omega)$ and $H_0^1(\Omega)$ respectively. Let $u_h \in V^h$ and $v_h \in V_0^h$, then (2.19) holds for u_h and v_h. As noted earlier we would like to approximate the potential u by piecewise polynomials of chosen degree. The simplest and most widely implemented approximations are piecewise linear functions, so we will assume that V^h is the space of piecewise linear functions.

In order to find a numerical solution we need to discretise the domain Ω so that we can approximate the potential at specific points and use the polynomials in V^h to interpolate the solution between these specified points. An approach that is very useful for approximating complex geometries is to partition the domain into tetrahedra resulting in a tetrahedral finite element mesh Ω_h, which is a geometric approximation to Ω. We label the vertices of the mesh x_i, for $i = 1, \ldots, n_V$ and the tetrahedra T_k, for $k = 1, \ldots, n_T$. A tetrahedral mesh is the geometric realisation of a simplicial complex which will be introduced in section 2.4, in which each tetrahedron is uniquely defined by its vertices and the intersection of two tetrahedra is either a single vertex, an edge defined by its endpoints, or a face.

We introduce the following nodal basis functions

$$\varphi_i(x_j) = \delta_{ij},$$

where δ_{ij} is the Kronecker delta and each φ_i is linear between vertices as we require for the piecewise linear approximation we specified above. Clearly, a function φ_i is non-zero only on tetrahedra for which x_i is a vertex. We can then represent the solution as a linear combination of the nodal basis functions,

$$u_h(x) = \sum_{i=1}^{n_V} u_i \varphi_i(x), \tag{2.20}$$

where u_i are nodal values which need to be found. Let the test function v_h be given by φ_i at each node i, then inserting this into (2.19) gives

$$\int_{\Omega_h} \nabla\varphi_i \cdot (\sigma \nabla u_h)\, \mathrm{d}V = \int_{\partial\Omega_h} \varphi_i \sigma \nabla u_h \cdot \nu \, \mathrm{d}S, \quad \text{for } i = 1, \ldots, n_V. \tag{2.21}$$

Inserting the expression for u_h from (2.20) into the above we have

$$\sum_{j=1}^{n_V} u_j \int_{\Omega_h} \nabla\varphi_i \cdot (\sigma \nabla \varphi_j)\, \mathrm{d}V = \sum_{j=1}^{n_V} u_j \int_{\partial\Omega_h} \varphi_i \sigma \nabla \varphi_j \cdot \nu \, \mathrm{d}S, \quad \text{for } i = 1, \ldots, n_V. \tag{2.22}$$

Depending on the model that is used to implement the applied current and measured voltage on the boundary, the right-hand side of (2.22) will take different forms. We refer the reader to [31] for a discussion and experimental validation of various boundary condition models. Since it is not the purpose of this work to solve to full finite element system we will just state that the finite element problem results in the problem of solving a linear system of equations of the form

$$Ku = f, \tag{2.23}$$

for the nodal potentials u and some vector f which is determined by the boundary condition model. We define the system matrix K such that its entries are given by

$$K_{ij} := \int_{\Omega_h} \nabla\varphi_i \cdot (\sigma \nabla \varphi_j)\, \mathrm{d}V, \tag{2.24}$$

for $i, j = 1, \ldots, n_V$. Then if we approximate the conductivity by a piecewise constant function such that

$$\sigma(x) = \sum_{k=1}^{n_T} \sigma_k \chi_k(x), \tag{2.25}$$

where χ_k is the characteristic function of tetrahedron T_k given by

$$\chi_k(x) = \begin{cases} 1 & \text{if } x \in T_k, \\ 0 & \text{if } x \notin T_k. \end{cases}$$

Then (2.24) reduces to

$$K_{ij} = \sum_{k=1}^{n_T} \int_{T_k} \nabla\varphi_i \cdot (\sigma_k \nabla\varphi_j)\, \mathrm{d}V. \tag{2.26}$$

Since the nodal basis function φ_i is non-zero only on tetrahedra for which x_i is a vertex and is linear on these tetrahedra, its gradient is constant on these tetrahedra and zero elsewhere. Moreover, as specified earlier, φ_i must increase linearly on each tetrahedron for which x_i is a vertex, from 0 at the other vertices, to 1 at x_i. Since the other vertices all share the face opposite x_i, the gradient φ_i has direction defined by the inward normal of the face opposite vertex x_i. To ensure that φ_i increases at the correct rate, $\nabla\varphi_i$ is scaled by the perpendicular distance of x_i to the face opposite x_i. Then the (i, j) entry of the system matrix is non-zero only when vertices x_i and x_j are on the same tetrahedron, that is, they share an edge. Then K_{ij} is given by

$$K_{ij} = \sum_{\{m:\{i,j\}\in T_m\}} |T_m| \frac{n_i^{(m)} \cdot (\sigma_m n_j^{(m)})}{h_i^{(m)} h_j^{(m)}}, \tag{2.27}$$

where $|T_m|$ is the volume of tetrahedron T_m, $n_i^{(m)}$ is the inward normal of the face in T_m opposite x_i and $h_i^{(m)}$ is the perpendicular distance of this face to vertex x_i and the sum is over tetrahedra that have $\{x_i, x_j\}$ as an edge. Let

$$c_i^{(m)} = \frac{n_i^{(m)}}{h_i^{(m)}},$$

and denote the vertices of the face opposite x_i in T_m by $\{x_j, x_k, x_l\}$. Then if we assume an orientation of the vertices in T_m consistent with figure 2.1 and denote by l_{ij} the length of the edge between x_i and x_j,

$$\begin{aligned} c_i^{(m)} &= \frac{(x_k - x_l) \times (x_j - x_l)}{\|(x_k - x_l) \times (x_j - x_l)\| h_i^{(m)}}, \\ &= \frac{(x_k - x_l) \times (x_j - x_l)}{l_{kl} l_{jl} \sin(\alpha_{l,jk}) l_{il} \sin(\alpha_{l,ij}) \sin(\theta_{jl})}. \end{aligned}$$

But $\sin\theta_{jl}$ can also be written in terms of cross-products as

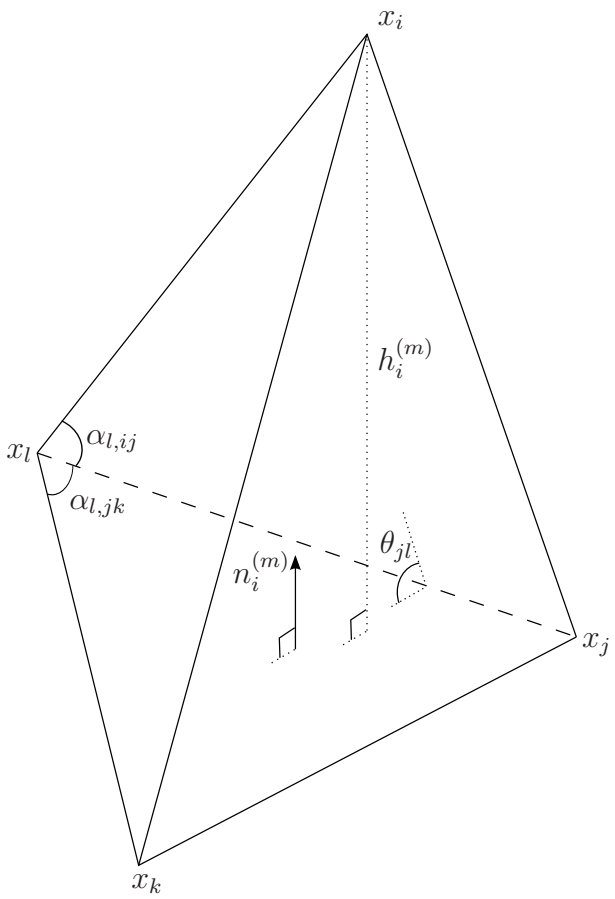


Figure 2.1: Labelling used for basis function gradient calculations.

$$\begin{aligned}\sin\theta_{jl} &= \frac{\|\left((x_j - x_l)\times(x_i - x_l)\right)\times\left((x_j - x_l)\times(x_k - x_l)\right)\|}{l_{jl}l_{il}\sin(\alpha_{l,ij})l_{jl}l_{kl}\sin(\alpha_{l,jk})},\\ &= \frac{\|\left((x_j - x_l)\cdot\left((x_i - x_l)\times(x_k - x_l)\right)\right)(x_j - x_l)\|}{l_{jl}l_{il}\sin(\alpha_{l,ij})l_{jl}l_{kl}\sin(\alpha_{l,jk})},\\ &= \frac{6|T_k|}{l_{jl}l_{il}l_{kl}\sin(\alpha_{l,ij})\sin(\alpha_{l,jk})}.\end{aligned}$$

Then

$$c_i^{(m)} = \frac{(x_k - x_l)\times(x_j - x_l)}{6|T_m|},$$

and similarly for $c_j^{(m)}$. Then for an anisotropic conductivity matrix defined on each tetrahedron, the system matrix is given by

$$K_{ij} = \sum_{\{m:\{i,j\}\in T_m=\{i,j,l,k\}\}} \frac{\left((x_k - x_l)\times(x_j - x_l)\right)\cdot\left(\sigma_m\left((x_i - x_l)\times(x_k - x_l)\right)\right)}{36|T_m|}. \quad (2.28)$$

In chapter 4 we will give simplifications of the expression in (2.28) for various forms of conductivity. For example, in the case of isotropic conductivity, this expression reduces to the well known cotangent formula [21].

2.3 Differential geometry

In this section we introduce some concepts from differential geometry that are necessary for understanding problems related to anisotropy in EIT. It also turns out that many concepts in applied mathematics can be described more generally and elegantly using the theory in this section than are usually done so in more elementary texts. Some of the concepts in this section can seem fairly abstract at first glance so we attempt to give physical or intuitive explanations after giving formal definitions. Most of the theory outlined in this chapter, along with many other related topics, can be found in the books [2] and [46].

2.3.1 Manifolds

In the following, let M be a topological space. We refer the reader to [22] and [2] for the general abstract definition of a topological space.

Definition 2.3.1. A *chart* is a pair (U, φ_U), where $U \subset M$ and φ_U is a bijection $\varphi_U : U \to W$, mapping U to some open set $W \subset \mathbb{R}^n$.

So a chart is a mapping that assigns coordinates to a subset U of M locally. For this reason a chart is sometimes called a *coordinate patch* and the function φ_U called a *coordinate map*. The above definition can be extended to a subset W of any Banach space but we restrict ourselves here to Euclidean space $\mathbb{R}^n$ so as not to overcomplicate things.

Definition 2.3.2. Let (U, φ_U) and (V, φ_V) be two charts on M such that $U \cap V \neq \emptyset$. The *overlap map* is the map

$$\varphi_{VU} = \varphi_V \circ \varphi_U^{-1}|_{\varphi_U(U \cap V)},$$

such that φ_{VU} is a diffeomorphism, that is, φ_{VU} and its inverse are differentiable.

The overlap map can be thought of as assigning coordinates to $U \cap V$ using φ_U then applying a change of coordinates to those associated with the chart (V, φ_V).

Definition 2.3.3. A C^k *atlas* on M is a family of charts $\mathcal{A} = \{(U_i, \varphi_i) : i \in I\}$ for some index set I, such that $M = \bigcup_{i \in I} U_i$ and the overlap map between any two members of $\mathcal{A}$ is a C^k diffeomorphism.

So an atlas is a collection of charts whose domains cover M and any overlap maps are differentiable. Then for any point that lies in the intersection of two chart domains, we can apply a differentiable change of coordinates to those assigned by one chart to those assigned by the other. This is an important fact since it allows us to apply the chain rule when we want to differentiate a function with respect to a different set of local coordinates.

Definition 2.3.4. We say that two atlases $\mathcal{A}_1$ and $\mathcal{A}_2$ are *equivalent* if $\mathcal{A}_1 \cup \mathcal{A}_2$ is also an atlas.

Clearly, the union of two atlases covers the underlying set M by the definition of an atlas. Therefore two atlases are equivalent if the overlap map for a chart from each atlas is a diffeomorphism, that is, there exists a differentiable change of coordinates between local coordinate systems defined in different atlases.

Definition 2.3.5. A C^k *differentiable structure* $\mathcal{D}$ on M is an equivalence class of C^k atlases on M, and the union of all atlases equivalent to $\mathcal{A}$ is the C^k differentiable structure *generated by* $\mathcal{A}$.

Definition 2.3.6. If we have a differentiable structure on M then M is a *differentiable manifold*. If every chart on M maps to values in an n-dimensional vector space then M is an *n-manifold*.

Since a differentiable structure on M is an equivalence class of atlases that cover M and defines a differentiable change of coordinates between the charts that form the atlases, it seems reasonable to define a single atlas on M to determine the manifold.

2.3.2 Tangent vectors and vector fields

In the following we define $p \in M$ as a point in the abstract set M. Only by applying a coordinate map of a chart (U, ϕ) such that $p \in U$, do we assign coordinates to the point p. We follow the approach of [2] and [46] of defining vectors as the velocity vector of a curve in the manifold M. Let $a, b \in \mathbb{R}$ so that $[a, b]$ is an interval on the real line, then let $c : [a, b] \to M$ be a curve such that c is differentiable, by which we mean the composition $\varphi \circ c$ is differentiable in $\mathbb{R}^n$. Then a point $p \in M$ is the realisation of c for a particular value on the interval $[a, b]$, that is, $p = c(t)$ for $t \in [a, b]$. The velocity

vector at a point $p_0 = c(0) \in U$ with local coordinates given by $x_U = (x_U^1, \ldots, x_U^n)$ is the vector

$$\dot{c}(0) = \left(\left.\frac{\mathrm{d}x^1}{\mathrm{d}t}\right|_{t=0}, \ldots, \left.\frac{\mathrm{d}x^n}{\mathrm{d}t}\right|_{t=0} \right).$$

Then if p also lies in the patch V, we can use the chain rule, since the overlap map of U and V is a diffeomorphism, to change coordinates so that

$$\left.\frac{\mathrm{d}x_V^i}{\mathrm{d}t}\right|_{t=0} = \frac{\partial x_V^i}{\partial x_U^j}(p_0) \left.\frac{\mathrm{d}x_U^j}{\mathrm{d}t}\right|_{t=0}.$$

We now define the notion of a vector on an n-manifold.

Definition 2.3.7. We define a *tangent vector* or *contravariant vector* or simply *vector* X at $p \in U \subset M$, as the n-tuple

$$X_U = \left(\frac{\mathrm{d}x_U^1}{\mathrm{d}t}, \ldots, \frac{\mathrm{d}x_U^n}{\mathrm{d}t} \right),$$

such that if $p \in U \cap V$,

$$X_V^i = \frac{\partial x_V^i}{\partial x_U^j}(p) X_U^j.$$

Again, we have used the fact that the overlap map between two charts U and V is a diffeomorphism which allows us to use the chain rule to change coordinates. We see then that the numbers X^i can be thought of as derivatives of the corresponding local coordinates with respect to the artificial time we introduced to parameterise the curve. We now define the vector space to which these contravariant vectors belong.

Definition 2.3.8. Let $p \in M$, then the *tangent space* of M at p, denoted by M_p is the vector space consisting of all vectors tangent to M at p. If $x = (x^1, \ldots, x^n)$ is a coordinate system for p then the set of directional vectors

$$\left.\frac{\partial}{\partial x^1}\right|_p, \ldots, \left.\frac{\partial}{\partial x^n}\right|_p$$

forms a basis for this vector space. We call this basis a *coordinate frame*.

Definition 2.3.9. We call the disjoint union of the tangent spaces of M the *tangent bundle*, denoted TM. That is,

$$TM = \bigsqcup_{p \in M} M_p.$$

In order to define vectors as differentiable operators and use the above basis notation we use the following definition.

Definition 2.3.10. A *vector field* on U is the assignment of a vector X to each point in U such that for $p \in M$, the assignment $X(p)$ is in the tangent space M_p.

Since we have a basis for the tangent space given in definition 2.3.8, we can write a vector field X in terms of local coordinates $x = (x^1, \ldots, x^n)$ as a linear combination of functions $X^i(x)$ and the basis functions $\partial/\partial x^i$ as follows

$$X = X^i(x)\frac{\partial}{\partial x^i}. \tag{2.29}$$

We now introduce the notion of the differential of a mapping between two manifolds since this is used in the anisotropic non-uniqueness result of Tartar (see section 2.1.3 for this result). Let M be an m-manifold and N an n-manifold and suppose $F : M \to N$ is a mapping between the manifolds such that for a point $p \in M$, local coordinates are denoted by $x = (x^1, \ldots, x^m)$ and for $F(p) \in N$ we have local coordinates $y = (y^1, \ldots, y^n)$, such that $y^i = F^i(x^1, \ldots, x^m)$ for $i = 1, \ldots, n$.

Definition 2.3.11. The *differential* of $F : M \to N$ at a point $p \in M$ is the map $F_* : M_p \to N_{F(p)}$, such that for the vector $X \in M_p$ that is a velocity vector $\dot{p} = \dot{c}(0)$, F_*X is the velocity vector

$$\left.\frac{\mathrm{d}F(p)}{\mathrm{d}t}\right|_{t=0} = Y^i(y)\frac{\partial}{\partial y^i},$$

at the point $F(p)$ on N. The *Jacobian* of the mapping is the matrix

$$(F_*)^i{}_j = \frac{\partial F^i}{\partial x^j}(p) = \frac{\partial y^i}{\partial x^j}(p).$$

We next introduce some concepts that rely on the differential of a mapping between two manifolds. We will use the following concept later in this chapter when mapping from discrete manifolds to Euclidean space and again in chapters 3 and 4.

Definition 2.3.12. A map $\varphi : S \to T$ between two topological spaces is a *homeomorphism* if it is a bijection and both φ and φ^{-1} are continuous. If such a map exists between S and T then S and T are said to be *homeomorphic*.

The classic example of two spaces being homeomorphic is that of a teacup-shaped object and a torus, since there exists a continuous deformation of the teacup into a torus and back again. That is, the map defining the deformation is continuous, as is the inverse.

Definition 2.3.13. A mapping $F : M \to N$ is an *immersion* if $F_* : M_p \to N_{F(p)}$ is injective for each point $p \in M$.

Definition 2.3.14. An immersion $F : M \to N$ is an *embedding* if F is a homeomorphism onto $F(M)$ with the relative topology induced from N.

We will give examples of immersions and embeddings in section 2.4 in terms of discrete manifolds since we will use them in this context in our work.

2.3.3 Tensors and differential forms

We now extend the notion of vectors introduced in the previous subsection by introducing tensors, before going on to describe the important concepts of exterior and differential forms.

Let $E_1, \ldots, E_k$ be Banach spaces, and let $L^k(E_1, \ldots, E_k; \mathbb{R})$ denote the vector space of continuous k-multilinear maps from the product space $E_1 \times \ldots \times E_k$ to $\mathbb{R}$. We use the above notation in the following definitions and begin with general definitions of tensors before introducing them in the context of differentiable manifolds.

Definition 2.3.15. Let E be a vector space, then a *tensor* is an element of the space $T^r{}_s(E) = L^{r+s}(E^*, \ldots, E^*, E, \ldots, E; \mathbb{R})$, with r copies of E^* and s copies of E, *contravariant* of order r and *covariant* of order s.

The above definition matches the indexing notation we used in the previous subsection in that we used upper indices to define elements of contravariant vectors. By E^* we denote the dual space of E, which is the vector space of linear functionals on E. So an element $f \in E^*$ is a mapping $f : E \to \mathbb{R}$. Let E be an n-dimensional vector space, for example $\mathbb{R}^n$. Let $\{e_1, \ldots, e_n\}$ be a basis for E with corresponding dual basis $\{e^1, \ldots, e^n\}$. As an example of the above definition let $t \in T^0{}_2(E)$, then we can think of the components of t as an $n \times n$ matrix $t_{ij} = t(e_i, e_j)$, which is a bilinear map acting on e_i, e_j.

Definition 2.3.16. Let $t_1 \in T^{r_1}_{s_1}(E)$ and $t_2 \in T^{r_2}_{s_2}(E)$ be two tensors, then their *tensor product* $t_1 \otimes t_2 \in T^{r_1+r_2}_{s_1+s_2}(E)$, is given by

$$(t_1 \otimes t_2)(\alpha^1, \ldots, \alpha^{r_1}, \beta^1, \ldots, \beta^{r_2}, a_1, \ldots, a_{s_1}, b_1, \ldots, b_{s_2}) = \\ t_1(\alpha^1, \ldots, \alpha^{r_1}, a_1, \ldots, a_{s_1}) t_2(\beta^1, \ldots, \beta^{r_2}, b_1, \ldots, b_{s_2}),$$

for $\alpha^i, \beta^i \in E^*$ and $a_i, b_i \in E$.

We now define tensors in the context of differentiable manifolds. Recall the tangent bundle TM is the vector space made up of the disjoint union of the tangent spaces of the manifold M.

Definition 2.3.17. A *tensor* at a point $p \in M$ is an element of $T^r{}_s(M_p)$. A *tensor field* is an element of $T^r{}_s(M) := T^r{}_s(TM)$.

We can now define covariant vectors in terms of tensors.

Definition 2.3.18. A *covariant vector*, *covector*, or *1-form* at $p \in M$ is a tensor of the form $T^0_1(M_p) = M_p^*$. The dual vector space M_p^* is called the *cotangent space*. Similarly, a *covector* is an element of $T^0_1(TM) = T^*M$. The space T^*M is called the *cotangent bundle*.

We now work towards defining a basis for the dual space M_p^* in terms of derivatives as we did earlier for M_p.

Definition 2.3.19. Let $f : M \to \mathbb{R}$ be some function. The *differential* of f at a point $p \in M$ is the linear functional $\mathrm{d}f_p : M_p \to \mathbb{R}$.

So the differential $\mathrm{d}f_p$ is a 1-form at $p \in M$ which maps vectors in the tangent space of a point in M to a real number. Hence $\mathrm{d}f_p \in M_p^*$. Writing in terms of local coordinates and the basis of the tangent space M_p defined in definition 2.3.8, we have

$$\mathrm{d}f_p(X) = \mathrm{d}f_p\left(X^j \frac{\partial}{\partial x^j}\right) = X^j \mathrm{d}f\left(\frac{\partial}{\partial x^j}\right) = X^j \frac{\partial f}{\partial x^j}(p) = X_p(f), \tag{2.30}$$

for $X \in M_p$. If we apply the differential of a coordinate map x^i to a vector $X \in M_p$, we have

$$\mathrm{d}x^i\left(X^j \frac{\partial}{\partial x^j}\right) = X^j \mathrm{d}x^i\left(\frac{\partial}{\partial x^j}\right) = X^j \frac{\partial x^i}{\partial x^j} = X^i.$$

Then $\mathrm{d}x^1, \ldots, \mathrm{d}x^n$ is a basis for the cotangent space M_p^*. This means we can write any 1-form α as a linear combination of functionals $\alpha_j(x)$ and the basis functionals $\mathrm{d}x^i$. That is,

$$\alpha = \alpha_j(x)\mathrm{d}x^j. \tag{2.31}$$

Definition 2.3.20. A *differential 1-form* is a covector of the form expressed in (2.31).

We now define the inner product on the tangent space.

Definition 2.3.21. An *inner product* $\langle \cdot, \cdot \rangle$ is a bilinear form that is symmetric and nondegenerate. That is, for vectors $v, w \in M_p$, $\langle v, w \rangle = \langle w, v \rangle$, and if $\langle v, w \rangle = 0$, then either $v = 0$ or $w = 0$.

Suppose we have two vectors $v, w \in M_p$, then since the vectors $\partial/\partial x^i$ form a basis for the tangent space we can write their inner product as

$$\langle v, w \rangle = \left\langle v^i \frac{\partial}{\partial x^i}, w^j \frac{\partial}{\partial x^j} \right\rangle = v^i \left\langle \frac{\partial}{\partial x^i}, w^j \frac{\partial}{\partial x^j} \right\rangle = v^i \left\langle \frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j} \right\rangle w^j. \tag{2.32}$$

Then the values $\left\langle \frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j} \right\rangle$ form a matrix, which we define in a moment.

Definition 2.3.22. The inner product $\langle \cdot, \cdot \rangle$ is *positive definite* if for any $v \neq 0$, $\langle v, v \rangle > 0$.

Definition 2.3.23. The assignment of a positive definite inner product $\langle \cdot, \cdot \rangle$ on the tangent bundle TM is called a *Riemannian metric* on M. A manifold with a Riemannian metric is a *Riemannian manifold*. We define the *metric tensor* for a tangent space with local coordinates x as the matrix (g_{ij}) with entries

$$g_{ij}(x) = \left\langle \frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j} \right\rangle.$$

From the above definition we write the expression (2.32) in terms of the metric tensor as

$$\langle v, w \rangle = v^i g_{ij} w^j.$$

We can also use the metric tensor to associate the 1-form ω with a vector w by "lowering" the index,

$$\omega = \omega_i \mathrm{d}x^i = g_{ij} w^j \mathrm{d}x^i.$$

If we define the inverse metric as the matrix $(g^{ij}) = (g_{ij})^{-1}$, then we can associate a vector with a given 1-form as follows

$$w = w^i \frac{\partial}{\partial x^i} = g^{ij} \omega_j \frac{\partial}{\partial x^i}.$$

We see from this that $g_{ij} \in T^0_2(M)$ and $g^{ij} \in T^2_0(M)$ are tensor fields that vary between points on the manifold depending on the local coordinates assigned at each point.

Definition 2.3.24. The *index lowering operator* $^\flat : TM \to T^*M$ associated with the matrix (g_{ij}) maps vectors to 1-forms,

$$w^\flat = g_{ij} w^j \mathrm{d}x^i.$$

The *index raising operator* $^\sharp : T^*M \to TM$ associated with the matrix (g^{ij}) maps 1-forms to vectors,

$$\omega^\sharp = g^{ij} \omega_j \frac{\partial}{\partial x^i}.$$

We now have a way to associate a 1-form with a vector based on the Riemannian metric.

We now extend the concept of a 1-form to a k-form.

Definition 2.3.25. A *(differential) k-form* is an antisymmetric covariant tensor field $\omega \in T^0_k(M)$, that is

$$\omega(\ldots, w_i, \ldots, w_j, \ldots) = -\omega(\ldots, w_j, \ldots, w_i, \ldots).$$

We denote the space of k-forms on a manifold M by $\bigwedge^k M$.

We note here that a 0-form is simply a function that acts on the underlying coordinates. From the above definition if a k-form ω has a repeated vector in its list of arguments then $\omega = 0$. We will introduce a way of forming the product of two k-forms in a moment but first we define a useful notational device.

Definition 2.3.26. The symbol δ^I_J is the *generalised Kronecker delta*, which for two index sets $I = (i_1, \ldots, i_k)$ and $J = (j_1, \ldots, j_k)$, is defined as

$$\delta^I_J = \begin{cases} 1 & \text{if } J \text{ is an even permutation of } I, \\ -1 & \text{if } J \text{ is an odd permutation of } I, \\ 0 & \text{if } J \text{ is not a permutation of } I. \end{cases}$$

Then the usual permutation symbol, the *Levi-Civita symbol*, which tests if a set of indices is an odd or even permutation of $\{1, \ldots, k\}$, can be written in terms of δ^I_J as

$$\epsilon_{i_1 i_2 \cdots i_k} = \delta^{i_1 i_2 \cdots i_k}_{1,2,\ldots,k}.$$

Definition 2.3.27. The *wedge product*, also called the *exterior product*, is the map

$$\wedge : \bigwedge^{p} M \times \bigwedge^{q} M \to \bigwedge^{p+q} M.$$

For $\alpha \in \bigwedge^p M$, $\beta \in \bigwedge^q M$,

$$(\alpha \wedge \beta)(v_I) = \delta_I^{JK} \alpha(v_J)\beta(v_K),$$

where $I = (i_1, \ldots, i_{(p+q)})$ is a set of indices, $J = (j_1 < \ldots < j_p)$, $K = (k_1 < \ldots < k_q)$ are ordered index sets such that $J, K \subset I$.

The above definition gives us a way of calculating the product of two forms. Clearly, $\alpha \wedge \beta$ is antisymmetric due to the appearance of the generalised Kronecker delta. As an example of the wedge product, consider $\alpha, \beta \in \bigwedge^1 M$, for M a 2-manifold. Let

$$\alpha = \alpha_1 \mathrm{d}x^1 + \alpha_2 \mathrm{d}x^2,$$
$$\beta = \beta_1 \mathrm{d}x^1 + \beta_2 \mathrm{d}x^2.$$

Then

$$\begin{aligned} \alpha \wedge \beta &= (\alpha_1 \mathrm{d}x^1 + \alpha_2 \mathrm{d}x^2) \wedge (\beta_1 \mathrm{d}x^1 + \beta_2 \mathrm{d}x^2), \\ &= (\alpha_1\beta_2 - \alpha_2\beta_1)\mathrm{d}x^1 \wedge \mathrm{d}x^2. \end{aligned}$$

Another important concept relating forms and vectors, and one which appears in the geometric formulation of the EIT problem (see section 2.1.2), is the interior product whose definition we now state along with an illustrative example.

Definition 2.3.28. For $v, v_2, \ldots, v_k \in M_p$ and $\omega \in \bigwedge^k M$, their *interior product* is given by

$$\iota_v : \bigwedge^{k} M \to \bigwedge^{k-1} M,$$
$$\omega(v_2, \ldots, v_k) \mapsto \omega(v, v_2, \ldots, v_k),$$

such that $\iota_v \omega = 0$, if ω is a 0-form (function).

To illustrate the above definition we give the following example.

Example 2.3.1. Let M be a differentiable 3-manifold with a point $p \in M$, and let $v \in M_p$ be a vector on M. Suppose $\partial_i = \partial/\partial x^i$, for $i = 1, 2, 3$ is a basis for M_p and

$\mathrm{d}x^i$, for $i = 1, 2, 3$, is the corresponding basis for M_p^*. Then taking the interior product of v and the 3-form $\omega = \mathrm{d}x^1 \wedge \mathrm{d}x^2 \wedge \mathrm{d}x^3$ gives

$$\begin{aligned}
\iota_v \, \mathrm{d}x^1 \wedge \mathrm{d}x^2 \wedge \mathrm{d}x^3 &= \iota_{v^j \partial_j} \, \mathrm{d}x^1 \wedge \mathrm{d}x^2 \wedge \mathrm{d}x^3, \\
&= v^j \iota_{\partial_j} \, \mathrm{d}x^1 \wedge \mathrm{d}x^2 \wedge \mathrm{d}x^3, \\
&= v^j \left(\mathrm{d}x^1(\partial_j) \mathrm{d}x^2 \wedge \mathrm{d}x^3 - \mathrm{d}x^2(\partial_j) \mathrm{d}x^1 \wedge \mathrm{d}x^3 + \mathrm{d}x^3(\partial_j) \mathrm{d}x^1 \wedge \mathrm{d}x^2 \right), \\
&= v^j \delta^1{}_j \mathrm{d}x^2 \wedge \mathrm{d}x^3 - v^j \delta^2{}_j \mathrm{d}x^1 \wedge \mathrm{d}x^3 + v^j \delta^3{}_j \mathrm{d}x^1 \wedge \mathrm{d}x^2, \\
&= v^1 \mathrm{d}x^2 \wedge \mathrm{d}x^3 - v^2 \mathrm{d}x^1 \wedge \mathrm{d}x^3 + v^3 \mathrm{d}x^1 \wedge \mathrm{d}x^2,
\end{aligned}$$

which is clearly a 2-form.

We now introduce a way of differentiating forms.

Definition 2.3.29. The *exterior derivative* is the map

$$\mathrm{d} : \bigwedge^k M \to \bigwedge^{k+1} M,$$

such that the following conditions hold:

1. $\mathrm{d}(\alpha + \beta) = \mathrm{d}\alpha + \mathrm{d}\beta$;
2. for $f \in \bigwedge^0 M$, $\mathrm{d}f$ is the usual differential defined in definition 2.3.19;
3. for $\alpha \in \bigwedge^p$, $\beta \in \bigwedge^q$,

$$\mathrm{d}(\alpha \wedge \beta) = \mathrm{d}\alpha \wedge \beta + (-1)^p \alpha \wedge \mathrm{d}\beta;$$

4. $\mathrm{d}^2\alpha = \mathrm{d}(\mathrm{d}\alpha) = 0$, for all forms.

2.3.4 Orientation and Volume

We now introduce the idea of orientation and volume for a Riemannian manifold. An analogy in $\mathbb{R}^3$ is that we think of orientation as "handedness", that is, given the basis vectors of the plane e_x and e_y, if we have a right-handed coordinate system e_x, e_y, e_z, the third basis vector is given by $e_z = e_x \times e_y$, and the oriented volume of a parallelepiped P defined by vectors x, y and z is $V(P) = \det(x, y, z)$.

Before we define orientation,we introduce a few related concepts.

Definition 2.3.30. Let $\varphi : M \to M$ be a differentiable map between from an n-manifold to itself. Let $\omega \in \bigwedge^k N$ be a k-form on M. Then we define the *pull-back* φ^* of ω by φ as the map

$$\varphi^* : \bigwedge^k M \to \bigwedge^k M,$$
$$\omega(v_1, \ldots, v_k) \mapsto \omega(\varphi_* v_1, \ldots, \varphi_* v_k),$$

for $v_1, \ldots, v_k \in M_p$, where φ_* is the differential as defined in 2.3.11.

So the pull-back φ^* maps k-forms on M back to k-forms on M with the aid of the Jacobian matrix φ_*.

Definition 2.3.31. Let $\varphi : M \to M$ be a mapping between two n-manifolds. The *determinant* is the unique function $\det \varphi$ such that

$$\varphi^* \omega = (\det \varphi)\, \omega,$$

for $\omega \in \bigwedge^n M$.

In this sense, we can think of $\det \varphi$ as being the usual determinant of the Jacobian matrix φ_*. In order to introduce the notion of orientation we begin by defining the orientation of a tangent space and then a manifold.

Definition 2.3.32. For the tangent space M_p at $p \in M$, we call n-forms that are nowhere zero, that is elements of $\bigwedge^n M_p$, *volume forms*. We define an equivalence class $[\omega]$ on volume forms by saying ω_1 and ω_2 are equivalent if there exists some positive function c such that $\omega_1 = c\omega_2$. An equivalence class $[\omega]$ is called an *orientation* on M_p. The equivalence class $[-\omega]$ is a *reverse orientation*. A basis $e_1, \ldots, e_n$ of M_p is *positively (negatively) oriented* if $\omega(e_1, \ldots, e_n) > 0$ (< 0).

Suppose we have the volume form $\omega = \mathrm{d}y^1 \wedge \ldots \wedge \mathrm{d}y^n$ defined at point $p \in M$ with positively oriented orthonormal coordinate system $y^1, \ldots, y^n$, that is, $\langle \partial/\partial y^i, \partial/\partial y^j \rangle = \delta_{ij}$. Suppose $x_1, \ldots, x_n$ is any coordinate system at p. Then since the coordinate system given by y is orthonormal, we can write the metric tensor $g(x)$ as

$$g_{ij} = \frac{\partial y^k}{\partial x^i} \delta_{kl} \frac{\partial y^l}{\partial x^j}.$$

Then

$$\begin{aligned}\det(g_{ij}) &= \det\left(\frac{\partial y^k}{\partial x^i}\delta_{kl}\frac{\partial y^l}{\partial x^j}\right),\\ &= \det\left((\varphi_*)^T(\varphi_*)\right),\\ &= (\det\varphi)^2.\end{aligned}$$

Suppose we have the orientation preserving change of coordinates $\varphi : M \to M$, then the pull-back of ω under φ is

$$\begin{aligned}\varphi^*\omega &= \varphi^*(\mathrm{d}y^1 \wedge \ldots \wedge \mathrm{d}y^n),\\ &= (\det\varphi)(\mathrm{d}x^1 \wedge \ldots \wedge \mathrm{d}x^n),\\ &= \sqrt{\det(g_{ij})}\,\mathrm{d}x^1 \wedge \ldots \wedge \mathrm{d}x^n.\end{aligned}$$

So this gives a volume form that can be defined for any coordinate system.

Definition 2.3.33. Let M be a Riemannian n-manifold with metric g_{ij}, then the *Riemannian volume form* μ_g is given by

$$\mu_g = \sqrt{\det(g_{ij})}\,\mathrm{d}x^1 \wedge \ldots \wedge \mathrm{d}x^n. \tag{2.33}$$

Definition 2.3.34. A manifold M is *orientable* if there exists a volume form $\mu_g(p) \neq 0$, for all $p \in M$.

We now have a way of orienting an orientable manifold, which will be important later when we come to integration of forms.

2.3.5 Vector calculus on manifolds

In this section we introduce some concepts that will allow us to write the well-known vector calculus operators in terms of differential forms. We begin by defining an operator that maps k-forms to $(n-k)$-forms on n-manifolds.

Definition 2.3.35. For $\alpha, \beta \in \bigwedge^k M$, the *Hodge star operator* is the isomorphism $\star : \bigwedge^k M \to \bigwedge^{n-k} M$ satisfying

$$\alpha \wedge \star\beta = \langle \alpha, \beta \rangle\, \mu_g. \tag{2.34}$$

Applied to a single k-form ω we have

$$\star\omega = \sqrt{\det(g_{ij})}\, g^{i_1 j_1} \ldots g^{i_k j_k} \omega_{j_1 \ldots j_k} \epsilon_{i_1 \ldots i_k l_1 \ldots l_{n-k}} \mathrm{d}x^{l_1} \wedge \ldots \wedge \mathrm{d}x^{l_{n-k}}.$$

So the Hodge star maps k-forms to $(n-k)$-forms using a transformation that depends on the Riemannian metric tensor. We now use the Hodge star in the following definition of the adjoint operator of exterior differentiation.

Definition 2.3.36. The *codifferential* operator δ is the mapping

$$\begin{aligned}\delta : \bigwedge^{k} M &\to \bigwedge^{k-1} M, \\ \omega &\mapsto (-1)^{n(k+1)+1} \star \mathrm{d} \star \omega,\end{aligned}$$

with $\delta\omega = 0$ if $\omega \in \bigwedge^0 M$.

We now have all the tools required to do some vector calculus related to the inverse conductivity problem on manifolds. In the following two definitions let M be a Riemannnian manifold with metric tensor g.

Definition 2.3.37. Let $f \in \bigwedge^0 M$ be a 0-form (a function), then the *gradient* of f, denoted $\mathrm{grad}_g f$ or $\nabla_g f$ is the vector associated with the 1-form $\mathrm{d}f$, that is,

$$\mathrm{grad}_g f = (\mathrm{d}f)^\sharp .$$

We note that in some cases we will drop the subscript g in the above definition. In terms of basis vectors and forms we can write the above definition as

$$\mathrm{grad} f = (\mathrm{d}f)^\sharp = \left(\frac{\partial f}{\partial x^j}\mathrm{d}x^j\right)^\sharp = g^{ij}\frac{\partial f}{\partial x^j}\frac{\partial}{\partial x^i},$$

so the components of the vector $\mathrm{grad} f$ are $g^{ij}\partial f/\partial x^j$. Clearly if g is the identity matrix as in $\mathbb{R}^3$ this reduces to the usual gradient vector of a function f.

Definition 2.3.38. Let F be a vector field on M. The *divergence* of F, denoted $\mathrm{div} F$ or $\nabla \cdot F$ is given by the scalar

$$\mathrm{div} F = \star \mathrm{d} \left(\star F^\flat\right).$$

To show this in terms of components and basis vectors we will apply each operator in turn. We restrict ourselves to a 3-manifold to simplify the notation. Let $|g|$ denote the determinant $\det(g_{ij})$, then

$$\star F^\flat = \star(g_{ij}F^j \mathrm{d}x^i) = \sqrt{|g|}F^i \epsilon_{ijk} \mathrm{d}x^j \wedge \mathrm{d}x^k.$$

Then applying the exterior derivative, we have

$$\mathrm{d}\left(\star F^{\flat}\right) = \frac{\partial}{\partial x^l}\left(\sqrt{|g|}F^i\epsilon_{ijk}\mathrm{d}x^j \wedge \mathrm{d}x^k\right)\mathrm{d}x^l = \frac{1}{\sqrt{|g|}}\frac{\partial}{\partial x^i}\left(\sqrt{|g|}F^i\right)\mu_g.$$

Then applying the Hodge star operator again gives

$$\mathrm{div}F = \star\mathrm{d}\left(\star F^{\flat}\right) = \frac{1}{\sqrt{|g|}}\frac{\partial}{\partial x^i}\left(\sqrt{|g|}F^i\right).$$

Applying div to grad$f$ results in the following operator.

Definition 2.3.39. The *Laplace-Beltrami operator* on functions on a Riemannian manifold is given by div grad $= \nabla^2$.

So for a function f we have the following expression

$$\mathrm{div}\,\mathrm{grad}f = \nabla^2 f = \frac{1}{\sqrt{|g|}}\frac{\partial}{\partial x^i}\left(\sqrt{|g|}g^{ij}\frac{\partial f}{\partial x^j}\right).$$

This is exactly equation (2.8) from the geometric formulation of the EIT problem. A related operator is defined in the following.

Definition 2.3.40. The *Laplace-deRham* operator $\Delta : \bigwedge^k M \to \bigwedge^k M$ is defined by $\Delta = \mathrm{d}\delta + \delta\mathrm{d}$. A k-form ω for which $\Delta\omega = 0$ is called harmonic and we denote the space of harmonic k-forms by $\mathcal{H}^k(M) = \{\omega \in \bigwedge^k M : \Delta\omega = 0\}$.

For a 0-form ϕ, we have

$$\Delta\phi = \mathrm{d}\delta\phi + \delta\mathrm{d}\phi = \delta\mathrm{d}\phi,$$

since $\delta\phi = 0$ by definition 2.3.36. Then

$$\Delta\phi = (-1)^{2n+1} \star \mathrm{d} \star \mathrm{d}\phi = -\mathrm{div}\,\mathrm{grad}\,\phi = -\nabla^2\phi.$$

2.3.6 Integration on manifolds

In this section we introduce the concept of integration of differential forms on manifolds. We then use this theory to give the weak form of the EIT problem in terms of differential forms. We first introduce a concept that will be necessary in giving a formal definition of an integral on a manifold.

Definition 2.3.41. A *partition of unity* on a manifold M is a locally finite open covering $\{U_i\}$ of M along with continuously differentiable non-negative functions $f_i : M \to \mathbb{R}$, such that the support of each f_i is a closed subset of U_i and $\sum_i f_i(m) = 1$ for all $m \in M$. If $\mathcal{A} = \{(V_\alpha, \varphi_\alpha)\}$ is an atlas on M, a *partition of unity subordinate to* $\mathcal{A}$ is a partition of unity $\{(U_i, f_i)\}$ such that for each set U_i, $U_i \subset V_{\alpha(i)}$.

We can think of a partition of unity as a covering of coordinate patches with a set of functions mapping to the real line that are normalised in the sense that the sum of the functions at any point is 1.

We now define the integral of a differential form on an open set of $\mathbb{R}^n$.

Definition 2.3.42. Let $\omega \in \bigwedge^n U$ have compact support for $U \subset \mathbb{R}^n$ open. Then we can write $\omega(x) = \omega_{1\cdots n}(x)\mathrm{d}x^1 \wedge \ldots \wedge \mathrm{d}x^n$. Then the *integral of* ω *over* U is given by the Riemann integral

$$\int_U \omega = \int_{\mathbb{R}^n} \omega_{1\cdots n}(x)\, \mathrm{d}x^1 \ldots \mathrm{d}x^n.$$

So integrating a volume form over a subset of $\mathbb{R}^n$ reduces to the usual Riemann integral of the components of the form over $\mathbb{R}^n$.

Definition 2.3.43. Let M be an oriented manifold and $\mathcal{A}$ be an atlas of positively oriented charts. Let $P = \{U_i, f_i\}$ be a partition of unity subordinate to $\mathcal{A}$. Let ω be a volume form on M and $\omega_i = f_i\omega$ so that ω_i is compactly supported on U_i, then

$$\int_P \omega = \sum_i \int_{U_i} \omega_i.$$

Moreover, for any other atlas $\mathcal{B}$ and subordinate partition of unity Q, $\int_P \omega = \int_Q \omega$. Then the *integral* of ω is

$$\int_M \omega = \int_P \omega.$$

So we can define the integral of a volume form over a manifold by considering any atlas and subordinate partition of unity. Thus we can think of the integration of a volume form as the sum of the Riemann integrals of the volume form on each local coordinate chart.

We now give a formal definition of the boundary of a subset of $\mathbb{R}^n$ before defining the boundary of a manifold. This will be required for the statement of Stokes' theorem, which is the generalisation to manifolds of the Green's identities we used in section 2.2.1.

Definition 2.3.44. Let λ be some linear functional on $\mathbb{R}^n$. We call the space $\mathbb{R}^n_\lambda = \{x \in \mathbb{R}^n : \lambda(x) \geq 0\}$ a *half-space* of $\mathbb{R}^n$. We denote by $\mathbb{R}^n_+$ the positive nth half-space with λ the projection on x^n.

Definition 2.3.45. Let $U \subset \mathbb{R}^n_\lambda$ be open, then the *interior* of U is given by $\operatorname{Int} U = U \cap \{x \in \mathbb{R}^n : \lambda(x) > 0\}$ and the *boundary* of U is given by $\partial U = U \cap \ker \lambda$.

We can think of the function λ as a level set function whose zero level set defines the boundary of the subspace U. We now give a few definitions that lead up to the definition of a manifold with boundary. In the following let M be an n-manifold.

Definition 2.3.46. A *chart with boundary* is a coordinate chart (U, φ) on M, such that $\varphi(U) \subset \mathbb{R}^n_\lambda$.

Definition 2.3.47. An *atlas with boundary* is an atlas on M whose charts are charts with boundary.

Definition 2.3.48. A *manifold with boundary* is a topological space with an atlas of charts with boundary on M. We denote the boundary of M, a $(n-1)$-manifold, by ∂M.

So a manifold with boundary is a manifold whose coordinate charts have boundary determined by the level set λ.

Before stating the very important result given by Stokes' theorem we define the following map, that maps volume forms on a manifold boundary to the manifold.

Definition 2.3.49. The *inclusion map* is the map $i : \partial M \to M$.

We now state Stokes' theorem for the integration of volume forms on a differentiable manifold.

Theorem 2.3.1. *Let M be an oriented n-manifold with boundary and let $\omega \in \bigwedge^{n-1} M$ have compact support. Then*

$$\int_M \mathrm{d}\omega = \int_{\partial M} i^*\omega.$$

We usually drop the inclusion map notation and just write

$$\int_M \mathrm{d}\omega = \int_{\partial M} \omega.$$

We give one last definition before we proceed to the weak formulation of the geometric EIT problem.

Definition 2.3.50. The L^2 *inner product* of two k-forms on M is given by

$$\langle \alpha, \beta \rangle_{L^2} = \int_M \alpha \wedge \star\beta.$$

We now let M be a Riemannian manifold with metric g_{ij}. Let $u \in \bigwedge^0 M$ be a harmonic function, that is, u satisfies the geometric formulation of the EIT problem in (2.7). Then taking the inner product with the 0-form v we have

$$0 = \langle v, \Delta u \rangle_{L^2} = \langle v, \delta \mathrm{d} u \rangle_{L^2} = \int_M v \wedge \star\delta \mathrm{d} u.$$

Given a 0-form α and a 1-form β, from definition 2.3.29 we have

$$\begin{aligned} \mathrm{d}(\alpha \wedge \star\beta) &= \mathrm{d}\alpha \wedge \star\beta + \alpha \wedge \mathrm{d} \star \beta, \\ &= \mathrm{d}\alpha \wedge \star\beta - \alpha \wedge \star\delta\beta, \end{aligned}$$

so

$$\alpha \wedge \star\delta\beta = \mathrm{d}\alpha \wedge \star\beta - \mathrm{d}(\alpha \wedge \star\beta).$$

Then

$$\begin{aligned} 0 &= \langle v, \Delta u \rangle_{L^2}, \\ &= \langle \mathrm{d}v, \mathrm{d}u \rangle_{L^2} - \int_M \mathrm{d}(v \wedge \star \mathrm{d}u), \\ &= \langle \mathrm{d}v, \mathrm{d}u \rangle_{L^2} - \int_{\partial M} v \wedge \star \mathrm{d}u, \end{aligned}$$

by Stokes' theorem. Then the weak form of the geometric EIT problem is

$$\int_M \mathrm{d}v \wedge \star \mathrm{d}u = \int_{\partial M} v \wedge \star \mathrm{d}u, \quad \forall v \in \bigwedge^0 M. \tag{2.35}$$

If we compare this to the expression in (2.19) we can see that the conductivity σ acts as a Hodge star operator.

The reason for posing the problem in this way will become clearer when we discuss discrete differential geometry in section 2.4 and when we come to apply these ideas in chapter 4.

2.3.7 Curvature

In chapter 3 we will want to measure how much an n-manifold differs from $\mathbb{R}^N$. The concept of curvature is such a measure and can be thought of as the rate of change of tangent vectors defined on the manifold. In this section we give some formal definitions relating to curvature. We note that [41] has some good introductory material on this topic.

Recall from section 2.3.2 that we define a vector field in the tangent bundle as an assignment of a vector $X(p)$ at each point $p \in M$, where each $p \in M$ is the realisation of a differentiable curve $c : [a, b] \to M$ for a particular value, that is, $p = c(t)$ for $t \in [a, b]$.

In the following, let $p \in M$ be a point, Y be a tangent vector field to M along the curve c in some neighbourhood of p and $X = X(p) \in M_p$ a vector tangent to any curve through p. Let $x = (x^1, \ldots, x^n)$ be a local coordinate system for M at p. Let ∂_i denote the basis vector $\partial/\partial x^i$.

Definition 2.3.51. The *Levi-Civita connection* $\nabla(X, Y) \mapsto \nabla_X Y$ is given by the *covariant derivative* of the vector field Y at p. This is given by

$$\nabla_X Y = \left(X^i Y^j \Gamma^k_{ij} + X^i \frac{\partial Y^k}{\partial x^i} \right) \frac{\partial}{\partial x^k}, \tag{2.36}$$

where the Γ^k_{ij} are called *Christoffel symbols* and are given by

$$\Gamma^k_{ij} = \frac{1}{2} \left(\frac{\partial}{\partial x^i} g_{jl} + \frac{\partial}{\partial x^j} g_{li} - \frac{\partial}{\partial x^l} g_{ij} \right) g^{lk}.$$

In particular,

$$\nabla_{\partial_i} \partial_j = \Gamma^k_{ij} \partial_k.$$

If we have a Euclidean coordinate system with curve c parameterised by t, and $p_0 = c(0)$, then (2.36) reduces to

$$\nabla_X Y = X^i \frac{\partial Y^k}{\partial x^i} \frac{\partial}{\partial x^k} = X(p_0) \left(Y^k \frac{\partial}{\partial x^k} \right) = X(p_0) \left. \frac{\mathrm{d}Y}{\mathrm{d}t} \right|_{t=0},$$

which is just the tangential component of the velocity of vector Y. So the expression involving the Christoffel symbols takes care of the non-Euclidean nature of the manifold M.

We can now define curvature in terms of the Levi-Civita connection.

Definition 2.3.52. The *Riemann curvature* of the Riemannian manifold M is the operator $R(X,Y) : TM \to TM$ given by

$$R(X,Y)Z = \nabla_Y \nabla_X Z - \nabla_X \nabla_Y Z + \nabla_{[X,Y]} Z,$$

where $[X,Y]$ is the *Lie bracket* defined as the following vector field

$$[X,Y] = (XY - YX) = \left(X^j \frac{\partial Y^i}{\partial x^j} - Y^j \frac{\partial X^i}{\partial x^j} \right) \frac{\partial}{\partial x^i}.$$

We define the *Riemann curvature tensor* as the tensor R^l_{ijk} such that

$$\begin{aligned} R(\partial_i, \partial_j)\partial_k &= \nabla_{\partial_j} \nabla_{\partial_i} \partial_k - \nabla_{\partial_i} \nabla_{\partial_j} \partial_k, \\ &= \nabla_{\partial_j} \left(\Gamma^m_{ik} \partial_m \right) - \nabla_{\partial_i} \left(\Gamma^m_{jk} \partial_m \right), \\ &= \left(\Gamma^l_{jm} \Gamma^m_{ik} - \Gamma^m_{jk} \Gamma^l_{im} + \frac{\partial}{\partial x^j} \Gamma^l_{ik} - \frac{\partial}{\partial x^i} \Gamma^l_{jk} \right) \partial_l, \\ &=: R^l_{ijk} \partial_l. \end{aligned}$$

Definition 2.3.53. The *Ricci curvature tensor* is the contraction of the Riemann curvature tensor, that is,

$$\begin{aligned} R_{ij} &:= R^k_{ikj}, \\ &= \Gamma^k_{jm} \Gamma^m_{ik} - \Gamma^m_{jk} \Gamma^k_{im} + \frac{\partial}{\partial x^j} \Gamma^k_{ik} - \frac{\partial}{\partial x^i} \Gamma^k_{jk}. \end{aligned}$$

So the Ricci curvature tensor gives us a way of measuring how the metric g_{ij} varies in each direction throughout the manifold. Clearly, if $g_{ij}(x) = \delta_{ij}$ for all $x \in M$, then $R_{ij} = 0$. To give a scalar value that measures the overall curvature at each point in the manifold we have the following definition.

Definition 2.3.54. The *(Ricci) scalar curvature* is the trace of the Ricci curvature tensor, that is,

$$R := g^{ij} R_{ij}.$$

So the scalar curvature is a weighted sum of the Ricci curvatures in each coordinate direction which provides an overall curvature value.

2.4 Algebraic topology and discrete differential geometry

In this section we will introduce some concepts from algebraic topology related to discrete differential geometry, which will be useful when we come to solve some real problems defined on manifolds. For a discussion of the topological concepts introduced in this section, along with many other related topics and more generalised concepts, the reader is referred to the books [22] and [11]. For a more direct introduction to this theory applied to physical problems, see [46], and for electromagnetic problems specifically, see [49]. For an excellent introduction to differential forms in a discrete setting the reader is referred to the book chapter [39].

2.4.1 Chains and simplicial complexes

Suppose we want to discretise a manifold, then we need a way of breaking the space into discrete finite pieces that approximate the manifold. We call these pieces *p-cells*, where p is the dimension of the object. We start by defining discrete points in the manifold which are 0-cells, then by inserting 1-cells between given pairs. We can then define 2-cells as the 2-dimensional pieces bounded by 1-cells, 3-cells bounded by 2-cells and so on until the dimension of the cells is equal to the dimension of the manifold. The structure defined by these cells is a *cell complex* or *CW complex*. We shall not give a more formal definition of these general complexes since we only really need to keep in mind the structure we described above. However, we give a formal definition of a smaller class of cell complexes that we will use in our work.

Definition 2.4.1. The *standard p-simplex* is the set

$$\Delta_p = \left\{ (\lambda_0, \ldots, \lambda_p) \in \mathbb{R}^{p+1} : \sum_{i=0}^{p} \lambda_i = 1,\ 0 \leq \lambda_i \leq 1 \right\},$$

where the λ_i are called *barycentric coordinates*.

Definition 2.4.2. A *singular p-simplex* is a map $\sigma : \Delta_p \to \mathbb{R}^n$ that assigns coordinates to the points in Δ_p. Then

$$\sigma = \left\{ x \in \mathbb{R}^n : x = \sum_{i=0}^{p} \lambda_i v_i,\ v_1, \ldots, v_p \in \mathbb{R}^n \right\},$$

is the convex hull of the *vertices* $v_0, \ldots, v_p$.

We shall denote a simplex by $\sigma = \{v_0, \ldots, v_p\}$ or simply $\{0, \ldots, p\}$. As an example, a 3-simplex is a tetrahedron defined by its four vertices. A triangular face of the tetrahedron is a 2-simplex and an edge is a 1-simplex. We define a vertex as a 0-simplex.

Definition 2.4.3. A $(p-1)$-*face* of a p-simplex is a $(p-1)$-simplex whose set of vertices is a subset of the vertices of the p-simplex.

For example, a triangular face of a tetrahedron is a 2-face. An edge of a triangular face is a 1-face.

Definition 2.4.4. A *simplicial complex* is a collection $\mathcal{K}$ of simplices such that every face of each simplex is in $\mathcal{K}$, and the intersection of two simplices is either empty or a common face.

Definition 2.4.5. Given a simplicial complex $\mathcal{K}$ of dimension n, the p-*skeleton* $\mathcal{S}_p$ is the simplicial complex formed by the q-simplices, for $q \leq p$.

As an example of the above definition, consider a single tetrahedron. This can be thought of as a simplicial complex formed by all the p-simplices for $p = 0, 1, 2, 3$. The 2-skeleton is then just the simplicial complex defined by only the vertices, edges and faces of the tetrahedron.

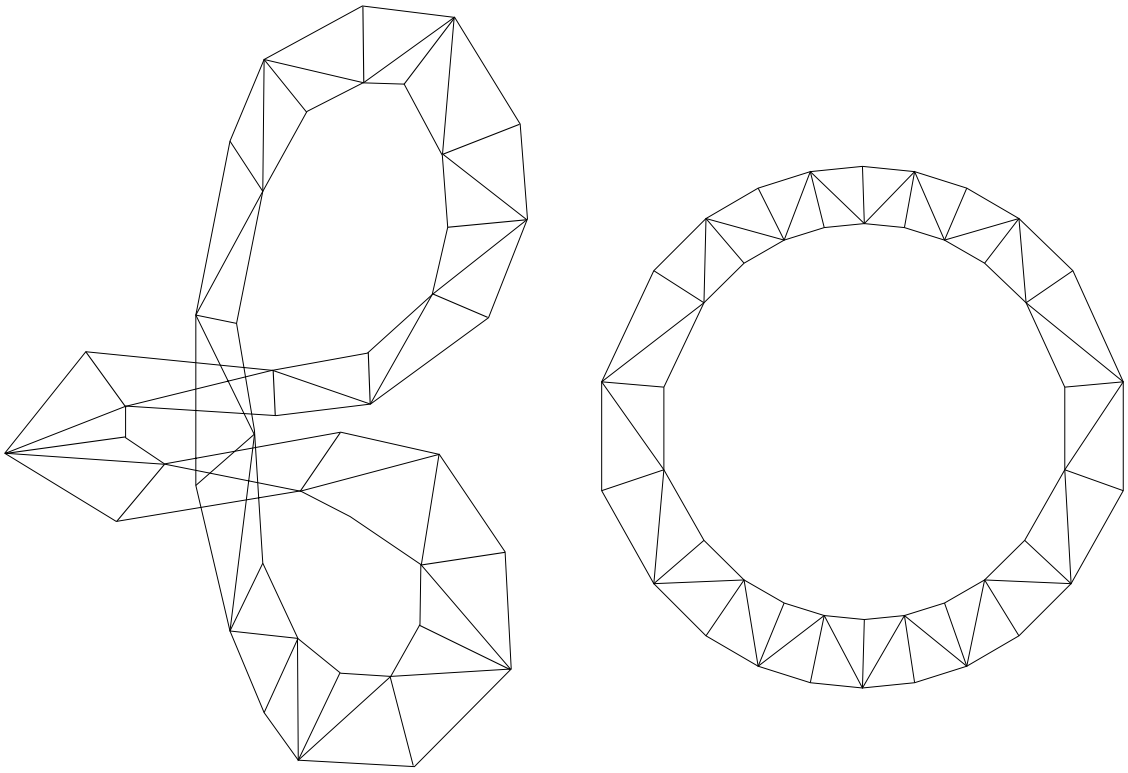
As for manifolds, an important concept that needs consideration is that of orientation which we define for a simplex below.

Definition 2.4.6. We define an equivalence class $[\tau]$ on orderings of the vertices of a p-simplex by saying two orderings τ_1 and τ_2 are equivalent if τ_2 is an even permutation of τ_1. An equivalence class $[\tau]$ is an *orientation*.

As an example, consider the 2-simplex $\{v_0, v_1, v_2\}$. Then $\{v_0, v_1, v_2\} = \{v_1, v_2, v_0\}$ and $\{v_0, v_1, v_2\} = -\{v_0, v_2, v_1\}$.

Definition 2.4.7. We define the *boundary* of a p-simplex σ to be the signed sum of the $(p-1)$-faces of σ. Let $\sigma = \{v_0, \ldots, v_p\}$, then the *boundary operator* ∂_p maps σ to its $(p-1)$-faces as follows

$$\partial_p \sigma = \sum_{i=0}^{p} (-1)^i \{v_0, \ldots, \hat{v}_i, \ldots, v_p\},$$



(a) Immersion that is not an embedding. (b) Immersion that is an embedding.

Figure 2.2: Examples of immersions of planar piecewise flat triangulated manifolds in $\mathbb{R}^2$.

where the notation $\hat{v}_i$ signifies that v_i is omitted from the sequence.

We now introduce the concept of a triangulated manifold. By triangulation of a manifold we mean that the manifold is represented by a simplicial complex embedded in the manifold. We use the definition given in [47] since it suits our needs and is fairly simple to picture.

Definition 2.4.8. A *triangulated manifold* $(M, \mathcal{K})$ is a topological n-manifold M together with a triangulation $\mathcal{K}$ of M. A *piecewise flat triangulated manifold* $(M, \mathcal{K}, l)$ is a triangulated manifold together with a positive function l defined on the 1-faces of the triangulation such that the simplices can be embedded in Euclidean space with embedded edge lengths given by l.

As noted in [30], a piecewise flat triangulated manifold is an alternative name for a piecewise linear triangulated manifold. We will use the terms discrete manifold and triangulated manifold interchangeably although a discrete manifold in general could be represented by a partition into any polyhedra. It is worthwhile giving an example

of an embedded triangulation here since we defined the somewhat abstract notions of immersions and embeddings earlier (definitions 2.3.13 and 2.3.14) without giving an example. The two-dimensional piecewise flat triangulated manifolds shown in figure 2.2 are examples of immersions in $\mathbb{R}^2$. The example in 2.2a is an immersion since the map between tangent spaces is locally injective but it is not an embedding because the map to the plane is not globally injective, therefore it is not a homeomorphism. The example in 2.2b is an immersion for the same reaosns as the previous example. It is also an embedding since the map to the plane is a continuous bijection with continuous inverse.

We now move on to the notion of chains, an important concept in discrete differential geometry.

Definition 2.4.9. A *p-chain* on an oriented simplicial complex $\mathcal{K}$ is a linear combination of the p-simplices in $\mathcal{K}$. We denote the group of p-chains on $\mathcal{K}$ by $\mathcal{C}_p(\mathcal{K})$. Then $c \in \mathcal{C}_p(\mathcal{K})$ is given by

$$c = \sum_{\sigma \in \mathcal{K}} c_\sigma \sigma,$$

with coefficients $c_\sigma \in \mathbb{R}$.

We can see that the boundary ∂_p of a p-simplex is a $(p-1)$-chain on the $(p-1)$-faces with coefficients -1 or 1. In fact, we can extend the definition of the boundary operator so that it can be applied to p-chains as

$$\partial_p c = \partial_p \sum_{\sigma \in \mathcal{K}} c_\sigma \sigma = \sum_{\sigma \in \mathcal{K}} c_\sigma \partial_p \sigma.$$

Then the boundary operator defines a mapping $\partial_p : \mathcal{C}_p(\mathcal{K}) \to \mathcal{C}_{p-1}(\mathcal{K})$ from p-chains to $(p-1)$-chains, with $\partial_p = 0$, for $p \leq 0$. We have the following important result regarding the boundary operator applied to chains.

Lemma 2.4.1. *Let $\partial_p : \mathcal{C}_p(\mathcal{K}) \to \mathcal{C}_{p-1}(\mathcal{K})$ be the boundary operator on p-chains. Then*

$$\partial_{p-1} \partial_p = 0.$$

Proof. If we apply the operator twice to a p-simplex σ_p we have

$$\partial_{p-1} \partial_p \sigma = \partial_{p-1} \left(\sum_{\sigma_{p-1} \in \operatorname{im} \partial_p} \sigma_{p-1} \right)$$

which is the sum over each $(p-2)$-simplex on σ_p twice with opposite sign. Since the boundary of a chain is a linear combination of boundaries of simplices, the result follows. □

Definition 2.4.10. A *p-cycle* is a p-chain c such that $\partial_p c = 0$. That is, a p-cycle is in the kernel of ∂_p.

This is a fairly intuitive definition if we consider 1-cycles, since these are closed loops on 1-simplices. For example, consider the oriented 2-simplex with vertices $\{v_0, v_1, v_2\}$. If we define the oriented edges $e_0 = \{v_1, v_2\}$, $e_1 = \{v_2, v_0\}$, $e_2 = \{v_0, v_1\}$, then the 1-chain

$$c = \sum_{i=0}^{2} e_i$$

is a 1-cycle since

$$\partial_1 c = \sum_{i=0}^{2} \partial_1 e_i = (v_2 - v_1) + (v_0 - v_2) + (v_1 - v_0) = 0.$$

Definition 2.4.11. A *p-boundary* is a p-chain that is the boundary of a $(p+1)$-chain. That is, a p-boundary is in the image of ∂_{p+1}.

In the example above, we see that the 0-chain $v_2 - v_1$ is a 0-boundary since $\partial_1 e_0 = v_2 - v_1$.

The series of p-chain spaces related by the boundary operator can be represented by the following notation:

$$0 \longrightarrow \mathcal{C}_n \xrightarrow{\partial_n} \mathcal{C}_{n-1} \xrightarrow{\partial_{n-1}} \ldots \xrightarrow{\partial_2} \mathcal{C}_1 \xrightarrow{\partial_1} \mathcal{C}_0 \xrightarrow{\partial_0} 0,$$

which defines a *chain complex*. We now introduce the concept of homology that relates p-cycles to p-boundaries in a chain complex. This will be useful in subsequent chapters when we need to find the independent cycles of p-chains.

Definition 2.4.12. We define an equivalence class $[c]$ on p-chains by setting $c_1 \sim c_2$ if $c_1 - c_2 = \partial_{p+1} c_3$, where $c_1, c_2 \in \mathcal{C}_p(\mathcal{K})$ and $c_3 \in \mathcal{C}_{p+1}(\mathcal{K})$. If c_1 and c_2 are p-cycles then we say they are *homologous*. The quotient group

$$H_p(\mathcal{K}) = \ker \partial_p / \text{im } \partial_{p+1}$$

is called the *pth simplicial homology group* of $\mathcal{K}$.

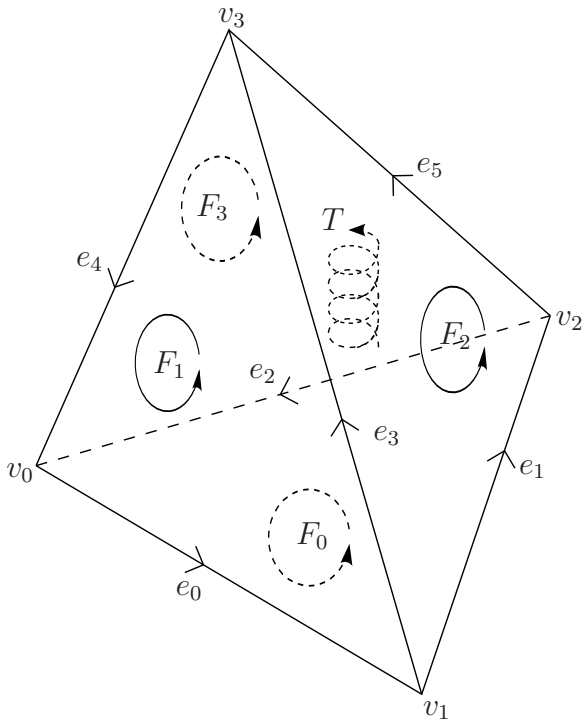


Figure 2.3: Oriented 3-simplex T.

So two p-cycles are homologous if they differ by the boundary of a $(p+1)$-chain. This concept will be useful later when we need to find independent cycles in graphs defined on simplicial complexes.

Definition 2.4.13. The dimension of the pth homology group $H_p(\mathcal{K})$ is called the *pth Betti number* which we denote

$$\beta_p = \dim H_p(\mathcal{K}).$$

The Betti numbers are topologically invariant, that is, they are invariant under homeomorphisms.

Definition 2.4.14. We define the*Euler characteristic* of a manifold M as the topological invariant

$$\chi(M) = \sum_p (-1)^p \beta_p(\mathcal{K}), \tag{2.37}$$

where $\mathcal{K}$ is some simplicial complex representing M.

As a simple example, consider the oriented 3-simplex depicted in figure 2.3. It is homeomorphic to the 3-ball so it will have the same Euler characteristic. We start by computing H_0. Since $\partial v_i = 0$ for $i = 0, 1, 2, 3$, every 0-chain is a cycle. Since the

difference of any two 0-cycles is the boundary of an edge, each 0-cycle is homologous to any other 0-cycle so $H_1 = \langle v_0 \rangle$, where $X = \langle x_1, \ldots, x_n \rangle$ denotes that $x_1, \ldots, x_n$ span the space X. Thus $\beta_0 = 1$. If there were multiple components, for example, 2 separate 3-simplices with no edges between them, then we would have another equivalence class generated by one of the vertices in the second 3-simplex, in which case we would have $\beta_0 = 2$. Extending this argument inductively, we can see that β_0 is the number of connected components of a manifold.

For H_1, we consider 1-cycles. We can see from the figure that any 1-cycle will be the boundary of a 2-chain, therefore $\ker \partial_1 = \text{im } \partial_2$ and $\beta_1 = 0$. For H_2, we see that multiples of $\partial_3 T = F_0 + F_1 + F_2 + F_3$ are the only boundaries of 3-chains since T is the only 3-simplex. But the only 2-cycles are multiples of $F_0 + F_1 + F_2 + F_3$, therefore $\ker \partial_2 = \text{im } \partial_3$ and $\beta_2 = 0$. For H_3, there are no 3-cycles so $\beta_3 = 0$. Since there are no p-chains for $p > 3$, $H_p = 0$ for $p > 3$. Then the Euler characteristic of a solid 3-simplex, and therefore the solid three-dimensional ball, is $\chi = 1$.

The famous Euler-Poincaré theorem gives another way to compute the Euler characteristic of a cell complex. We state it here for reference.

Theorem 2.4.1. *Let $\mathcal{K}$ be a cell complex and let k_i be the number of i-cells in $\mathcal{K}$. Then the Euler characteristic is given by the alternating sum*

$$\chi(\mathcal{K}) = \sum_i (-1)^i k_i.$$

This means that it if we can find all but one of the Betti numbers in a simple way, then we can compute the Euler characteristic using the above formula and find the unknown Betti number by rearranging (2.37).

2.4.2 Cochains and integration on discrete manifolds

In this section we will introduce the discrete counterparts of forms and the evaluation of these objects via integration.

Definition 2.4.15. A *p-cochain* ω is a linear functional that maps p-chains to scalars. That is, p-cochains are dual to p-chains. Denote by $\mathcal{C}^p$ the space of p-cochains, then for $\omega \in \mathcal{C}^p$,

$$\omega : \mathcal{C}_p \to \mathbb{R},$$
$$c \mapsto \omega(c).$$

Recall for the continuous case, that we evaluate a k-form by integrating over a k-dimensional (sub)manifold. In the discrete setting we evaluate a p-cochain by integrating over a p-chain. Let $\omega \in \mathcal{C}^p(\mathcal{K})$, $c \in \mathcal{C}_p(\mathcal{K})$, then denote by $[\omega, c]$ the dual pairing between p-cochains and p-chains where

$$\omega(c) = [\omega, c] = \int_c \omega = \int_{\sum_i c_i \sigma_i} \omega = \sum_{\sigma \in \mathcal{K}} c_\sigma \int_\sigma \omega.$$

The last integral in the expression above is the integration of a p-form ω over simplex σ which gives a discretisation of the p-form as a value on each p-simplex. Then taking the linear combination of these integrals over each simplex results in the scalar value $\omega(c)$.

We now introduce the discrete counterpart to the exterior derivative. We use the discrete version of Stokes theorem (theorem 2.3.1) to define this discrete exterior derivative which gives a straightforward definition in terms of concepts already introduced and ensures that Stokes' theorem is satisfied automatically.

Definition 2.4.16. The *coboundary* operator denoted $\mathrm{d} : \mathcal{C}^p(\mathcal{K}) \to \mathcal{C}^{p+1}(\mathcal{K})$ is the adjoint of the boundary operator ∂. If $[\omega, c]$ is the dual pairing between p-cochains and p-chains, then the coboundary operator is defined as

$$[\mathrm{d}\omega, c] = [\omega, \partial c]\,.$$

Since d is dual to ∂, we also have $\mathrm{dd} = 0$. Clearly the above definition of d is just Stokes' theorem in a discrete setting. To see how this would be implemented we have

$$[\mathrm{d}\omega, c] = \int_c \mathrm{d}\omega = \int_{\sum_i c_i \sigma_i} \mathrm{d}\omega = \int_{\partial \sum_i c_i \sigma_i} \omega = \int_{\sum_i c_i \partial \sigma_i} \omega = \sum_{\sigma \in \mathcal{K}} c_\sigma \int_{\partial \sigma} \omega = [\omega, \partial c]\,.$$

Now that we have defined the dual of the boundary operator we have all the tools required to define the dual notion of the homology groups. We can define the cochain complex

$$0 \xleftarrow{\mathrm{d}^n} \mathcal{C}^n \xleftarrow{\mathrm{d}^{n-1}} \ldots \xleftarrow{\mathrm{d}^1} \mathcal{C}^1 \xleftarrow{\mathrm{d}^0} \mathcal{C}^0 \longleftarrow 0.$$

Then we can define equivalence classes using d as we did for ∂.

Definition 2.4.17. The pth *cohomology group* $H^p(\mathcal{K})$ of $\mathcal{K}$ is given by the quotient group

$$H^p(\mathcal{K}) = \ker \mathrm{d}^p / \mathrm{im}\, \mathrm{d}^{p-1}.$$

Before we go any further we need to introduce the concept of duality with respect to the discretised manifold.

Definition 2.4.18. The *dual complex* $\mathcal{K}^*$ of a simplicial complex $\mathcal{K}$ is the cell complex constructed by defining an $(n-p)$-cell σ^* for each p-simplex $\sigma \in \mathcal{K}$.

In the following, we restrict ourselves to triangulations that are "well-centred", that is, the circumcentre of each n-simplex lies within the simplex. It is known that the dual cells of well-centred meshes are orthogonal, that is, the dual of an interior primal 0-simplex is a convex polygon, [94], which is important in the following definition. The dual cells can be defined in many ways, the most obvious being the circumcentric definition. The circumcentric dual complex is constructed by computing the circumcentre of each n-simplex of $\mathcal{K}$, then each dual p-cell is the convex hull of the n-simplex circumcentres that share the *primal* $(n-p)$-face. For example, in three dimensions, the dual 0-cells are the circumcentres of the 3-simplices, dual 1-cells are the lines joining the circumcentres of the two 3-simplices that share the primal 2-face, a dual 2-cell is the two dimensional convex hull of the 3-simplex circumcentres that share the corresponding primal edge and a dual 3-cell is the volume constructed by the convex hull of the circumcentres of the surrounding 3-simplices.

We introduce this concept here since it is required for introducing a particular form of the wedge product. Many defintions of the discrete wedge product are introduced in [55], but since we are interested in the wedge product of a form and a Hodge star applied to a form we require the notion of the primal-dual wedge product.

Definition 2.4.19. Let $\alpha \in \mathcal{C}^p(\mathcal{K})$ be a p-cochain and $\beta \in \mathcal{C}^{n-p}(\mathcal{K}^*)$ be a dual $(n-p)$-cochain. The *primal-dual wedge product* denoted $\alpha \wedge \beta$ is an n-cochain defined on the union of p-simplices and dual $(n-p)$-cells with the same properties as for the continuous wedge product given in definition 2.3.27.

To illustrate this definition consider two triangles sharing an edge e in a two-dimensional simplicial complex $\mathcal{K}$. Let $\alpha \in \mathcal{C}^1(\mathcal{K})$ and $\beta \in \mathcal{C}^1(\mathcal{K}^*)$ be given by

constants a and b on edge e and zero elsewhere. Then the wedge product $\alpha \wedge \beta$ is defined on the set D, the convex hull of the two endpoints of e and the two circumcentres of the triangles sharing edge e, that is, the endpoints of the dual edge e^*. Let $c \in \mathcal{C}_1(\mathcal{K})$ be the 1-chain with value 1 on every primal edge, and $c' \in \mathcal{C}_1(\mathcal{K}^*)$ be the 1-chain with value 1 on every dual edge. Since the integral of $\alpha \wedge \beta$ is zero over all edges other than e the dual pairing $[\alpha \wedge \beta, c]$ is given by

$$\begin{aligned}[\alpha \wedge \beta, c] &= \sum_i c_i c_i' \int_D \alpha \wedge \beta, \\ &= ab.\end{aligned}$$

We now introduce the last discretised concept that is present in our work. Recall from definition 2.3.35 that the Hodge star operator is defined in terms of the wedge product and inner product of two k-forms. In the discrete case we require the Hodge star to map p-cochains to $(n-p)$-cochains and we do this by mapping to the dual cochain. We use the following definition given in [55] which defines the Hodge star in terms of the dual pairing between chains and cochains.

Definition 2.4.20. The *discrete Hodge star* defined on a simplicial complex $\mathcal{K}$ is the map $\star : \mathcal{C}^p(\mathcal{K}) \to \mathcal{C}^{n-p}(\mathcal{K}^*)$ that satisfies the following relation on each p-simplex σ,

$$\frac{1}{|\sigma^*|} [\star\omega, \sigma^*] = \frac{1}{|\sigma|} [\omega, \sigma],$$

for a p-cochain $\omega \in \mathcal{C}^p(\mathcal{K})$.

In the above definitions we have applied the dual pairing of cochains and chains to a cochain and a simplex. To extend the definition to a chain defined on every p-simplex we would take the linear combination of the above expression on each simplex, where the coefficients of the linear combination are given by the coefficients of the chain as usual.

2.5 Graph Theory

In this section we introduce some basic concepts from graph theory which will be useful in conjunction with the previous section. Where appropriate we will draw attention to analogous concepts from the previous section. The material in this section is covered in more detail and with more generality in the books [17] and [61].

Definition 2.5.1. We define a *graph* G as the pair $G = (V, E)$, where $V = \{v_1, \ldots, v_{n_V}\}$ is the set of *vertices* and $E \subset \{\{v_i, v_j\} : v_i, v_j \in V\}$ is the set of *edges*. If the pair $\{v_i, v_j\} \in E$ then we say that v_i and v_j are *adjacent*.

We can see from this definition that the elements of E are unordered pairs of vertices. For a general graph, an edge may join a vertex to itself but for our work we will only consider graphs with the extra stipulation that $E \subset \{\{v_i, v_j\} : v_i, v_j \in V,\ i \neq j\}$, that is, an edge is an unordered pair of distinct vertices.

So we can think of a graph as an unoriented simplicial complex of 0-simplices and 1-simplices. This satisfies the definition of a simplicial complex since every 0-face of an edge is just a vertex which is clearly in the complex, and the intersection of two edges is a common vertex or is empty.

We now define some notions related to sequences of edges in a graph. For the following definitions let $W = (e_1, \ldots, e_n)$ with $e_i \in E$, be a sequence of edges in G.

Definition 2.5.2. If there exist vertices $v_0, \ldots, v_n \in V$ such that $e_i = \{v_{i-1}, v_i\} \in E$ for $i = 1, \ldots, n$, then the sequence W is called a *walk*. If $v_0 = v_n$ then W is a *closed walk*.

Definition 2.5.3. If the edges in W are distinct then W is a *trail*. If $v_0 = v_n$ then W is a *closed trail*.

Definition 2.5.4. If the $v_0, \ldots, v_n$ are distinct then W is called a *path* and if $v_0 = v_n$ with $v_0, \ldots, v_{n-1}$ distinct then the path is a *cycle*.

We can see from the last definition that a cycle in a graph is a specific case of a 1-cycle when defined in terms of chains. We can think of a 1-chain as a union of trails with weights and orientations. It is a union of trails because we only include each edge once since we take a linear combination of the edges to produce a chain.

We can relate the cycles of a graph to the first homology group of the simplicial complex formed by the vertices and edges of the graph. Recall that $H_1(G) = \ker \partial_1 / \operatorname{im} \partial_2$, then since there are no 2-simplices $\operatorname{im} \partial_2 = \emptyset$, so $H_1(G) = \ker \partial_1$. But $\ker \partial_1$ is just the space of 1-cycles, so $\beta_1(G) = \dim H_1(G)$ is just the number of independent cycles in the graph G. This is also known as the *cyclomatic number* and there is a theorem which gives an alternative formula for the cyclomatic number, but we require a few more related concepts before stating this theorem.

Definition 2.5.5. Two vertices $u, v \in V$ are *connected* if there exists a walk such that the start and end vertices are u and v respectively. If all pairs of vertices of a graph are connected then the graph is called connected.

Definition 2.5.6. We define an equivalence class on a graph G by saying that vertices u and v are equivalent if they are connected. The equivalence classes defined by this relation are called *connected components* of G.

Definition 2.5.7. A graph G with no cycles is called *acyclic*.

Definition 2.5.8. A *tree* is a graph that is connected and acyclic.

Definition 2.5.9. A *spanning tree* $T = (V_T, E_T)$ of a graph $G = (V, E)$ is a subgraph of G that is a tree and contains every vertex of G. That is, $V_T = V$ and $E_T \subseteq E$.

The following theorem is stated as a corollary in [17] but we state it here since it is important for our later work.

Theorem 2.5.1. *Every connected graph has a spanning tree.*

Proof. To create a spanning tree T we simply remove edges from the graph. For every edge removed, we either remove a cycle or the graph is no longer connected. Therefore only removing edges that are components of a cycle until it is acyclic ensures that the graph stays connected. □

Let $Z(G)$ denote the cycle space of the graph G, then the cyclomatic number which is simply $\dim Z(G)$ is given by the following theorem (see [17] for a proof).

Theorem 2.5.2. *For a graph G with k connected components, m edges and n vertices, the cyclomatic number is given by*

$$\dim Z(G) = m - n + k.$$

2.6 Singular Value Decomposition

In this section we provide an introduction to the singular value decomposition (SVD), which is a useful tool from linear algebra. The theory in this section, including proofs of the stated theorems, can be found in the book [48]. Throughout this section let $A \in \mathbb{R}^{m \times n}$ be a real matrix. We introduce the SVD through the theorem that shows its existence.

Theorem 2.6.1. *There exist orthogonal matrices*

$$U = (u_1 \ \cdots \ u_m) \in \mathbb{R}^{m\times m}, \quad V = (v_1 \ \cdots \ v_n) \in \mathbb{R}^{n\times n}$$

such that

$$U^T AV = \Sigma = \ \mathrm{diag}(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{m\times n}, \quad p = \min\{m, n\},$$

where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$.

Definition 2.6.1. The *singular value decomposition (SVD)* of A is given by

$$A = U\Sigma V^T,$$

for matrices U, V and Σ defined in theorem 2.6.1. The values $\sigma_1, \ldots, \sigma_p$ are the *singular values* of A, the $u_1, \ldots, u_m$ are the left singular vectors of A and the $v_1, \ldots, v_n$ are the right singular vectors of A.

We note here that the singular values have no correspondence with the conductivity in EIT, but the notation σ_i for these values is standard in linear algebra texts.

The SVD is related to the eigensystems of AA^T and A^TA as we show in the following.

Corollary 2.6.1. *If* $m \geq n$ *then for* $i = 1, \ldots, n$

$$Av_i = \sigma_i u_i, \quad A^T u_i = \sigma_i v_i.$$

Following this we have

$$A^TAv_i = \sigma_i V\Sigma^T U^T u_i = \sigma_i V\Sigma^T e_i = \sigma_i^2 v_i,$$
$$AA^T u_i = \sigma_i U\Sigma V^T v_i = \sigma_i U\Sigma^T e_i = \sigma_i^2 u_i,$$

where e_i is the ith Euclidean basis vector. Hence (σ_i^2, v_i) and (σ_i^2, u_i) are eigenpairs of A^TA and AA^T respectively.

A useful property of the SVD is that it can be used to characterise the range and null space of a matrix as shown in the following.

Corollary 2.6.2. *If* A *has* r *positive singular values, then* $\mathrm{rank}\, A = r$ *and*

$$\mathrm{null}\, A = \mathrm{span}\{v_{r+1}, \ldots, v_n\},$$
$$\mathrm{range}\, A = \mathrm{span}\{u_1, \ldots, u_r\}.$$

Since the row space of A is orthogonal to the null space we can see that $\{v_1, \ldots, v_r\}$ is a basis for the row space of A.

2.7 Conclusion

In this chapter we have defined the EIT problem and reviewed some of the uniqueness and stability results from the literature. As the reader may have noticed, almost all these results are based on the continuous problem, yet when we attempt to solve the problem in practice we will work on a discretised problem. There has not been much work in attempting to extend the results for the continuous case to the more practical discrete problem. In the following chapters we will tackle some problems that attempt to bridge the gap between the theory of the continuous problem and what we can hope to achieve in the discrete problem. We will review some results in this area in the relevant chapters.

The rest of this chapter gave an introduction to the mathematical theory that we will need to tackle the problems in the following chapters.

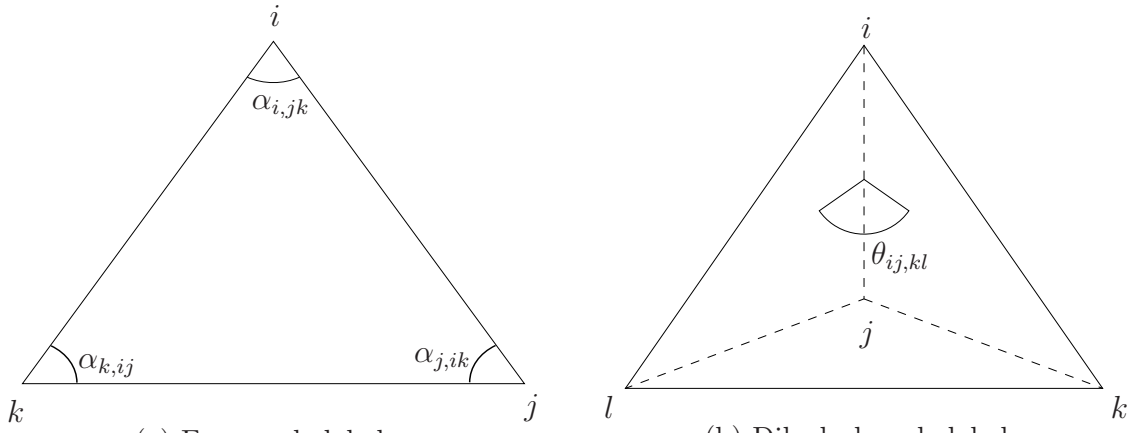
Chapter 3

A three-dimensional triangulation problem

3.1 Introduction

In this chapter we lay the foundations for finding an embedding of a finite element mesh associated with anisotropic conductivity in $\mathbb{R}^3$. Before we begin to add information related to the conductivity, we solve a three-dimensional triangulation problem which consists of constructing a triangulation given the dihedral angles of the tetrahedra that form the triangulation. The work in this chapter extends the two-dimensional embedding of a planar triangulation of Al-Humaidi [5], in which a nice review of the historical applications of such triangulation problems can be found. We therefore review this work before extending to the three-dimensional case.

When working in three dimensions we are required to find a tetrahedral mesh that satisfies certain consistency conditions, some of which are very different to those for the planar case. In this chapter, we list the consistency conditions that the dihedral angles must satisfy and prove that these conditions locally define a unique embedding. We then provide a constructive algorithm to compute the vertex positions given these angles. This algorithm is an extension of the algorithm outlined in [5].



(a) Face angle labels.

(b) Dihedral angle labels.

Figure 3.1: Illustration of angle notation.

3.2 Notation

Here we give an overview of the notation we will use throughout this chapter for both the planar and three-dimensional cases. Let $\mathcal{K}$ be a simplicial complex of the required dimension as in definition 2.4.4. So for the planar case $\mathcal{K}$ is a collection of 2-simplices and $\mathcal{K}$ is a collection of 3-simplices in the three-dimensional case. In both cases let V be the set of 0-simplices (vertices), E be the set of 1-simplices (edges) and F be the set of 2-simplices (triangles). In the three-dimensional case we denote the set of 3-simplices (tetrahedra) by T.

When we consider the embedding of the manifold that is represented by the simplicial complex, we use subscripts I and B to denote the set of interior and boundary k-simplices respectively. For example, the sets of interior and boundary 0-simplices are denoted V_I and V_B respectively, such that $V_I \cup V_B = V$.

Since we require some results from graph theory, we let $G = (V, E)$ be the graph whose vertices V and edges E are the 0-simplices and 1-simplices of $\mathcal{K}$ respectively.

We denote the face angle at vertex i in the 2-simplex $\{i, j, k\}$ by $\alpha_{i,jk}$ such that $\alpha_{i,jk} \in (0, \pi)$ for any i, j and k. For the three-dimensional case we also require dihedral angles which are denoted $\theta_{ij,kl}$ for the dihedral angle of edge $\{i, j\}$ in tetrahedron $\{i, j, k, l\}$, such that $\theta_{ij,kl} \in (0, \pi)$ for any i, j, k and l. This notation is illustrated in figure 3.1. We also denote by l_{ij} the length of edge $\{i, j\}$ when the graph is embedded in $\mathbb{R}^N$. That is, l is the nonnegative function $l : E \to \mathbb{R}_+$ such that

$$l_{ij} = l(\{i, j\}) = \|v_i - v_j\|_2,$$

where v_i is the position of vertex i in $\mathbb{R}^N$.

3.3 Planar triangulations

If we embed the simplicial complex in Euclidean space then we have a piecewise flat manifold $(M, \mathcal{K}, l)$ where l are the edge lengths. In this section we restrict ourselves to the case of $\mathbb{R}^2$. The problem in two dimensions is to find an embedding given a set of consistent face angles. The 1-skeleton of a single triangular face is geometrically self-dual, that is, a triangle with edges dual to vertices. Therefore in two dimensions we can think of face angles as being dual to edge lengths in a geometric sense.

In order to embed $\mathcal{K}$ in Euclidean space there are certain consistency conditions that must be met. We define the first of these in a moment, but first let us give some definitions. We define the curvature as in [47] which we give here for reference.

Definition 3.3.1. The scalar *curvature* R_i at vertex i is

$$R_i = 2\pi - \sum_{\{j,k\}\sim i} \alpha_{i,jk}, \tag{3.1}$$

where $\{j, k\} \sim i$ denotes that the edge $\{j, k\}$ shares a triangle with vertex i.

Note that the sum is over the triangles for which i is a vertex. We see that the curvature defined here is the discrete analogy to the scalar curvature defined in 2.3.54. Since we require the two-dimensional manifold to be embedded in $\mathbb{R}^2$, that is, we have a flat manifold, we require that the curvature be zero. Therefore our first consistency condition is

$$\sum_{\{j,k\}\sim i} \alpha_{i,jk} = 2\pi, \tag{3.2}$$

for all interior vertices $i \in V_I$.

For the next condition we must consider the space of cycles as defined in 2.4.1. Let us define the graph G' that is formed from the vertices and edges of the dual complex of $\mathcal{K}$. Then $G' = (F, E')$, since each face in F has a dual vertex, and $E' = \{\{f_i, f_j\} : f_i, f_j \in F,\ f_i \cap f_j \in E\}$[1]. Suppose we have a cycle in G' that encloses an interior vertex i and let $\{f_1, \ldots, f_k\}$ be the vertices of G' that form this cycle. In order

[1]This is not technically the dual graph of G, since this would also have an extra vertex that is not dual to any face and extra edges connecting this extra vertex to the vertices dual to faces containing boundary edges of G.

to ensure than the last simplex in the cycle intersects with the first simplex such that the definition of a simplicial complex is maintained (see definition 2.4.4), the angles opposite the enclosed vertex satisfy the following sine rule constraint

$$\sum_{\{j,k\}\sim i} (\ln \sin \alpha_{j,ki} - \ln \sin \alpha_{k,ij}) = 0. \tag{3.3}$$

Again we sum over the triangles $\{i, j, k\}$ that contain vertex i and we assume that the orientation of simplices is consistent.

We also have the obvious condition that all angles within a triangle sum to π. It is shown in [5] that these 3 conditions are necessary and sufficient to ensure that a planar simplicial complex can be embedded in $\mathbb{R}^2$.

3.4 The three-dimensional problem

3.4.1 Problem overview

We now move onto the problem of embedding the three-dimensional piecewise flat manifold $(M, \mathcal{K}, l)$ in $\mathbb{R}^3$. In this case we wish to determine an embedding from the knowledge of the tetrahedron dihedral angles and the vertex positions of a single face in order to fix similarity transformations. This is analogous to the two-dimensional case since in three dimensions the geometric dual of the 2-skeleton of a single tetrahedron is also a tetrahedron (see figure 3.3). Therefore we can think of dihedral angles as being dual to edge lengths in the three-dimensional problem. Posing the problem in terms of dihedral angles also has the advantage that it may be extended to the problem of embedding an isotropic finite element mesh in $\mathbb{R}^3$, since the edge conductances are given in terms of edge lengths and dihedral angles. In the following section we give an overview of a related problem that has been well studied.

3.4.2 Distance geometry

The problem of finding the coordinates of a set of points given some distances between pairs of points is known in the literature as the distance geometry problem (DGP). This problem has received much attention particularly in the fields of molecular conformation [69], [52], [33] and sensor networks [97].

For a review of the theory and history of the DGP as well as an overview of many methods for solving the problem particularly when applied to molecular conformation, the reader is referred to [72], and references therein. We also give an overview of the problem to highlight its similarities and differences with our problem. The theory of distance geometry dates back to 1841, when Cayley [28] introduced the now well-known Calyey-Menger determinant for k points as

$$D(v_1,\ldots,v_k)=\begin{vmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & 0 & l_{12}^2 & \cdots & l_{1k}^2 \\ 1 & l_{12}^2 & 0 & \cdots & l_{2k}^2 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & l_{1k}^2 & l_{2k}^2 & \cdots & 0 \end{vmatrix}, \tag{3.4}$$

along with the volume formula for an n-simplex formed by $n+1$ points in Euclidean space:

$$V(v_1,\ldots,v_{n+1})=\frac{1}{n!}\sqrt{\frac{(-1)^{n+1}}{2^n}D(v_1,\ldots,v_{n+1})}. \tag{3.5}$$

Many related geometrical concepts relating set congruences and distances were introduced by Menger in the German paper of 1928 [78] and in 1931 [79], along with a theorem stating that matrices of the form (3.4) are in fact matrices of squared distances if certain conditions are satisfied, (see [42] for an account of these historic results).

The distance geometry problem is defined in [72] as follows: Given an integer $N>0$ and an undirected graph $G=(V,E)$ whose edges are weighted by their Euclidean lengths $l_{ij}=l(\{i,j\})$, determine whether there exists a function $x:V\to\mathbb{R}^N$ such that $\|x(i)-x(j)\|_2=l_{ij}$. Simply put, if we have a graph on which we know the Euclidean lengths of edges, does there exist an embedding of the vertices that is consistent with these lengths?

Th the DGP is clearly similar to our problem in that we also want to find an embedding of V given some information about G. The theory behind the DGP is very general and the methods that have been developed to solve the problem allow arbitrary graphs to be considered or at most only some a priori knowledge of the general form of the graph. Our problem differs from the DGP in that the graph for which the lengths arise forms a simplicial complex. Since we know that the graph is always of this type it seems prudent to use this information in the problem formulation. Due to the reasons

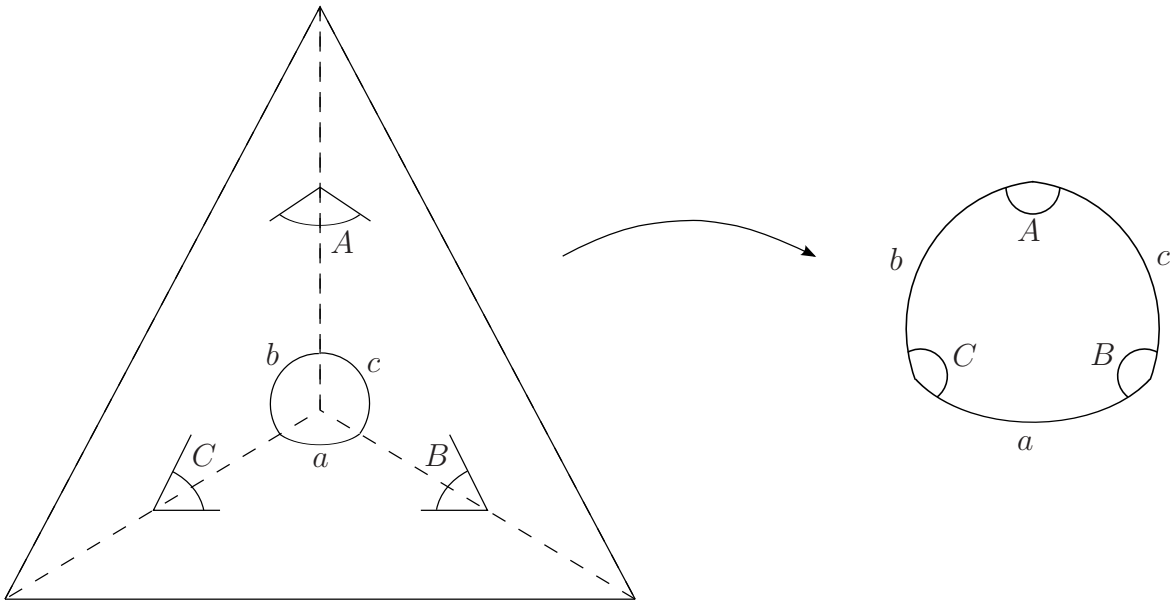


Figure 3.2: Angles at a vertex of a tetrahedron transformed to a spherical triangle. The face angles a, b, c map to geodesic distances on the unit sphere. The dihedral angles A, B, C map to the spherical angles between these geodesics.

stated in section 3.4.1 and the fact that geometric constraints can be directly applied to the dihedral angles, we formulate the problem in terms of such angles rather than distances between points, which would be the edge lengths in our case. The following section describes the constraints that the dihedral angles must satisfy in order for the embedding of the simplicial complex to be geometrically consistent.

3.4.3 Spherical trigonometry

Another geometrical concept that will be useful for our work in this chapter is spherical trigonometry, which is the study of angles and related lengths on the surface of the unit sphere. In our problem it will be necessary to convert dihedral angles to face angles and vice versa. Spherical trigonometry allows us to do this by relating the face angles of the three faces that meet at a vertex of a tetrahedron to the dihedral angles on the three edges that meet at the same vertex. We can form a spherical triangle using the face angles as the (curved) edges of the spherical triangle and the dihedral angle of the edge opposite a given face becomes the angle opposite the corresponding edge. Figure 3.2 illustrates this transformation from a tetrahedron to a spherical triangle for a single vertex. There are many identities for a spherical triangle that are analogous to those for a Euclidean triangle. Since we only require a conversion from dihedral

angles to face angles we quote the "spherical law of cosines for sides" [51], which is given by

$$\cos a = \cos b \cos c + \sin b \sin c \cos A, \tag{3.6}$$

and the "spherical law of cosines for angles", which is given by

$$\cos A = -\cos B \cos C + \sin B \sin C \cos a. \tag{3.7}$$

So to write the dihedral angle A in terms of the face angles a, b and c we rearrange (3.6) to give

$$\cos A = \frac{\cos a - \cos b \cos c}{\sin b \sin c}. \tag{3.8}$$

To write the face angle a in terms of the dihedral angles that meet at this vertex we rearrange (3.7) to give

$$\cos a = \frac{\cos A + \cos B \cos C}{\sin B \sin C}. \tag{3.9}$$

Clearly (3.8) can be rewritten for dihedral angles B and C, and (3.9) can be written for face angles b and c.

3.5 Consistency conditions

To ensure that our embedding is consistent with the definition of a simplicial complex and to ensure that we can embed it in $\mathbb{R}^3$, we need to derive some constraints on the dihedral angles. We show this in the following subsections then check that we do in fact have the correct number of independent constraints to compute the desired variables, the vertex positions.

3.5.1 Spherical Cayley-Menger consistency condition

Before we consider representing the manifold by a three-dimensional simplicial complex, we consider the construction of a single tetrahedron. In order to fix similarity transformations (three translation variables, three rotations and one scaling), we are required to fix seven vertex coordinates. Since a single tetrahedron has four vertices, each with 3 coordinates, we require

$$3 \times 4 - (3 + 3 + 1) = 5$$

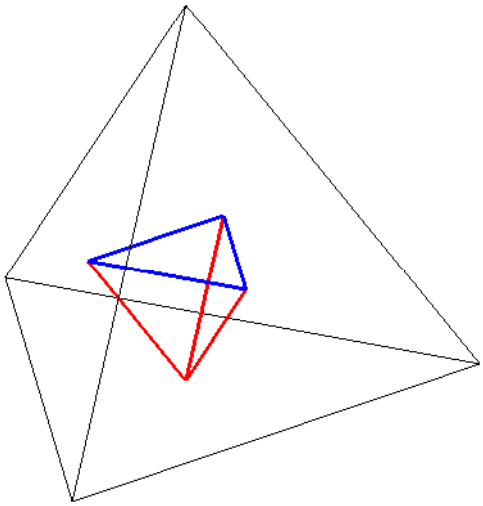


Figure 3.3: 2-skeleton of a single tetrahedron and graph defined by the vertices and edges of its dual complex (blue) with overlaid minimal spanning tree (red).

independent coordinate variables. We therefore have five independent dihedral angles and since there are six edges in a tetrahedron, we have one degree of freedom and so we require one constraint to uniquely embed a single tetrahedron in $\mathbb{R}^3$.

We define the following "spherical Cayley-Menger" matrix for the tetrahedron $\{i, j, k, l\}$,

$$M_{ijkl} = \begin{pmatrix} -1 & \cos\theta_{ij,kl} & \cos\theta_{ik,jl} & \cos\theta_{il,jk} \\ \cos\theta_{ij,kl} & -1 & \cos\theta_{jk,il} & \cos\theta_{jl,ik} \\ \cos\theta_{ik,jl} & \cos\theta_{jk,il} & -1 & \cos\theta_{kl,ij} \\ \cos\theta_{il,jk} & \cos\theta_{jl,ik} & \cos\theta_{kl,ij} & -1 \end{pmatrix}. \tag{3.10}$$

In [14], the following condition on the dihedral angles of a single tetrahedron is proved:

$$\det M_{ijkl} = 0. \tag{3.11}$$

Hence, given five independent dihedral angles of a tetrahedron, the sixth is constrained by (3.11). If we wished to embed a three-dimensional simplicial complex in $\mathbb{R}^N$ for some large N the Cayley-Menger constraint would still hold. If, for example, we began with a tetrahedron in $\mathbb{R}^3$ and we add a dimension every time we add a new vertex, since the Cayley-Menger condition is a constraint on each individual tetrahedron which can be embedded in $\mathbb{R}^3$, the condition still holds.

Above we noted that in [5] it was shown that in order to embed a triangulated surface in $\mathbb{R}^N$ for some N, the angles at the triangle vertices must satisfy a set of sine rule conditions for each cycle in the graph whose vertices and edges are derived from

the dual complex of the triangulated surface. Since the faces of a tetrahedron form a triangulation of the surface of a sphere in $\mathbb{R}^3$, the face angles must satisfy these sine conditions. Let $G'_F = (F_C, E_F)$ be the graph with vertex and edge sets given by the vertices and edges of the 2-skeleton dual complex. The vertices of G'_F are the face circumcentres F_C, and E_F are the edges connecting these vertices. From section 2.5, the dimension of the cycle space Z for a graph is

$$\dim Z = n_e - n_v + n_c, \tag{3.12}$$

where n_c is number of connected components, n_e is the number of edges and n_v is the number of vertices. This can also be computed as the number of edges in the graph that do not belong to a spanning tree of the graph. Then as shown in figure 3.3, the dimension of the cycle space of G'_F is $6-4+1=3$ and so there are three independent sine constraints for the face angles. As shown above, there are 5 independent face angles, with 12 face angles in total, therefore there are 7 dependent face angles for a single tetrahedron. Since we have found 3 constraints from the sine rule condition, we require $7-3=4$ additional constraints. These additional constraints are the obvious constraints on triangular face angles, that the three face angles sum to π. Since there are 4 faces in a tetrahedron, we have the 4 extra constraints. Since four points that satisfy the Cayley-Menger condition in (3.11) form a tetrahedron, this single constraint for the dihedral angles incorporates the seven consistency conditions for the face angles of a single tetrahedron.

Proposition 3.5.1. *The spherical Cayley-Menger condition in* (3.11) *is the three-dimensional analogue to the π constraint in the planar case.*

Proof. In the planar case the spherical Cayley-Menger matrix is given by

$$M_{ijk} = \begin{pmatrix} -1 & \cos\alpha_{i,jk} & \cos\alpha_{j,ik} \\ \cos\alpha_{i,jk} & -1 & \cos\alpha_{k,ij} \\ \cos\alpha_{j,ik} & \cos\alpha_{k,ij} & -1 \end{pmatrix}. \tag{3.13}$$

For a single triangle, let the angles be given by α, β and γ. Then

$$\det M = \cos^2\alpha + \cos^2\beta + \cos^2\gamma + 2\cos\alpha\cos\beta\cos\gamma - 1.$$

For a triangle, we have the condition $\alpha+\beta+\gamma=\pi$, so $\gamma = \pi-\alpha-\beta$. Then

$$\begin{aligned}
\det M &= \cos^2\alpha + \cos^2\beta - 2\cos\alpha\cos\beta\cos(\alpha+\beta) + \cos^2(\alpha+\beta) - 1, \\
&= \cos^2\alpha + \cos^2\beta - 2\cos^2\alpha\cos^2\beta + \\
&\qquad 2\cos\alpha\cos\beta\sin\alpha\sin\beta + \cos^2(\alpha+\beta) - 1, \\
&= \cos^2\alpha\sin^2\alpha + \cos^2\beta\sin^2\beta + \\
&\qquad 2\cos\alpha\cos\beta\sin\alpha\sin\beta + \cos^2(\alpha+\beta) - 1, \\
&= (\cos\alpha\sin\beta + \sin\alpha\cos\beta)^2 + \cos^2(\alpha+\beta) - 1, \\
&= \sin^2(\alpha+\beta) + \cos^2(\alpha+\beta) - 1, \\
&= 0.
\end{aligned}$$

So the sum to π condition for angles on a triangle results in the corresponding spherical Cayley-Menger determinant to be zero also. □

3.5.2 Curvature consistency condition

We now move onto global conditions that we require when we fit multiple tetrahedra together. As in the two-dimensional case we require that the manifold that we embed to be flat. In [47] two types of curvature are defined for three-dimensional manifolds and are as follows.

Definition 3.5.1. For a three-dimensional piecewise flat manifold, the *edge curvature* R_{ij} is

$$R_{ij} = \left(2\pi - \sum_{\{k,l\}\sim\{i,j\}} \theta_{ij,kl}\right) l_{ij}, \tag{3.14}$$

where $\{k,l\} \sim \{i,j\}$ denotes that edge $\{k,l\}$ shares a tetrahedron with edge $\{i,j\}$ so the sum is over all tetrahedra containing edge $\{i,j\}$.

Definition 3.5.2. The *scalar curvature* of a three-dimensional piecewise flat manifold is defined at vertices as

$$R_i = \sum_j R_{ij} d_i{}^j, \tag{3.15}$$

where R_{ij} is the edge curvature and $d_i{}^j$ is the pre-metric that satisfies $l_{ij} = d_i{}^j + d_j{}^i$.

The edge curvature is the discrete analogue to Ricci curvature defined in 2.3.53 and the scalar curvature is the discrete analogue to the Ricci scalar curvature defined

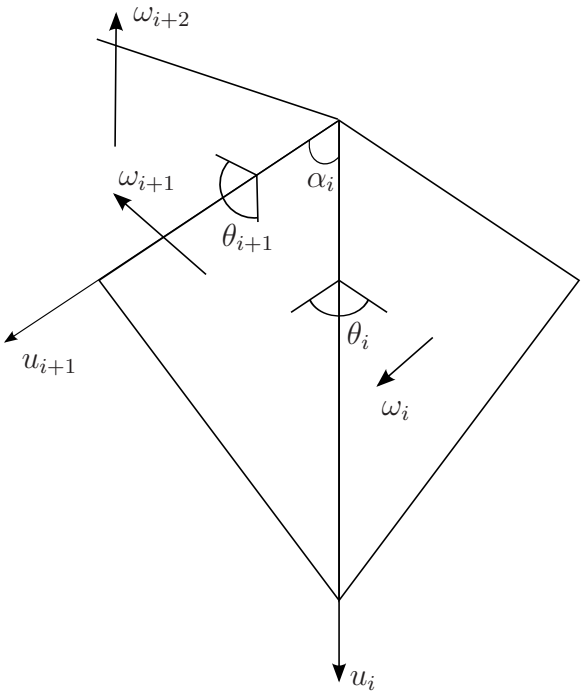


Figure 3.4: Vectors and angles required to rotate inward normal of face F_i to normal of F_{i+1}.

in 2.3.54. Again, since we wish to embed the manifold represented by the simplicial complex $\mathcal{K}$ in Euclidean space, we require zero curvature in the interior of the manifold. Since $R_{ij} = 0$ for $j \sim i$ implies $R_i = 0$, we can use the edge curvature condition. Since $l_{ij} > 0$ for all edges $\{i, j\} \in E$ the curvature consistency condition is

$$\sum_{\{k,l\} \sim \{i,j\}} \theta_{ij,kl} = 2\pi, \tag{3.16}$$

for all interior edges $\{i, j\} \in E_I$. Due to the relationship between scalar and edge curvature, it seems intuitive that some of the edge curvature conditions may depend on each other. Let us now define the graph $G' = (T, F_I)$ whose vertices and edges are given by the dual complex of $\mathcal{K}$. So the vertices of G' are represented by the 3-simplices of $\mathcal{K}$ and the edges of G' are dual to the interior 2-simplices of $\mathcal{K}$. From this point onwards when we refer to the dual graph, we mean G', not the actual dual of the graph $G = (V, E)$.

Lemma 3.5.1. *The edge curvature constraints are defined on the cycle space of the dual complex. Furthermore, the number of curvature constraints is* $\beta_0 - \chi + n_{E_I} - n_{V_I}$, *where* β_0 *is the zero'th Betti number and* χ *is the Euler characteristic of the domain.*

Proof. Consider a chain $c \in \mathcal{C}_1(G')$. Each edge connecting two nodes in this chain is

dual to a face in G. We also consider the vector-valued cochain $\omega \in \mathcal{C}^0(G')$ which for each tetrahedron T_i in the chain is given by the inward normal (with respect to T_i) of the face F_i dual to the edge we have traversed in the chain to reach T_i. Then using the following rotation formula [54], the inward normal ω_{i+1} is given by

$$\omega_{i+1} = \omega_i \cos\theta_i + (u_i \times \omega_i)\sin\theta_i + (1 - \cos\theta_i)(u_i \cdot \omega_i)u_i, \tag{3.17}$$

where u_i is the unit vector along the edge shared by F_i and F_{i+1} which is the axis of rotation, and θ_i is the dihedral angle on this edge in T_i. The vectors and angles used in this expression are illustrated in figure 3.4. By definition, ω_i is normal to any of the edges on F_i so the last term is zero. Since the axis of the next rotation is also the vector along an edge of the current tetrahedron we can write u_{i+1} in terms of u_i, ω_{i+1} and the face angle of the face shared by the two axes of rotation at their shared vertex a_i as

$$u_{i+1} = u_i \cos\alpha_i + (\omega_{i+1} \times u_i)\sin\alpha_i, \tag{3.18}$$

then inserting ω_{i+1} from (3.17) gives

$$u_{i+1} = u_i \cos\alpha_i + ((\omega_i \cos\theta_i + (u_i \times \omega_i)\sin\theta_i) \times u_i)\sin\alpha_i. \tag{3.19}$$

Using the vector triple product identity we have

$$(u_i \times \omega_i) \times u_i = \omega_i(u_i \cdot u_i) - u_i(u_i \cdot \omega_i). \tag{3.20}$$

But again the last term is zero since u_i is orthogonal to ω_i. Inserting (3.20) into (3.19) gives

$$u_{i+1} = u_i \cos\alpha_i + (\omega_i \times u_i)\cos\theta_i \sin\alpha_i + \omega_i \sin\theta_i \sin\alpha_i. \tag{3.21}$$

Let $m_i = u_i \times \omega_i$ so that

$$\omega_{i+1} = \omega_i \cos\theta_i + m_i \sin\theta_i, \tag{3.22}$$

and

$$\begin{aligned} m_{i+1} &= u_{i+1} \times (\omega_i \cos\theta_i + (u_i \times \omega_i)\sin\theta_i), \\ &= (u_i \cos\alpha_i + (\omega_i \times u_i)\cos\theta_i \sin\alpha_i + \omega_i \sin\theta_i \sin\alpha_i) \times \\ &\qquad (\omega_i \cos\theta_i + (u_i \times \omega_i)\sin\theta_i), \\ &= (u_i \times \omega_i)\cos\alpha_i \cos\theta_i - \omega_i \cos\alpha_i \sin\theta_i + \omega_i \times (u_i \times \omega_i)\sin\alpha_i, \end{aligned}$$

$$
\begin{aligned}
&= ((u_i \times \omega_i)\cos\theta_i - \omega_i \sin\theta_i)\cos\alpha_i + u_i \sin\alpha_i, \\
&= (m_i \cos\theta_i - \omega_i \sin\theta_i)\cos\alpha_i + u_i \sin\alpha_i. \qquad (3.23)
\end{aligned}
$$

If the same edge is used as the axis of rotation for two successive rotations then $u_i = u_{i+1}$ and $\alpha_i = 0$ so that (3.23) reduces to

$$m_{i+1} = m_i \cos\theta_i - \omega_i \sin\theta_i. \qquad (3.24)$$

Recall definition 2.4.10, a p-cycle c is a p-chain such that $\partial_p c = 0$, therefore

$$[\omega, \partial c] = 0, \qquad (3.25)$$

where $[\cdot,\cdot]$ is the dual pairing between chains and cochains. Then applying Stokes' theorem (theorem 2.3.1), (3.25) gives

$$[\omega, \partial c] = [\mathrm{d}\omega, c] = 0. \qquad (3.26)$$

where d is the coboundary operator which is defined for the dual edge e as

$$(\mathrm{d}\omega)_e = \omega_{i+1} - \omega_i. \qquad (3.27)$$

So combining (3.26) and (3.27), we have

$$\sum_{i=1}^{n} \omega_{i+1} - \omega_i = 0,$$

which reduces to $\omega_{n+1} = \omega_1$.

Now consider a cycle $c \in \mathcal{C}_1(G')$ of length $n+1$ defined on edges in E' that form the boundary of a face in G', that is, the edges on which c is defined are dual to faces in the primal graph G that all share a common edge in the primal graph and this edge is enclosed by the edges of the cycle. Then combining (3.22) and (3.24) we have the following expressions for each rotation:

$$
\begin{aligned}
\omega_2 &= \omega_1 \cos\theta_1 + m_1 \sin\theta_1, \\
m_2 &= m_1 \cos\theta_1 - \omega_1 \sin\theta_1, \\
\omega_3 &= \omega_2 \cos\theta_2 + m_2 \sin\theta_2, \\
&= (\omega_1 \cos\theta_1 + m_1 \sin\theta_1)\cos\theta_2 + (m_1 \cos\theta_1 - \omega_1 \sin\theta_1)\sin\theta_2, \\
&= \omega_1 \cos(\theta_1 + \theta_2) + m_1 \sin(\theta_1 + \theta_2),
\end{aligned}
$$

$$
\begin{aligned}
m_3 &= m_2 \cos\theta_2 - \omega_2 \sin\theta_2, \\
&= (m_1 \cos\theta_1 - \omega_1 \sin\theta_1)\cos\theta_2 - (\omega_1 \cos\theta_1 + m_1 \sin\theta_1)\sin\theta_2, \\
&= m_1 \cos(\theta_1 + \theta_2) - \omega_1 \sin(\theta_1 + \theta_2), \\
\omega_4 &= \omega_3 \cos\theta_3 + m_3 \sin\theta_3, \\
&= (\omega_1 \cos(\theta_1 + \theta_2) + m_1 \sin(\theta_1 + \theta_2))\cos\theta_3 + \\
&\qquad (m_1 \cos(\theta_1 + \theta_2) - \omega_1 \sin(\theta_1 + \theta_2))\sin\theta_2, \\
&= \omega_1 \cos(\theta_1 + \theta_2 + \theta_3) + m_1 \sin(\theta_1 + \theta_2 + \theta_3), \\
m_4 &= m_3 \cos\theta_3 - \omega_3 \sin\theta_3, \\
&= (m_1 \cos(\theta_1 + \theta_2) - \omega_1 \sin(\theta_1 + \theta_2))\cos\theta_3 - \\
&\qquad (\omega_1 \cos(\theta_1 + \theta_2) + m_1 \sin(\theta_1 + \theta_2))\sin\theta_2, \\
&= m_1 \cos(\theta_1 + \theta_2 + \theta_3) - \omega_1 \sin(\theta_1 + \theta_2 + \theta_3), \\
&\vdots \\
\omega_{n+1} &= \omega_n \cos\theta_n + m_n \sin\theta_n, \\
&= \left(\omega_1 \cos\left(\sum_{i=1}^{n-1}\theta_i\right) + m_1 \sin\left(\sum_{i=1}^{n-1}\theta_i\right)\right)\cos\theta_n + \\
&\qquad \left(m_1 \cos\left(\sum_{i=1}^{n-1}\theta_i\right) - \omega_1 \sin\left(\sum_{i=1}^{n-1}\theta_i\right)\right)\sin\theta_n, \\
&= \omega_1 \cos\left(\sum_{i=1}^{n}\theta_i\right) + m_1 \sin\left(\sum_{i=1}^{n}\theta_i\right). \qquad (3.28)
\end{aligned}
$$

Recall from above that for cycles we have $\omega_{n+1} = \omega_1$. Then (3.28) gives

$$\omega_1 = \omega_1 \cos\left(\sum_{i=1}^{n}\theta_i\right) + m_1 \sin\left(\sum_{i=1}^{n}\theta_i\right) \qquad (3.29)$$

which holds if and only if $\sum_{i=1}^{n}\theta_i = p\pi$ for $p \in 2\mathbb{Z}$. But $p = 0$ would result in degenerate tetrahedra and $p > 2$ results in intersecting tetrahedra, hence $p = 2$. This is exactly the edge curvature condition from definition 3.16, hence the edge curvature condition is defined for cycles in the dual graph.

In order to compute the number of independent cycles we will need to use some concepts from algebraic topology. Since the dimension of the cycle space for a graph is equal to the dimension of the 1st homology group, the number of independent curvature constraints is given by the first Betti number (see section 2.5). If we consider

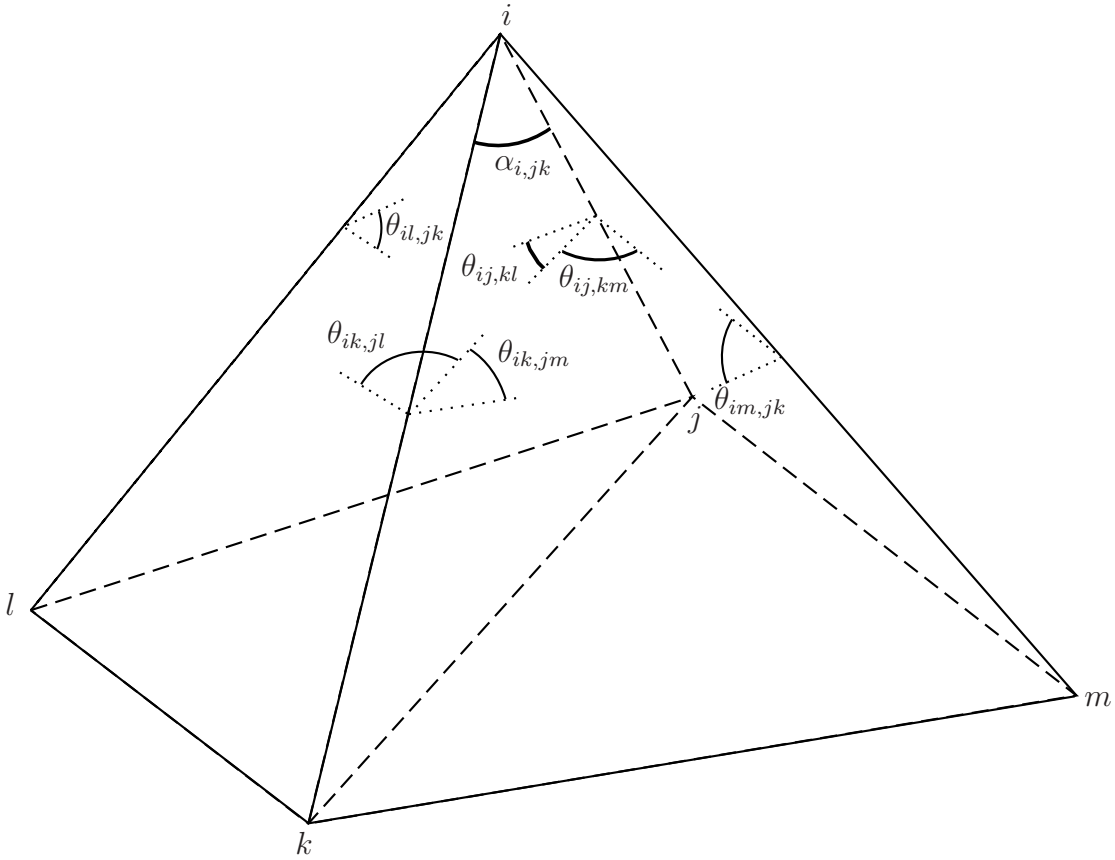


Figure 3.5: Dihedral angles required to compute a face angle for two neighbouring tetrahedra.

the graph G' as being made of only the vertices and edges, then from the Euler-Poincaré formula we have

$$\chi(G') = \chi(X) - n_2 + n_3,$$

where n_2 is the number of 2-cells we removed to form G' and n_3 is the number of 3-cells removed. Since each 2-cell of G' is dual to an interior edge of G, we have $n_2 = n_{E_I}$ and since each 3-cell is dual to an interior vertex in G, we have $n_3 = n_{V_I}$. Here n_{E_I} and n_{V_I} denote the number of interior edges and vertices of G respectively. Since the alternating sum of Betti numbers is equal to the Euler characteristic and $\beta_p = 0$ for $p > 3$, the number of independent curvature constraints is given by

$$\beta_1 = \beta_0 - \chi + n_{E_I} - n_{V_I}, \tag{3.30}$$

as required. □

3.5.3 Face angle constraints

The next set of conditions must be satisfied when two tetrahedra share a face. In order for two adjacent tetrahedra to fit together consistently, the shared face must be formed consistently in both tetrahedra. As shown in section 3.4.3, we can write a face angle at a given vertex in terms of the dihedral angles of the three edges of a tetrahedra that meet at the vertex using the spherical law of cosines for sides. Rewriting (3.8) in our standard notation for the angles of tetrahedron T_{ijkl}, at vertex i we have

$$\alpha_{i,jk} = \frac{\cos\theta_{il,jk} + \cos\theta_{ij,kl}\cos\theta_{ik,jl}}{\sin\theta_{ij,kl}\sin\theta_{ik,jl}}. \tag{3.31}$$

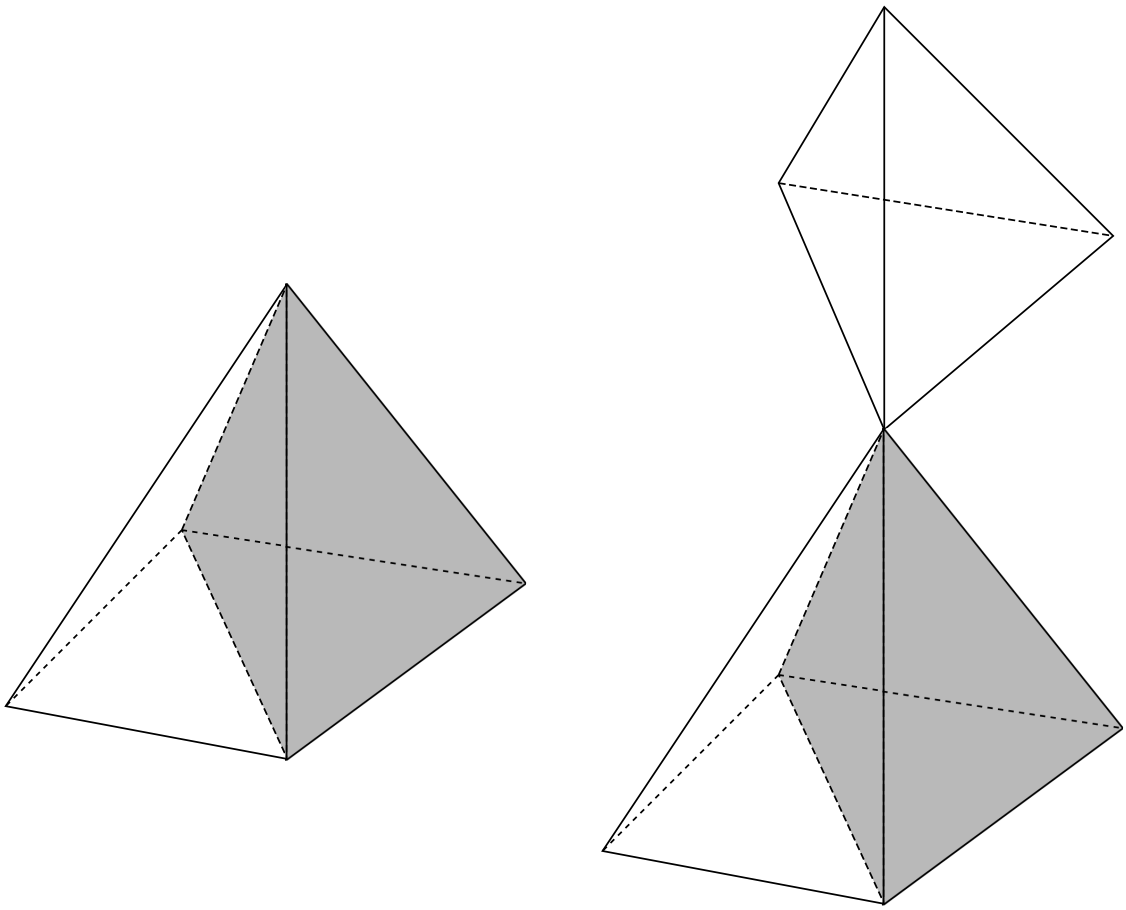
Then for the tetrahedra to fit together consistently the face angles for the shared face must be equal at each vertex. Then for the two tetrahedra T_{ijkl} and T_{ijkm} sharing face F_{ijk}, at vertex i we have the condition

$$\frac{\cos\theta_{il,jk} + \cos\theta_{ij,kl}\cos\theta_{ik,jl}}{\sin\theta_{ij,kl}\sin\theta_{ik,jl}} = \frac{\cos\theta_{im,jk} + \cos\theta_{ij,km}\cos\theta_{ik,jm}}{\sin\theta_{ij,km}\sin\theta_{ik,jm}}. \tag{3.32}$$

The angles used in this condition are illustrated in figure 3.5. This constraint is somewhat analogous to the sine rule constraint in the planar case. In the planar case, the sine rule constraint ensures that edge lengths between adjacent triangles match, whereas here we ensure that the face angles match which in turn forces edge lengths to match since we can determine edge lengths from angles using the sine rule and the initial scaling introduced by fixing the seven vertex coordinates. For each interior face of the mesh (i.e. a face shared by two tetrahedra) there are three of these conditions, but due to the π rule for face angles only two are independent (see section 3.3).

3.6 Uniqueness of three-dimensional problem

In this section we lay out a constructive procedure for finding the independent constraints and dependent dihedral angles for a contractible domain. We show that this procedure produces the correct number of equations for calculating the unknown vertex positions. We then use the implicit function theorem to show that the consistency conditions defined in the previous section uniquely determine the $3n_V - 7$ unknown vertex coordinates.



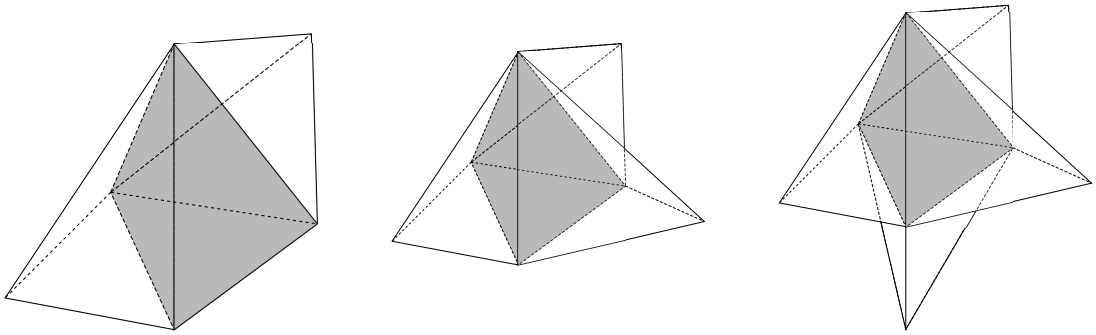
(a) (b)

Figure 3.6: Case 1 of constructive procedure. The current tetrahedron is the shaded tetrahedron.

3.6.1 Constructive Procedure

We now give a constructive procedure for finding the independent set of constraints that uniquely determines an embedding of a three-dimensional simplicical complex in $\mathbb{R}^3$. We restrict ourselves to the case of a contractible domain. In order to identify the angles that are independent of the constraints and those that depend on the constraints, we traverse a spanning tree of the dual mesh.

Lemma 3.6.1. *Consider a three-dimensional triangulation $\mathcal{K}$ of a three-dimensional contractible domain. Let $G' = (V', E') = (T, F_I)$ be the dual graph defined in section 3.5.2. Let n_X be the cardinality of X, where X is replaced by the relevant set of k-simplices. For example, n_V is the number of vertices. Then there are $n_T + (n_{E_I} - n_{V_I}) + (2n_{F_I} - 3n_{V_I})$ independent constraints on the dependent dihedral angles of the embedded simplicial complex $\mathcal{K}$.*



(a) Case 2 (b) Case 3 (c) Case 4

Figure 3.7: Cases 2, 3 and 4 of constructive procedure. The current tetrahedron is the shaded tetrahedron.

Proof. Let S' be a spanning tree of G'. We begin the construction at the root of this tree and construct a single tetrahedron. As shown in section 3.5.1, we have five independent dihedral angles and one degree of freedom which is constrained by the spherical Cayley-Menger constraint. We now traverse the edges of S'. Whenever we reach a new node (tetrahedron) in the tree, one of the following holds:

Case 1: Only one dual edge has been traversed between the current node and previously visited nodes. There are two possibilities in this case:

(a) Only the tetrahedra whose vertices are shared by the face dual to this edge have been visited so we know only one face of the current tetrahedron;

(b) Some other tetrahedron which does not share a face with the current tetrahedron but that shares the fourth vertex has been visited;

Case 2: Two adjacent tetrahedra have been visited so we know two faces and all vertices;

Case 3: Three adjacent tetrahedra have been visited therefore we know three faces and four vertices;

Case 4: Four adjacent tetrahedra have been visited therefore we know all four faces.

Considering each case we have the following constructive procedure for computing the constrained dihedral angles.

Case 1 (a): If we have only previously visited the previous node in this branch of the tree there is one vertex left to find, therefore we have three variables that

can be chosen arbitrarily. Let the dihedral angles at the edges of the known face be the independent variables in this tetrahedron then the three dihedral angles to be determined are those on the edges that meet at the fourth vertex. As described above, we have two independent constraints by ensuring that the face angles of the known face match when computed for the new tetrahedron and the previous one. We also have the Cayley-Menger constraint for this tetrahedron which is clearly independent of these face angle constraints and any previous constraints that may have been defined.

Case 1 (b): Since all vertices are known, all the dihedral angles of this tetrahedron are to be determined by constraints. The constraints which need to be satisfied involving the angles of this tetrahedron cannot yet be determined so we end the search down this branch of the tree and return later in which case this tetrahedron will fall into one of the following cases.

Case 2: If we have previously visited two tetrahedra that share faces with this tetrahedron then we have no vertices left to find and so all the angles in this tetrahedron are constrained. In this case there must be an edge in G' that has been removed to create S' so by temporarily reinserting this edge creates a cycle in G'. Hence, there is a curvature constraint around the interior edge of the primal graph G that is enclosed by reinserting this dual edge. We have two faces known for which we match two angles each with the previously visited adjacent tetrahedra. The final condition is again the Cayley-Menger condition for this tetrahedron. Therefore we have 6 new conditions for 6 new unknowns. Clearly, these three types of constraints are independent since they are defined by different geometric properties.

Case 3: In this case we must have three interior edges since we cannot have a tree that has already visited three adjacent tetrahedra otherwise. We must also enclose an interior vertex in this case which is the vertex at which the three known faces meet. As given by the first Betti number of G', for every interior vertex we lose one independent curvature constraint so we have two curvature constraints for this tetrahedron which are found by using the approach used in case 2 of temporarily reinserting the missing dual edges. We now have three faces each with two angles to match but we only require three more constraints. Considering the vertex at which the edges whose curvature constraints are independent meet, there is only one dihedral angle left to find so we use the face angle of the face opposite this edge to compute its dihedral

angle. Then using the face angles of the other two known faces at the vertices that are not on the edges that have a curvature constraint allows us to compute the fourth and fifth unknown angles. Then the Cayley-Menger constraint gives six independent constraints. Hence, whenever we enclose an interior vertex, we lose three independent face angle constraints.

Case 4: In this case all the edges of this tetrahedron are interior edges. We remove a curvature constraint associated with an edge adjacent to each vertex such that the edges associated with the two remaining independent constraints share a common vertex. For the faces sharing these edges in this tetrahedron, the process for finding the independent face angle constraints is the same as in case 3. For the other interior vertices of this tetrahedron, we find another tetrahedron that shares one of the curvature constraint edges, apply the curvature constraint to this edge and repeat the process for removing face angles until three face angle constraints have been removed for each of the interior vertices of this tetrahedron, ensuring that we only remove three face angle constraints for an interior vertex in a single tetrahedron.

In each case we have the Cayley-Menger constraint so we have n_T of these constraints. Every case involved matching 2 face angles of previously visited faces except for the last tetrahedron to enclose an interior vertex, and since this involves matching every interior face in the mesh, we have $2n_{F_I} - 3n_{V_I}$ independent face angle matching conditions. We also have the $n_{E_I} - n_{V_I}$ independent curvature constraints so adding together we have the specified total number of independent constraints. □

3.6.2 Number of equations

In this section we will check that the number of equations from the consistency conditions matches the number of constraints for a contractible domain. We will also see how a change in topology affects the required number of constraints.

Lemma 3.6.2. *For a contractible domain, the number of independent dihedral angles is equal to the number of degrees of freedom in the survey problem, that is, the number of unknown vertex coordinates. For domains with topology such that $\beta_0 - \chi \neq 0$, we require 6 extra constraints.*

Proof. We have the following relations that we will use to check the number of equations. Since $V_I \cup V_B = V$ and $V_I \cap V_B = \emptyset$, and similarly for E and F, we have the following relations

$$n_V = n_{V_I} + n_{V_B}, \tag{3.33}$$

$$n_E = n_{E_I} + n_{E_B}, \tag{3.34}$$

$$n_F = n_{F_I} + n_{F_B}. \tag{3.35}$$

Since $\mathcal{K}$ is a discretisation of a three-dimensional volume, the Euler characteristic for $\mathcal{K}$ is

$$\chi_{\mathcal{K}} = n_V - n_E + n_F - n_T = \chi. \tag{3.36}$$

Likewise, the Euler characteristic for the dual complex $\mathcal{K}^*$ is

$$\chi_{\mathcal{K}^*} = n_T - n_{F_I} + n_{E_I} - n_{V_I} = \chi. \tag{3.37}$$

Let $\mathcal{K}_B$ be the simplicial complex formed by the 2-simplices representing the surface triangulation. Then since this is the surface of a three-dimensional domain, the Euler characteristic for the boundary triangulation is

$$\chi_{\mathcal{K}_B} = n_{V_B} - n_{E_B} + n_{F_B} = 2\chi, \tag{3.38}$$

where the second equality is a result that can be found in [49]. Each tetrahedron is formed by 4 faces and each interior face is shared by 2 tetrahedra, whilst each boundary face belongs to only 1 tetrahedron, hence

$$4n_T = 2n_{F_I} + n_{F_B}. \tag{3.39}$$

If we consider the boundary of M, each boundary face has 3 boundary edges and 2 boundary faces meet at each boundary edge, so

$$3n_{F_B} = 2n_{E_B}. \tag{3.40}$$

Let n_θ and n_{eq} be the number of dihedral angles and constraints respectively, then

$$\begin{aligned} n_\theta - n_{\text{eq}} &= 6n_T - (n_T + (\beta_0 - \chi + n_{E_I} - n_{V_I}) + (2n_{F_I} - 3n_{V_I}))\,, \\ &= 3n_V - \beta_0 - 3n_{V_B} + n_{F_B} + n_T + \chi - n_{E_I} + n_{V_I}, \qquad \text{(by (3.33) and (3.39))}, \\ &= 3n_V - \beta_0 - 6\chi + n_V - 2n_E + n_F + n_{F_B} + n_{V_I}, \qquad \text{(by (3.38) and (3.36))}, \end{aligned}$$

$$
\begin{aligned}
&= 3n_V - \beta_0 - 5\chi + n_T + n_V - n_{E_I} - n_{F_B}, && \text{(by (3.40) and (3.37))},\\
&= 3n_V - \beta_0 - 6\chi + n_{E_B} - n_{F_I} - 2n_{F_B} + 2n_T, && \text{(by (3.36) and (3.35))},\\
&= 3n_V - 7\beta_0 + 6(\beta_0 - \chi), && \text{(by (3.39) and (3.40))}.
\end{aligned}
$$

□

Therefore we have the correct number of constraints to ensure that we can find the vertex positions up to similarity for domains such that $\beta_0 = \chi$. Then for each connected component, by fixing two vertex positions and one coordinate of another vertex of the same face it is possible to find the relative vertex positions of all vertices of the embedding.

For domains with $\beta_0 > \chi$ we require $6(\beta_0 - \chi)$ extra equations compared to those given in the constructive proof of lemma 3.6.1. An example of such a domain is a solid torus.

When $\beta_0 < \chi$, $6(\beta_0 - \chi)$ of the constraints are redundant. An example of such a domain is a solid sphere with a smaller spherical void at its centre.

3.6.3 Proof of Uniqueness

By applying the constructive procedure of section 3.6.1 we can find the set of independent and dependent dihedral angles required to define and satisfy the constraints respectively. Let $\tilde{\theta}$ and θ be the vectors of independent and dependent dihedral angles respectively. As shown in section 3.6.2 there are $n_i = 3n_V - 7$ independent angles so $\tilde{\theta} \in \mathbb{R}^{n_i}$ and we have $n_d = n_T + (n_{E_I} - n_{V_I}) + (2n_{F_I} - 3n_{V_I})$ independent equations and dependent angles, so $\theta \in \mathbb{R}^{n_d}$. Let

$$
\begin{aligned}
f : \mathbb{R}^{n_i} \times \mathbb{R}^{n_d} &\to \mathbb{R}^{n_d},\\
(\tilde{\theta}, \theta) &\mapsto c,
\end{aligned}
$$

be the vector valued function representing the value of the non-linear functions that form the constraints for given values of $\tilde{\theta}$ and θ. We note here that the three forms f can take are as follows:

$$
\begin{aligned}
f_{\mathrm{CM}} &= \det(M),\\
f_{\mathrm{FA}} &= \frac{\cos\theta_{il,jk} + \cos\theta_{ij,kl}\cos\theta_{ik,jl}}{\sin\theta_{ij,kl}\sin\theta_{ik,jl}} - \frac{\cos\theta_{im,jk} + \cos\theta_{ij,km}\cos\theta_{ik,jm}}{\sin\theta_{ij,km}\sin\theta_{ik,jm}},
\end{aligned}
$$

$$f_{\text{curv}} = 2\pi - \sum_{\{k,l\}\sim\{i,j\}} \theta_{ij,kl},$$

as outlined in section 3.5. Here f_{CM} are the Cayley-Menger determinants for each tetrahedron as defined in (3.11), f_{FA} are the values of the face angle matching conditions from (3.32) and f_{curv} are the curvature conditions from (3.16).

We now quote the well-known implicit function theorem (see for example [90]) since we will use it in the following.

Theorem 3.6.1. *Let $f : U \times V \to W$ be a smooth function where $U \subset \mathbb{R}^m$ and $V, W \subset \mathbb{R}^n$. Let*

$$(x, y) = (x_1, \ldots, x_m, y_1, \ldots, y_n).$$

Let $f(x_0, y_0) = w_0$ for a fixed point (x_0, y_0). If the sub-matrix $J_{f,y}(x_0, y_0) = \partial f/\partial y|_{(x_0,y_0)}$ of the Jacobian of f evaluated at (x_0, y_0) is invertible, then there exists an open set U_0 of U containing x_0, an open set W_0 of W containing $f(x_0, y_0)$ and a unique smooth function $g : U_0 \to V$ with $g(x) = y$ such that $f(x, g(x)) = 0$ for $x \in U_0$.

Lemma 3.6.3. *Let $J_{f,\theta}(\tilde{\theta}, \theta) = \partial f/\partial\theta|_{(\tilde{\theta},\theta)}$ and let $(\tilde{\theta}_0, \theta_0)$ be a set of angles that satisfy the constraints so that $f(\tilde{\theta}_0, \theta_0) = 0$, then the Jacobian $J_{f,\theta}(\tilde{\theta}_0, \theta_0)$ has full rank.*

Proof. From lemma 3.6.1 the constraints are independent, therefore it follows that the Jacobian has full rank. □

Lemma 3.6.4. *For the set of dihedral angles $(\tilde{\theta}_0, \theta_0)$ satisfying the consistency constraints, there exists a neighbourhood $U \subset \mathbb{R}^{n_i}$ of $\tilde{\theta}_0$ and a neighbourhood $V \subset \mathbb{R}^{n_d}$ of θ_0, a unique continuously differentiable function $g : U \to V$ with $g(\tilde{\theta}) = \theta$ such that $f\left(\tilde{\theta}, g(\tilde{\theta})\right) = 0$ for $\tilde{\theta} \in U$.*

Proof. For angles $(\tilde{\theta}_0, \theta_0)$ satisfying the constraints the Jacobian $J_{f,\theta}(\tilde{\theta}_0, \theta_0)$ has full rank by lemma 3.6.3, therefore is invertible. Then by the implicit function theorem (theorem 3.6.1) the statement holds. □

The above lemma states that if the Jacobian with respect to the dependent angles is invertible for a complete set of angles that satisfy the constraints, then we can perturb the independent angles by a small amount such that it is possible to find a unique set of dependent angles that satisfy the constraints.

Theorem 3.6.2. *Given a set of dihedral angles $(\tilde{\theta}_0, \theta_0)$ that satisfy the consistency conditions defined in section 3.5, the three coordinates of two vertices v_1, v_2 sharing an edge and one coordinate of a third vertex sharing a face with the v_1 and v_2 defines a unique embedding of the simplicial complex $\mathcal{K}$.*

Proof. Given a set of dihedral angles we can compute all the face angles using (3.31). We can also compute all the edge lengths for E using the sine rule where the length of edge $\{1, 2\}$ defines the scaling throughout. Let F_1 be the face shared by the 3 known vertices. To compute the the unknown coordinates of the third vertex we solve the following non-linear system

$$\frac{(v_3 - v_1)^T(v_2 - v_1)}{\|v_3 - v_1\|\|v_2 - v_1\|} = \cos\alpha_{1,23},$$
$$\frac{(v_1 - v_3)^T(v_2 - v_3)}{\|v_1 - v_3\|\|v_2 - v_3\|} = \cos\alpha_{3,12},$$

for the two unknown coordinates of v_3. Let S' be a spanning tree of G' (not necessarily the same tree as used to define the independent angles) and root the tree at a tetrahedron that has F_1 as a face. By traversing the spanning tree we visit each tetrahedron of $\mathcal{K}$ via a face whose vertex coordinates are known. Therefore we are left with the problem of finding the fourth vertex of the tetrahedron. Let T_{ijkl} be the current tetrahedron and assume that the known vertices are v_i, v_j, v_k. To compute the fourth vertex v_l we rotate the vector $l_{jl}(v_k - v_j)/l_{jk}$ by $\alpha_{j,kl}$ about $n_{ijk,l}$, then rotate again about $(v_j - v_k)/l_{jk}$ by $(\pi - \theta_{jk,il})$. Here $n_{ijk,l}$ is the outward normal of F_{ijk} with respect to T_{ijkl} given by

$$n_{ijk,l} = \frac{(v_k - v_i) \times (v_j - v_i)}{\|v_k - v_i\|\|v_j - v_i\|}.$$

Since we define $\alpha, \theta \in (0, \pi)$, the above expression for computing the fourth vertex is unique given unique values of v_i, v_j and v_k.

As we traverse S' we will visit some tetrahedra whose fourth vertex may have already been located. We must check that the vertex position computed for the current tetrahedron is the same as the position computed from any other tetrahedron that shares this vertex. The face angle matching constraints ensures that both face angles and edge lengths are equal between tetrahedra that share the edges and faces. The curvature constraints ensure that the intersection of two neighbouring tetrahedra is

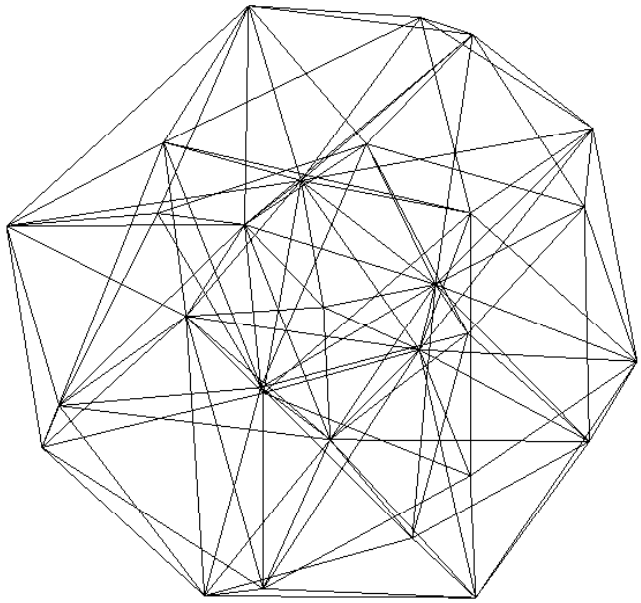
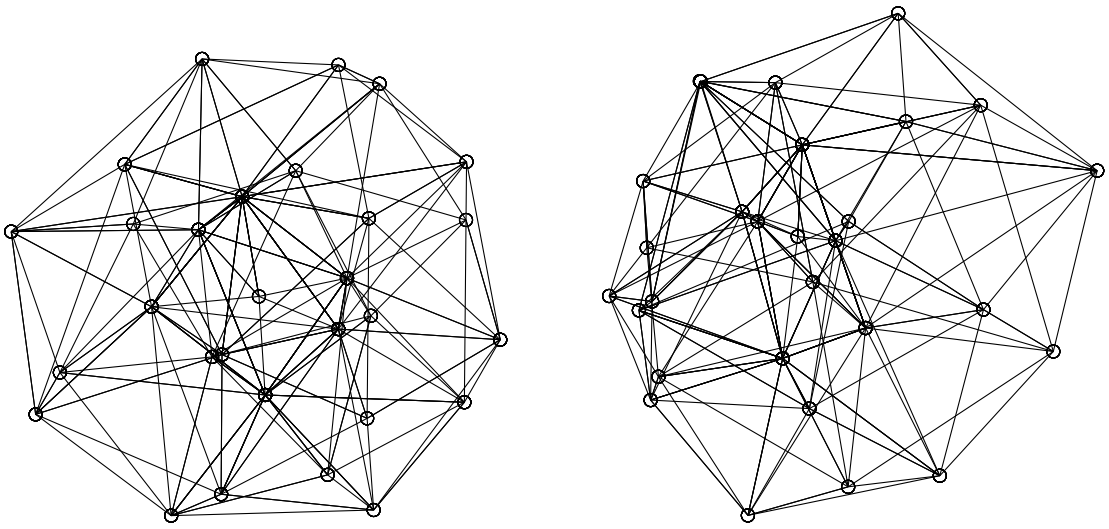


Figure 3.8: Example mesh.

exactly the face shared by both tetrahedra. The Cayley-Menger constraint ensures that the edges in a tetrahedron join up to form a non-degenerate tetrahedron. Therefore the constraints ensure that the angles and edge lengths used to compute the fourth vertex in the current tetrahedron are consistent with those used to compute the vertex in any previous tetrahedra that have been visited. Therefore, the position of the fourth vertex is unique in each tetrahedron, hence by induction on S' the embedding is unique. □

3.7 Numerical Example

In this section we will show an example of the triangulation problem in practice. We begin with a three-dimensional simplicial complex embedded in $\mathbb{R}^3$ and compute all the dihedral angles with knowledge of the vertex positions. We then traverse a spanning tree of the dual graph to find the independent equations and angles and perturb the angles by adding normally distributed random variables with a specified standard deviation. We then solve the resulting non-linear system using `fsolve` from MATLAB's Optimization Toolbox [77]. Once we have the dihedral angles we traverse a spanning tree of the dual graph and compute the vertex positions as described above. To compute the vertex positions we use an extension of the code used in [5] and the

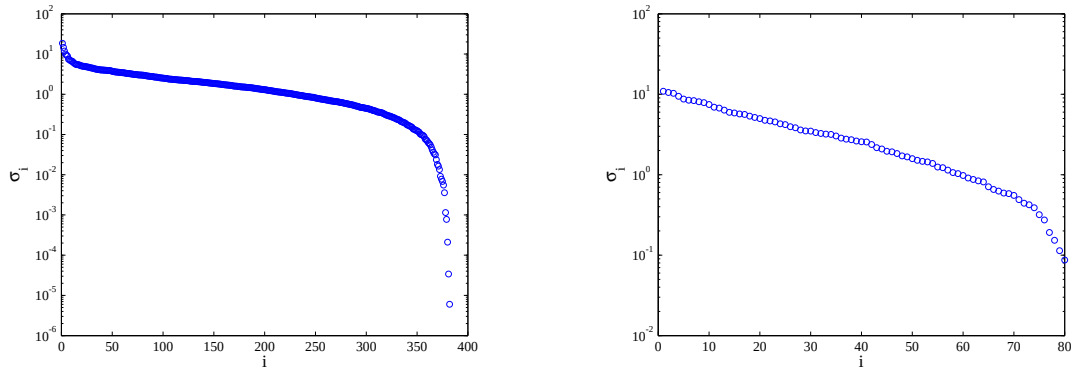


(a) $\epsilon = 1 \times 10^{-2}$. (b) $\epsilon = 8 \times 10^{-2}$.

Figure 3.9: Resulting embeddings for random perturbations of the independent angles with standard deviation ϵ.

graph theory toolbox of Sergey Iglin [57]. Figure 3.8 shows the mesh we will use as the example in this section. This mesh is a modification of a mesh produced by NETGEN [85]. The original mesh had one interior vertex but in order to illustrate the embedding problem with more degrees of freedom, six more interior vertices were added before remeshing using the `delaunay` function in MATLAB. The mesh has the following parameters: $n_V = 29$, $n_{V_I} = 7$, $n_E = 125$, $n_F = 174$ and $n_T = 77$.

Figure 3.9 shows the resulting meshes when the initial angles are perturbed by normally distributed random vectors with standard deviations $\epsilon = 1 \times 10^{-2}$ and $\epsilon = 8 \times 10^{-2}$. For perturbations larger than 8×10^{-2}, the optimisation routine did not converge. As illustrated by figure 3.9b, a perturbation of 8×10^{-2} applied to the independent angles results in large perturbations in some of the dependent angles. This example illustrates how it is possible to apply a small perturbation to the independent angles that originally satisfied the constraints, that produces a perturbation in the dependent angles such that the full set of angles satisfy the constraints described in section 3.5. This illustrates the practical application of lemma 3.6.4 and theorem 3.6.2. A plot of the singular values of $J_{f,\theta}$ defined in section 3.6 is given in figure 3.10a for the perturbed independent angles and resulting dependent angles that satisfy the constraints. This plot illustrates that the Jacobian has full rank up to a tolerance of 10^{-6}. The resulting value of the residual from `fsolve` was of the order of 10^{-12}



(a) Jacobian $J_f(\tilde{\theta}, \theta)$ of the mapping from independent to dependent dihedral angles.

(b) Jacobian $J_h(v)$ mapping the unknown vertex positions to all dihedral angles.

Figure 3.10: Plots of singular values σ_i, from largest to smallest, against the corresponding ranking i of each singular value, of the Jacobian for the two mappings f and h. The singular values are calculated for the dihedral angles that satisfy the constraints after applying a perturbation to the original independent angles with $\epsilon = 8 \times 10^{-2}$.

so using a tolerance much less than 10^{-6} to measure the rank of the Jacobian seems sensible. This is another illustration of the numerical results matching the theory given in section 3.6.

Although we prove theorem 3.6.2 directly using our constructive procedure, another possible way of proving this theorem would be to introduce a function from the unknown vertex coordinates to the independent dihedral angles and show that this function is invertible in a neighbourhood of the dihedral angles that satisfy the constraints. Let

$$h : \mathbb{R}^{n_i} \to \mathbb{R}^{n_i},$$
$$v \mapsto \tilde{\theta},$$

where n_i is the number of independent dihedral angles as before and v are the unknown vertex coordinates. Let $J_h(v)$ be the Jacobian of h for some set of vertex coordinates v. Here we state the well-known inverse function theorem (see [90] for example).

Theorem 3.7.1. *Let $f : U \to V$ be a smooth function between two open sets U, V, such that $U, V \subset \mathbb{R}^n$. Let $u_0 \in U$ and $v_0 \in V$ such that $f(u_0) = v_0$. Suppose the Jacobian $Df(u_0)$ is invertible, then there exists a neighbourhood U_0 of u_0 and a neighbourhood V_0 of v_0 such that $f : U_0 \to V_0$ is invertible and $f^{-1} : V_0 \to U_0$ is also smooth.*

Suppose $h(v_0) = \tilde{\theta}_0$, then if $J_h(v_0)$ is full-rank we can use the inverse function theorem to show that h is invertible in some neighbourhood of $\tilde{\theta}_0$. That is, if we start with a mesh with known vertex coordinates, we can perturb the independent angles by some amount such that the resulting angles produce a unique set of consistent vertex coordinates. Figure 3.10b shows a plot of the singular values of a finite difference approximation of J_h evaluated at the new vertex coordinates after perturbing the independent angles. For this plot it is clear that J_h is invertible for this set of vertex coordinates which again matches the theory outlined here. The MATLAB code that produced the results in this section can be found in Appendix A.

3.8 Conclusion

In this chapter we have extended the two-dimensional work of [5] to the three-dimensional case. We defined novel constraints on the dihedral angles of a tetrahedral mesh such that it can be embedded in $\mathbb{R}^3$ and proved that these constraints lead to a locally unique embedding of a three-dimensional simplicial complex. Without the work in this chapter it is not possible to tackle the problem in the next chapter which is to find an embedding of a simplicial complex with the added constraint that it is consistent with a finite element discretisation of the EIT problem.

Chapter 4

Discrete EIT problem in $\mathbb{R}^3$

4.1 Introduction

In this chapter we introduce the concept of resistor networks and relate this concept to the finite element (FE) system matrix introduced in section 2.2. In the resistor network formulation, the elements of the FE system matrix are shown to be equivalent to conductances of resistors placed on the edges of the mesh. We then extend the problem of embedding a 3-simplicial complex $\mathcal{K}$ in three-dimensional space to the case of having knowledge of only the edge conductances for each edge of the mesh. This assumes that we have solved the inverse problem on the mesh to obtain the edge conductances. We treat this problem because when solving the forward problem using finite elements, the conductivity only affects the data through the edge conductances. Therefore, when solving the inverse problem in practice, the best possible scenario is to accurately find the edge conductances and then hope to fit a conductivity distribution to the edge conductances. In this chapter we will investigate the uniqueness of this mapping between edge conductances and conductivities for some example meshes.

As described in section 2.1, it has been shown that the anisotropic electrical impedance tomography (EIT) problem does not have a unique solution. In the discrete problem, the diffeomorphism that causes the non-uniqueness could be a change in the position of interior vertices of the mesh with the change in conductivity given by equation (2.14). So the (possibly arbitrary) choice of finite element mesh would constrain the reconstructed conductivity distribution. Then the characterisation of the effect of mesh choice on the conductivity distribution becomes an important problem. In

this chapter we will use numerical experiments to investigate how this non-uniqueness affects the discrete problem and how we may overcome this problem by constraining the conductivity using a priori information about the form of the anisotropy.

The novelty of this work lies in the numerical experiments that suggest uniqueness and non-uniqueness for the discrete EIT problem given certain assumptions on the anisotropy. The MATLAB code that produced the results in this chapter can be found in appendix B.

4.2 Resistor networks

In this section we introduce some definitions and theory related to resistor networks before reviewing some results from past work related to resistor networks in the context of EIT.

4.2.1 Introduction

The theory introduced in this section is taken from [36]. Throughout this section let $G = (V, E)$ be a graph as defined in section 2.5. We also use the same notation for labelling boundary and interior components from chapter 3. In order to define a resistor network we need to introduce functions that will be defined on the graph.

Definition 4.2.1. A *conductivity* on G is a function that assigns a positive real number to each edge in G. So

$$\gamma : E \to \mathbb{R}^+,$$
$$e \mapsto \gamma_e,$$

and the number γ_e is called the *edge conductance* of the edge $e \in E$.

Definition 4.2.2. A *resistor network* $\Gamma = (G, \gamma)$ is a graph G and a conductivity function γ defined on G.

If we define a potential u on the vertices of G, as we have done in the FE formulation of the EIT problem, then the resistor network can be thought of as an electrical network whose nodes are the vertices of G. Furthermore, for edge $e = \{i, j\}$ the current c_e

through edge e is given by Ohm's Law:

$$c_e = \gamma_e (u_i - u_j). \tag{4.1}$$

We will consider only networks in which there is at most a single edge connecting two nodes so that c_e given in (4.1) is the total current from i to j. Kirchhoff's current law states that the current c_i entering the network at a node i is the sum of the currents from i to nodes adjacent to i, that is

$$c_i = \sum_{j \sim i} \gamma_{ij} (u_i - u_j). \tag{4.2}$$

The following definition is important for EIT in which we assume no internal current sources.

Definition 4.2.3. A potential u is *γ-harmonic* at node i if the total current from i to its adjacent nodes is zero, that is

$$c_i = 0. \tag{4.3}$$

If the above equation holds for all the interior nodes $i = 1, \ldots, n_{V_I}$ then u is a *γ-harmonic function.*

If we also impose the discrete version of the conservation of charge condition in (2.4) on a resistor network with boundary then a γ-harmonic function has zero current at every node.

Definition 4.2.4. The *Kirchhoff matrix* $K \in \mathbb{R}^{n_V \times n_V}$ is a symmetric matrix whose entries are given by

$$K_{ij} = \begin{cases} -\gamma_{ij}, & \text{for } i \sim j, \\ \sum_{i \sim k} \gamma_{ik}, & \text{for } i = j, \\ 0, & \text{for } i \nsim j. \end{cases}$$

The vector of currents at each node is given by the system $c = Ku$.

We can see here then that the Kirchhoff matrix is equivalent to the FE system matrix defined in section 2.2 for piecewise linear potential on well-centred simplices, where the edge conductances for the FE mesh are the integrals given in (2.26) for each pair of vertices sharing an edge. That is, the edge conductance of an edge $\{i, j\}$ is

the sum of contributions to this integral from each tetrahedron that has $\{i, j\}$ as an edge. So if we simulate voltages using FEM for EIT reconstruction, it is clear that the individual conductivities are not explicitly "seen" by the voltages, only their summed contributions to the edge conductances count towards the voltages. Thus the best we can hope for is that we can reconstruct the edge conductances from the boundary data and then try to recover the conductivities from the edge conductances. Later in this chapter we will explore the relationship between conductivities and edge conductances for various conductivity types.

4.2.2 Previous work related to EIT

In this section we review past work that has posed the EIT problem in terms of resistor networks. The problem of determining resistor values was considered in [70], in which the problem was posed as a discrete version of the inverse conductivity problem. It was shown that the conductivity of resistors in an integer lattice can be determined from boundary measurements provided the conductivities are small perturbations of a constant.

In [34] an inductive argument is used to prove global uniqueness for the conductivity values of a resistor network on an integer lattice. The argument involves finding the conductivities for two resistors in a corner of the network, then using this approach iteratively through the network. This also leads to a direct reconstruction method for the resistor conductivity values.

Uniqueness of the Dirichlet to Neumann map was proved in [35] based on a matrix representation of the map on a resistor network. This representation was also used to give a clearer representation of the reconstruction procedure given in [34].

Circular planar resistor networks are considered in [37]. It is shown that the edge conductances can be uniquely determined from the Dirichlet to Neumann map by making use of a specific type of graph transformation defined for planar graphs.

Circular planar resistor networks based on adaptive radial layered finite difference grids are considered in [20]. It is shown that the edge conductances can be uniquely recovered from boundary data for certain numbers of layers and the algorithm of [34] is used to recover the resistor values. An optimisation-based reconstruction that does not require a-priori information is then applied to recover the conductivity and a method

for incorporating a-priori information into the reconstruction is given. An extension of [20] to the partial data problem is given in [18]. The extension to partial data involves a coordinate transformation of the grids from the previous paper, which results in the grid being finer close to the parts of the boundary where the data is given. This work is extended further in [19] to the case of pyramidal finite difference grids, which are particularly useful in geophysical applications, where the domain is often modelled as a half-space with data only defined on the boundary of the half-space. Again, uniqueness of the edge conductances from boundary data is shown for an even number of layers in the grid. A layer-by-layer approach for finding the edge conductances is given, then the reconstructed conductivities are found using linear interpolation of edge conductances across cells.

Resistor networks derived from planar finite element models are considered in [66]. It is shown that there are unique edge conductances for a resistor network that produce a Dirichlet to Neumann map that matches the Dirichlet to Neumann map of a FE model for piecewise constant anisotropic conductivity. They show that the non-uniqueness in the continuum problem extends to the FE formulation for anisotropic conductivities, that is, perturbations in interior vertex positions that correspond to diffeomorphisms with the conductivity perturbed as in (2.14), do not change the Dirichlet to Neumann map. This is to be expected since the FEM is defined to converge to the continuous case as the mesh is refined. It is also shown that there exists another type of non-uniqueness in the discrete problem. This non-uniqueness presents itself in the form of adding a function to the weak form integral for a triangle and subtracting the same function in the adjacent triangle for interior edges. It is shown that if a change in conductivity is equivalent to this adding and subtracting of functions on interior edges then there is no change in the Dirichlet to Neumann map.

It seems that this non-uniqueness will also be present in the three-dimensional case, where an edge has multiple tetrahedra sharing it. In the three-dimensional case rather than adding and subtracting the same function to either side of the edge we would just require that the additional functions defined on tetrahedra around an edge sum to zero. We will give an example of this in the following section, in which we show that by perturbing the conductivity so that the contributions of the conductivity perturbation to the edge conductance sum to zero for each edge, the edge conductances do not

change, therefore the Dirichlet to Neumann map does not change.

Our work is a step towards extending the planar resistor network based work of [5] and [83] to the three-dimensional case. In [5], sufficient conditions were given for the existence and uniqueness of the planar embedding of a specific kind of two-dimensional layered simplicial complex with known edge conductances. To prove this, a labelling method is used that shows the Jacobian of the vertex position to angle and conductivity map has full rank. This result allows use of the inverse function theorem to prove that for a given embedded simplicial complex $\mathcal{K}_0$ with known edge conductances γ_0, there exists a set of edge conductances γ in some neighbourhood of γ_0 and a simplicial complex $\mathcal{K}$ with the same topology as $\mathcal{K}_0$ such that $\mathcal{K}$ can be embedded in the plane.

Paridis [83] considered the problem of embedding a two-dimensional simplicial complex in $\mathbb{R}^3$, such that the embedding is consistent with some known edge conductance values. The author uses a circumcircle representation of the embedded triangulation to give face angles in terms of edge conductances, then uses the work of Duffin [43] to prove that there is an open neighbourhood of edge conductances γ_0 for which a planar simplicial complex with edge conductances in this neighbourhood can be embedded in $\mathbb{R}^n$ for some n. Alexandrov's theorem is then used to prove existence and uniqueness of the embedding in $\mathbb{R}^3$ under certain curvature conditions. At first glance, the result proved by Paridis is not restricted to the planar case since it requires the dual graph of 2-simplices, however the result makes use of the fact that at most two faces meet at an edge and the edge conductances are functions of face angles belonging to single faces. Since there is no theoretical limit on the number of faces that meet at an edge in a three-dimensional simplicial complex we cannot directly extend this result to the three-dimensional case. The circumcircle representation of a two-dimensional simplicial complex [83] also results in a nonlinear resistor network problem defined on the graph based on the dual complex of the two-dimensional simplicial complex. This is possible due to the fact that the dual cells of 1-simplices are of the same dimension, that is they are also 1-cells. In three dimensions this is clearly not the case. It is possible that a better understanding of the discrete Hodge star, which incorporates duality, for a three-dimensional simplicial complex may lead to an extension of the result in [83] based on the Cayley-Menger determinant in each tetrahedron, rather than the sum to

π condition imposed in the two-dimensional case. In addition, removing the curvature constraints from the three-dimensional problem to introduce the edge conductances is not very practical since it would require embedding a three-dimensional manifold in $\mathbb{R}^4$.

4.3 Edge conductances in EIT

In the previous chapter we showed how a non-degenerate three-dimensional simplicial complex $\mathcal{K}$ can be embedded in $\mathbb{R}^3$ with knowledge of dihedral angles and the vertex positions of one face. We now attempt to find an embedding of a finite element discretisation of the EIT problem. As shown in section 2.2 and highlighted in section 4.2, for isotropic, piecewise constant conductivity, the discrete EIT problem produces a system matrix K that is equivalent to edge conductances γ on a resistor network. Note that in this section we do not pay attention to the given boundary conditions (or in fact the potential u at all) since we will only be interested in the problem of finding a triangulation of $\mathbb{R}^3$ and conductivity distribution that fit a known vector-valued function γ.

In this section, we will consider how various conductivity types are mapped to edge conductances for a given geometry and topology. Let

$$\mathcal{Q}_{X,\mathcal{K}} : \sigma \mapsto \gamma$$

be the linear map taking piecewise constant anisotropic conductivities to edge conductances for a given set of vertex positions X and triangulation given by the simplicial complex $\mathcal{K}$. Since $\mathcal{Q}_{X,\mathcal{K}}$ is a discrete linear mapping we can represent it by the matrix $Q \in \mathbb{R}^{n_E \times k n_T}$ where k is the number of unknown conductivity components per tetrahedron. For example, for the isotropic case $k = 1$, and for the anisotropic case $k = 6$, since the conductivity is represented by a 3×3 symmetric matrix in each tetrahedron. As we stated in the introduction to this chapter, we assume we have solved the inverse problem to get the edge conductances, therefore if we wish to find the conductivity distribution for a given mesh we are required to invert the mapping $\mathcal{Q}_{X,\mathcal{K}}$. Then if Q has a non-trivial null-space the mapping $\mathcal{Q}_{X,\mathcal{K}}$ is not injective and so we cannot find a unique piecewise constant conductivity that produces the given edge conductances

for the given mesh. That is, if rank $Q < kn_T$, we cannot expect to recover unique conductivity values. This is clearly a huge obstacle to overcome when solving the EIT problem, since for an arbitrary mesh our reconstructed conductivity distribution may be one of infinitely many possible solutions to the discrete EIT problem.

We will now give expressions for Q for various types of conductivity distributions in terms of vertex positions. It is well known, see for example [21], that for isotropic conductivities the edge conductances can be computed using the following cotangent formula

$$\gamma_{ij} = \frac{1}{6} \sum_{\{k,l\}} \sigma_{ijkl} l_{kl} \cot \theta_{kl,ij}, \tag{4.4}$$

where l_{kl} is the length of the edge opposite edge $\{i, j\}$ in the given tetrahedron and the sum is over tetrahedra for which $\{i, j\}$ is an edge. This is possible because the edge conductances are invariant to rotations and translations in the isotropic case. For general anisotropic conductivities the edge conductances are not invariant to rotations so knowledge of the angle between face normals and preferred directions of conductivity are required. Clearly, this information is not available if the conductivity tensor is unknown. This is the main reason for not parameterising the embedding by the dihedral angles as we did in the previous chapter.

Let Q_{mn} be the mapping from the nth conductivity variable (of the kn_T in total) to the mth edge. Let $\{i, j\}$ be the mth edge, then let the tetrahedron to which the nth conductivity variable belongs be $t = \{i, j, k, l\}$. Let x_i be the position of the ith vertex in Euclidean space as usual. Let

$$c_i^{(t)} = (x_k - x_l) \times (x_j - x_l), \tag{4.5a}$$

$$c_j^{(t)} = (x_i - x_l) \times (x_k - x_l), \tag{4.5b}$$

then the gradient of linear basis functions on tetrahedron t are given by the vectors

$$\nabla \varphi_i^{(t)} = \frac{c_i^{(t)}}{6|V_t|}, \tag{4.6a}$$

$$\nabla \varphi_j^{(t)} = \frac{c_j^{(t)}}{6|V_t|}, \tag{4.6b}$$

where $|V_t|$ is the volume of tetrahedron t as before. Let

$$s = (s_{11}, \ldots, s_{1k}, s_{21}, \ldots, s_{2k}, \ldots, s_{n_T k})^T \in \mathbb{R}^{kn_T}$$

be the vector of stacked conductivity variables.

4.3.1 Isotropic conductivity

For isotropic conductivity, $\sigma = \beta I$ in each tetrahedron for some scalar β, so $s \in \mathbb{R}^{n_T}$. Then

$$Q_{mn} = -|V_n| \nabla \varphi_i \cdot \nabla \varphi_j, \tag{4.7}$$

$$= -\frac{c_i^T c_j}{36|V_n|}, \tag{4.8}$$

where the index m indicates edge $m = \{i, j\}$, and we have dropped the superscript (t) since it is clear that we only consider functions on the tetrahedron to which the nth conductivity variable belongs.

4.3.2 Anisotropic conductivity

For a general anisotropic conductivity, the conductivity is of the form

$$\sigma = \begin{pmatrix} \sigma^{11} & \sigma^{12} & \sigma^{13} \\ \sigma^{12} & \sigma^{22} & \sigma^{23} \\ \sigma^{13} & \sigma^{23} & \sigma^{33} \end{pmatrix}, \tag{4.9}$$

for each tetrahedron. So there are 6 conductivity variables per tetrahedron and $s \in \mathbb{R}^{6n_T}$. Due to the anisotropy we require a different expression for Q for each component of the conductivity. We have

$$Q_{m,6(t-1)+1} = -c_i^1 c_j^1/(36|V_t|), \tag{4.10a}$$

$$Q_{m,6(t-1)+2} = -(c_i^1 c_j^2 + c_i^2 c_j^1)/(36|V_t|), \tag{4.10b}$$

$$Q_{m,6(t-1)+3} = -(c_i^1 c_j^3 + c_i^3 c_j^1)/(36|V_t|), \tag{4.10c}$$

$$Q_{m,6(t-1)+4} = -c_i^2 c_j^2/(36|V_t|), \tag{4.10d}$$

$$Q_{m,6(t-1)+5} = -(c_i^2 c_j^3 + c_i^3 c_j^2)/(36|V_t|), \tag{4.10e}$$

$$Q_{m,6(t-1)+6} = -c_i^3 c_j^3/(36|V_t|), \tag{4.10f}$$

where c_i^p is the pth coordinate of the vector c_i.

4.3.3 Anisotropy constrained by a priori information

In this section we assume that we have some information about the anisotropy, whether that be that we know some components of it, we know it in specific locations or we

have some information about the type of anisotropy present. In section 2.1.3 we gave references to uniqueness results for the continuous problem when partial knowledge of the conductivity is given.

Uniaxial anisotropy with known direction

Here we consider the case of an anisotropic conductivity distribution that has one preferred direction, which is known and is orthogonal to a surface. Practically, this could occur in geophysical applications when the subsurface has a layered structure that is finer than the resolution of our reconstruction and we know the directions of the layers from some other modality, such as seismic imaging.

Mathematically, this means that the conductivity tensor has one unique eigenvalue corresponding to the eigenvector orthogonal to the layer surfaces and a repeated eigenvalue that corresponds to the two eigenvectors tangential to the layers, w_1 and w_2. The eigenvalues and eigenvectors are dependent on the position, which allows for layers that vary in direction throughout the domain. The tangential eigenvectors are unique up to rotations in the plane tangential to the layer surfaces with the constraint that they remain orthogonal to each other. This technicality will not matter because we can decompose the conductivity so that it does not rely on the tangential eigenvectors. Let λ_n and λ_w be the eigenvalues corresponding to the normal and tangential eigenvectors respectively. Then since σ is a real symmetric matrix it has an eigenvalue decomposition

$$\sigma = U\Lambda U^T,$$

where U is orthogonal and has the eigenvectors of σ as columns. Then

$$\begin{aligned}
\sigma &= \begin{pmatrix} n & w_1 & w_2 \end{pmatrix} \operatorname{diag}(\lambda_n, \lambda_w, \lambda_w) \begin{pmatrix} n & w_1 & w_2 \end{pmatrix}^T, \\
&= \lambda_n n n^T + \lambda_w \left(w_1 w_1^T + w_2 w_2^T \right), \\
&= (\lambda_n - \lambda_w)\, n n^T + \lambda_w \left(n n^T + w_1 w_1^T + w_2 w_2^T \right), \\
&= (\lambda_n - \lambda_w)\, n n^T + \lambda_w U U^T, \\
&= (\lambda_n - \lambda_w)\, n n^T + \lambda_w I.
\end{aligned}$$

The reconstruction problem is then to compute the unknown eigenvalues given knowledge of the eigenvector n. So let $s = (\lambda_{1n}, \lambda_{1w}, \lambda_{2n}, \lambda_{2w}, \ldots, \lambda_{n_T n}, \lambda_{n_T w})^T \in \mathbb{R}^{2n_T}$,

then

$$\nabla\varphi_i \cdot (\sigma\nabla\varphi_j) = \lambda_n (n \cdot \nabla\varphi_i)(n \cdot \nabla\varphi_j) + \lambda_w (\nabla\varphi_i \cdot \nabla\varphi_j - (n \cdot \nabla\varphi_i)(n \cdot \nabla\varphi_j)).$$

Which gives the following expressions for the linear map from conductivity variables to edge conductances

$$Q_{m,2(t-1)+1} = -\frac{\left(n^T c_i\right)\left(n^T c_j\right)}{36|V_t|}, \tag{4.11a}$$

$$Q_{m,2(t-1)+2} = \frac{1}{36|V_t|}\left(\left(n^T c_i\right)\left(n^T c_j\right) - c_i^T c_j\right). \tag{4.11b}$$

Conformally related anisotropy known up to scaling factor

Here we consider the case of anisotropic conductivity that is known up to a scalar factor. The continuous version of this problem was considered in [73], as described in section 2.1.3. Here we let $\sigma = \alpha A$ where A is a known matrix-valued function and α is a scalar function to be found. The form of the matrix Q is similar to that of the isotropic case except the matrix A acts as a metric in the inner product of the gradients of the FE basis functions. That is

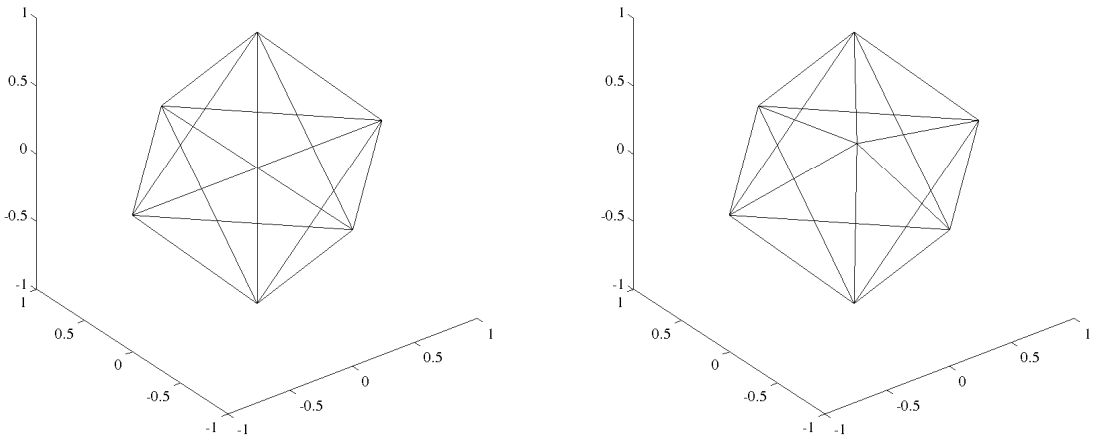
$$Q_{mn} = -\frac{c_i^T A c_j}{36|V_n|}, \tag{4.12}$$

and $s = (\alpha_1, \ldots, \alpha_{n_T})^T \in \mathbb{R}^{n_T}$.

4.3.4 Example of a non-injective $\mathcal{Q}_{X,\mathcal{K}}$

In this section we use a simple example to illustrate how a poor choice of mesh can result in a non-trivial null space of the matrix Q which causes non-uniqueness of the conductivity distribution. This is the three-dimensional analogy of the second type of non-uniqueness of the discrete problem described in [66], (see section 4.2.2 for a summary of the main results from this paper).

The example meshes we will use for the examples in this section are shown in figure 4.1. We will refer to the mesh in figures 4.1a and 4.1b as the original and perturbed meshes respectively. We label the tetrahedra in an anti-clockwise ordering starting with the tetrahedron whose centroid has positive (x, y, z) coordinates. The mesh has 6 boundary vertices, 1 interior vertex and 8 tetrahedral elements. It has 18 edges so we have $Q \in \mathbb{R}^{18\times 8k}$, where again k is the number of unknown conductivity components



(a) Original mesh.

(b) Original mesh with interior vertex perturbed by the vector $(0.1, 0.1, 0.1)^T$.

Figure 4.1: Mesh used for the example in section 4.3.4. The interior vertex is at the origin in (a) and is perturbed in (b).

per tetrahedron. We can observe that the tetrahedra in the original mesh are identical to each other, therefore for homogeneous conductivity the conductances of edges in the same position in each tetrahedron relative to the origin will be equal.

The figures that we will discuss can be found at the end of this section. Figure 4.2a shows the singular values of Q for the original mesh. We can see that for the original mesh, Q has four singular values that are below the threshold of what we can consider non-zero when using double precision floating point arithmetic, therefore Q has a non-trivial null space. A rational basis for the null space given by reducing Q to reduced row echelon form is shown in 4.2b. With the ordering of tetrahedra outlined above, we can see from the null vectors that if we add equal and opposite amounts to the conductivity of tetrahedra as we go around an edge, then we will produce no change in the edge conductances. Since the vectors shown here span the null space, we can use any linear combination of the vectors to perturb the conductivity and produce no change in the edge conductance values. For example, the first basis vector tells us that perturbations of conductivity values of the form $\delta s = \alpha\,(-1, 1, -1, 1, 0, 0, 0, 0)$, for some $\alpha \in \mathbb{R}$, such that the conductivity remains positive in each tetrahedron, results in no change in edge conductances.

Figure 4.3 shows the singular values of Q for the perturbed mesh. All the singular

values are non-zero, therefore Q has full rank for the perturbed mesh. This illustrates that by breaking the symmetry in the mesh by perturbing the interior vertex we can remove the non-uniqueness related to the discretisation.

Figure 4.4a shows the singular values of Q for anisotropic conductivity on the original mesh. If we didn't know the dimensions of Q, then this plot could lead us to think that Q has full rank. In fact, since $Q \in \mathbb{R}^{18\times 48}$, the maximum possible rank of Q is 18, therefore we will always have a null space of dimension at least 30. The vectors in figure 4.4b provide a basis for $\operatorname{null} Q$. Again, a perturbation in conductivity formed by any linear combination of these vectors will produce no change in the edge conductances. As an example, the first vector defines a perturbation in conductivity that is zero in every component except for σ^{22} in the first and second tetrahedra, where these two components of the perturbation are equal but with opposite sign. Figure 4.5 shows a similar situation for the perturbed mesh with the null space having the same dimension as for the original mesh. We note that $\operatorname{null} Q$ is slightly different for this mesh due to the perturbation applied to the interior vertex. So the non-uniqueness for anisotropic conductivity in the discrete problem appears due to the fact that there is not enough information in the edge conductances to uniquely determine the six components of the conductivity tensor in each tetrahedron. This non-uniqueness is observable before we even consider a diffeomorphism of the mesh that would result in the discrete analogy of Tartar's non-uniqueness result in (2.14).

In an attempt to reduce the number of unknown conductivity components, we could utilise some a priori information to constrain the anisotropy. We require $kn_T < n_E$. If we know in advance that the anisotropy is caused by fine layers, as described in section 4.3.3, we can formulate the problem so that we only have two unknown conductivity components for each tetrahedron. This will at least give us a chance of Q having full rank in this particular example. [83] provides the number of edges and tetrahedra for some example meshes. For many typical meshes it is shown that $2n_T \approx n_E$, with $2n_T > n_E$ in some cases and $2n_T < n_E$ in others. For the cases where the number of edges is less, we would need to constrain the conductivity further in some tetrahedra. In many physical applications this is typically done anyway. For example, in geophysical applications, borehole sample data is often used to provide prior information, so for elements close to the boreholes it would seem prudent to fix

the conductivity here since we can be reasonably sure of the conductivity distribution in these locations. Going back to our example, we can see from figure 4.6 that for the original mesh and the anisotropy defined by flat layers perpendicular to the z-axis, Q has non-trivial null space. Figure 4.6b shows a basis for $\operatorname{null} Q$. Figure 4.7 shows how perturbing the interior vertex increases the rank but Q still has a null space in this case. Figures 4.8 and 4.9 show the results for the original and perturbed meshes respectively in the case of layers perpendicular to the radial direction. In both cases we observe that Q has a null space. Figures 4.10 and 4.11 show the case of layered anisotropy where the direction perpendicular to the layers is a Gaussian random perturbation of $(0, 0, 1)^T$ in each tetrahedron. We note that the same random vector was applied for both the original and perturbed meshes to ensure no bias was introduced. Comparing figures 4.10a and 4.6a we can see that perturbing the eigendirections of the layers increases the rank in the original mesh, although $\operatorname{null} Q$ is still non-trivial. Comparing figures 4.11 and 4.7 we see that adding the random perturbation to the eigendirections increases the rank again. The combination of perturbing the interior vertex to remove symmetries in the mesh and perturbing the eigendirections results in Q being full-rank. We also note that this procedure was tried for a number of different random perturbations and the results were the same in each case.

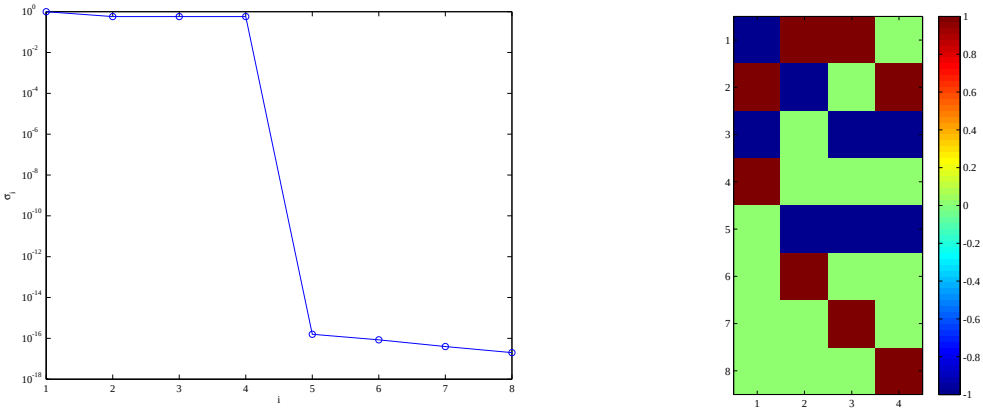
We next consider the case of conductivities conformally related to a known tensor field. The conductivity is of the form $\sigma = \alpha A$, for some unknown scalar function α and known matrix-valued function A. Figures 4.12, 4.13 and 4.14 show the singular values of Q for the original and perturbed meshes where the known matrix is constant across tetrahedra. The matrices used in the figures are

$$A = \begin{pmatrix} 1 & 0.1 & 0 \\ 0.1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0.1 & 0.1 \\ 0.1 & 1 & 0 \\ 0.1 & 0 & 1 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0.1 & 0.1 \\ 0.1 & 1 & 0.1 \\ 0.1 & 0.1 & 1 \end{pmatrix}.$$

In each figure we observe that for the perturbed mesh, in each case Q has full rank. This should be expected after seeing the results for isotropic conductivity since isotropic conductivities are just a special case of the conformal class with the matrix A fixed as the identity everywhere. We see that as we make the conductivity more anisotropic, the rank of Q increases. Figure 4.15 shows that if the known matrix-valued function is a random anisotropic perturbation of the identity in each tetrahedron, then even

for the original mesh, Q has full rank. In a sense we have broken the symmetry by defining the conductivity to have different preferred directions at different points in the domain.

Although the example in this section is somewhat contrived, it illustrates that without even considering the non-uniqueness of anisotropic conductivities due to diffeomorphisms, which will be considered in the next section, the solution to the discrete EIT problem can still be non-unique if we are not careful about the mesh that is used for solving the forward problem. It also illustrates how constraining the conductivity to be of a particular class can remove the non-uniqueness in some cases.



(a) Singular values.

(b) Rational basis for null Q.

Figure 4.2: Singular values of Q for the original mesh and vectors forming the rational basis for null Q with isotropic conductivity.

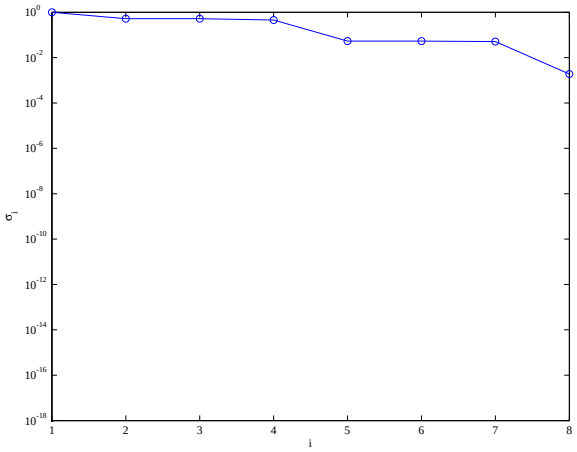
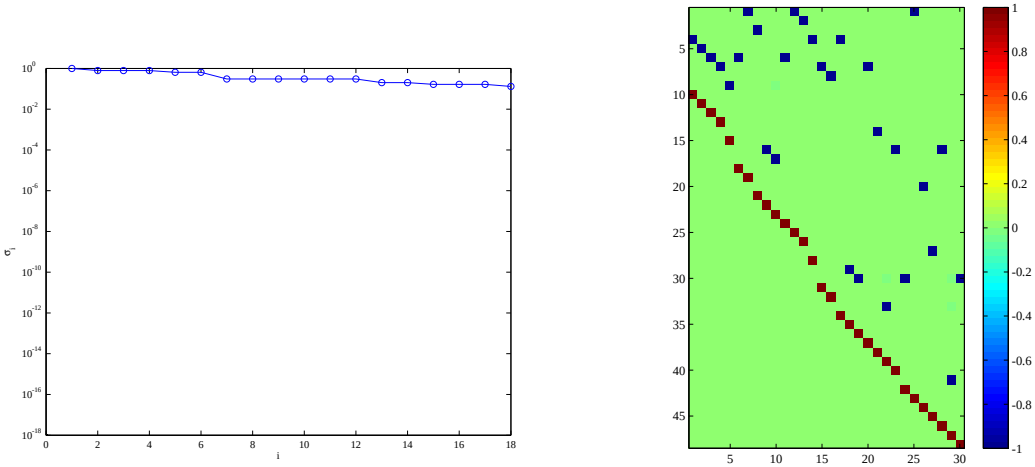
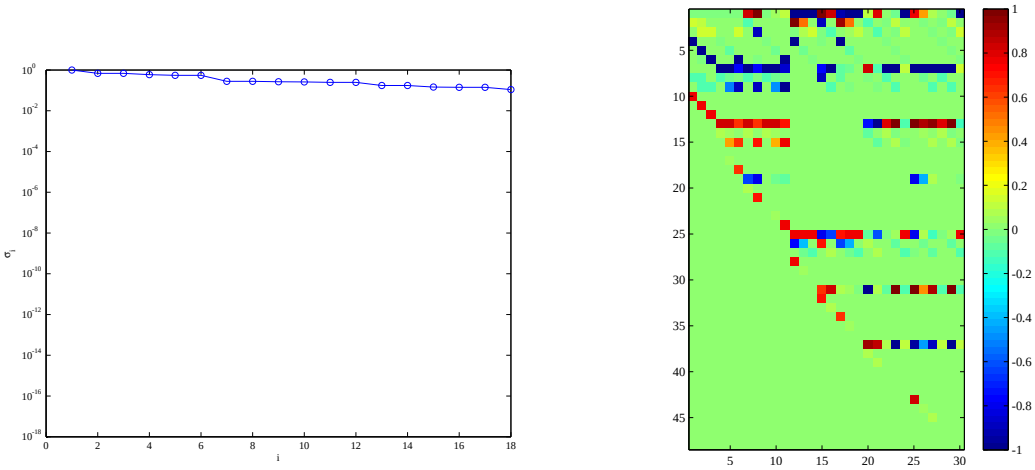


Figure 4.3: Singular values of Q for the perturbed mesh with isotropic conductivity.



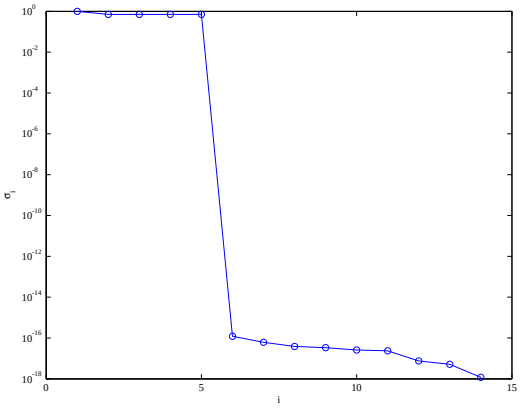
(a) Singular values. (b) Rational basis for null Q.

Figure 4.4: Singular values of Q for the original mesh and vectors forming the rational basis for null Q with anisotropic conductivity.

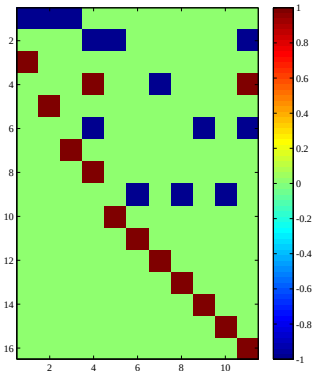


(a) Singular values. (b) Rational basis for null Q.

Figure 4.5: Singular values of Q for the perturbed mesh and vectors forming the rational basis for null Q with anisotropic conductivity.

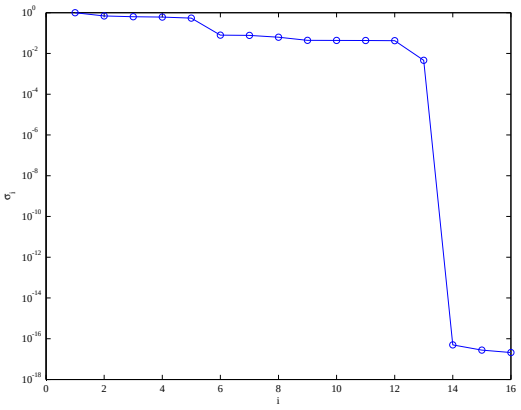


(a) Singular values.

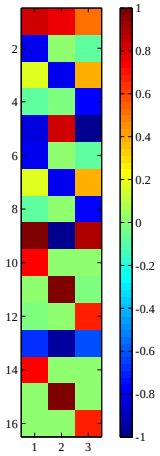


(b) Rational basis for null Q.

Figure 4.6: Singular values of Q for the original mesh and vectors forming the rational basis for null Q. The conductivity here was constrained to be formed by a layered structure, where the vector orthogonal to the layer directions is constant and given by $n = (0, 0, 1)^T$.

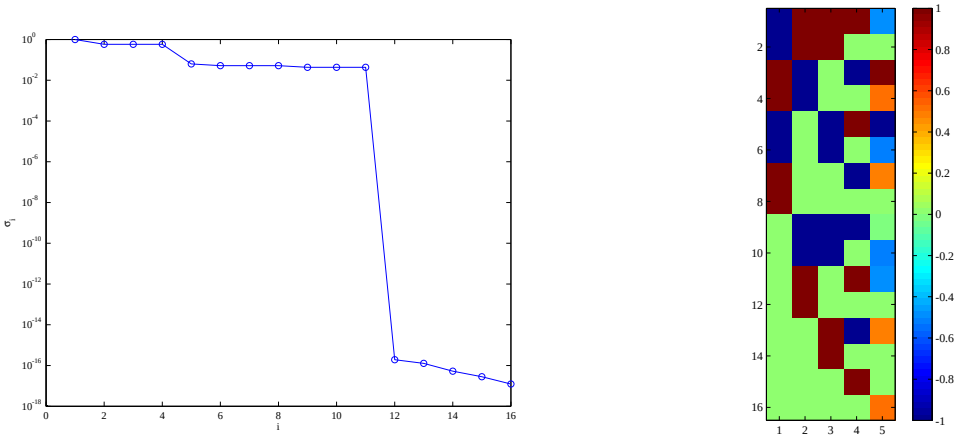


(a) Singular values.



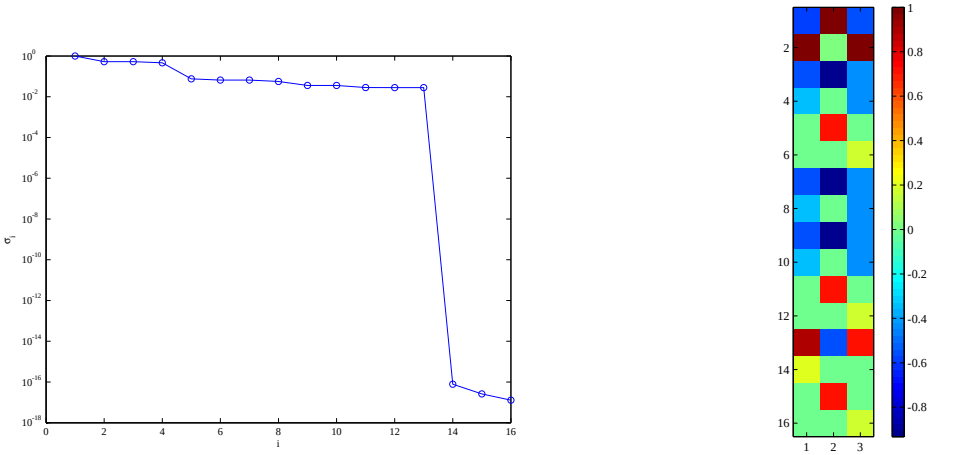
(b) Rational basis for null Q.

Figure 4.7: Singular values of Q for the perturbed mesh and vectors forming the rational basis for null Q. The conductivity here was constrained to be formed by a layered structure, where the vector orthogonal to the layer directions is constant and given by $n = (0, 0, 1)^T$.



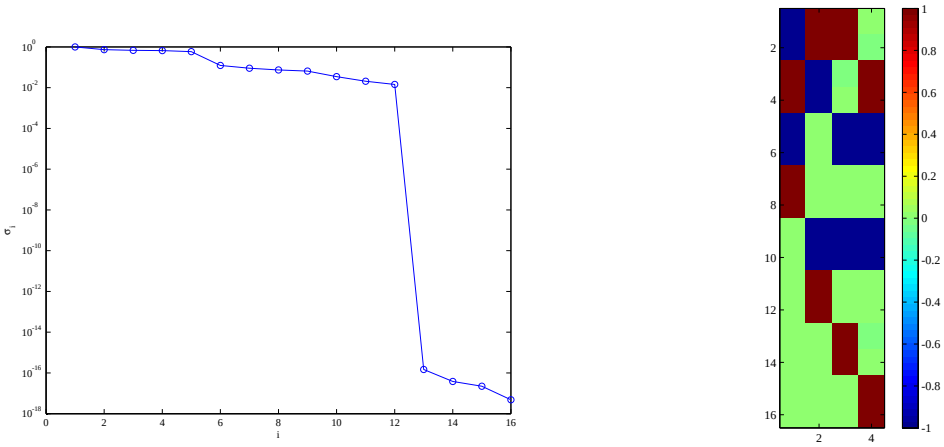
(a) Singular values. (b) Rational basis for null Q.

Figure 4.8: Singular values of Q for the original mesh and vectors forming the rational basis for null Q. The conductivity here was constrained to be formed by a layered structure, with layers perpendicular to the radial direction.



(a) Singular values. (b) Rational basis for null Q.

Figure 4.9: Singular values of Q for the perturbed mesh and vectors forming the rational basis for null Q. The conductivity here was constrained to be formed by a layered structure, with layers perpendicular to the radial direction.



(a) Singular values. (b) Rational basis for null Q.

Figure 4.10: Singular values of Q for the original mesh and vectors forming the rational basis for null Q. The conductivity here was constrained to be formed by a layered structure, where the vector orthogonal to the layers are random perturbations of $(0, 0, 1)^T$.

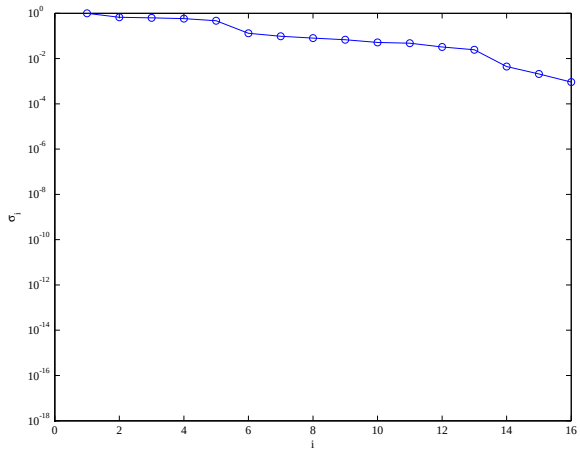
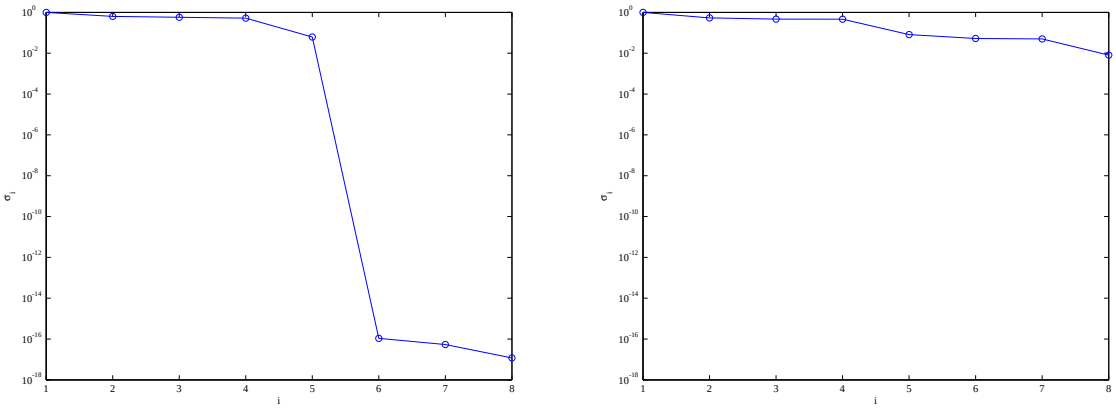
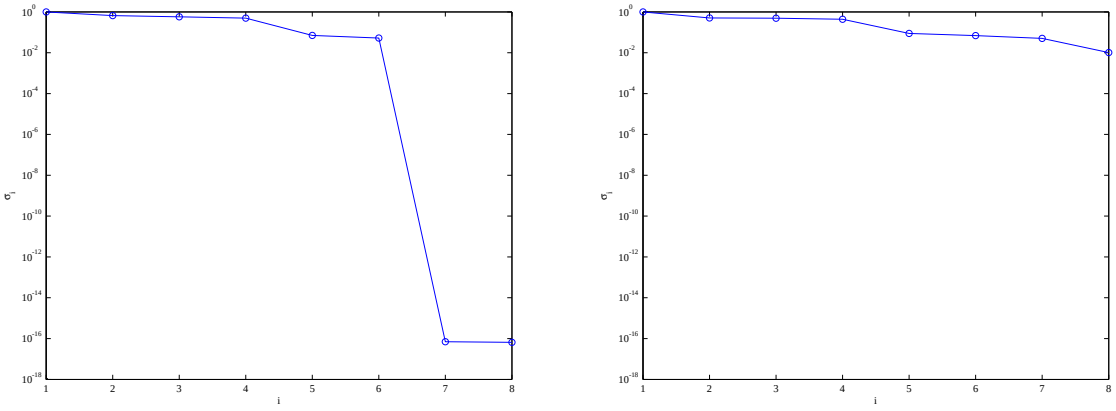


Figure 4.11: Singular values of Q for the perturbed mesh. The conductivity here was constrained to be formed by a layered structure, where the vector orthogonal to the layers are random perturbations of $(0, 0, 1)^T$.



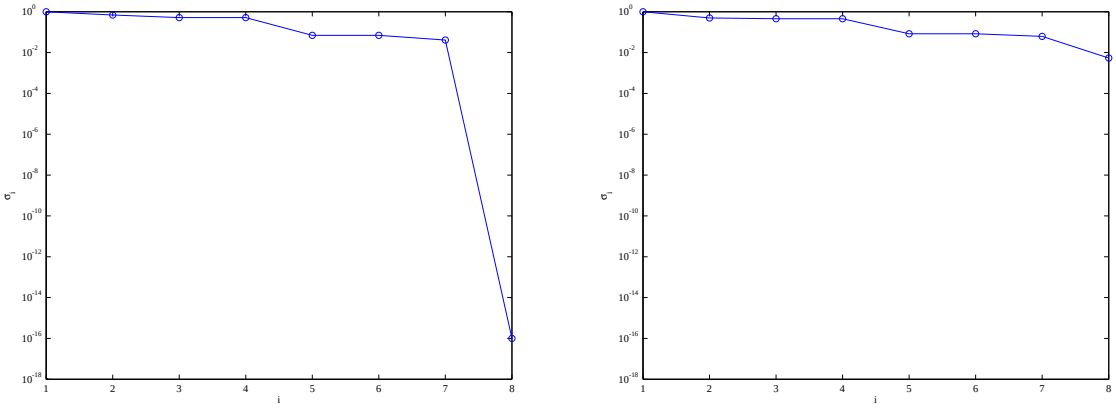
(a) Original mesh. (b) Perturbed mesh.

Figure 4.12: Singular values of Q for the original and perturbed meshes for conformal anisotropy with constant matrix function A, (see text for A).



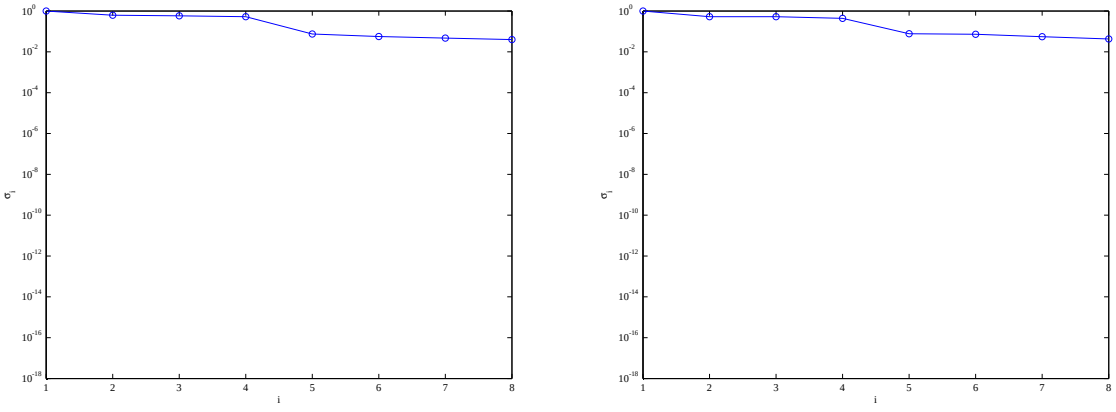
(a) Original mesh. (b) Perturbed mesh.

Figure 4.13: Singular values of Q for the original and perturbed meshes for conformal anisotropy with constant matrix function B, (see text for B).



(a) Original mesh. (b) Perturbed mesh.

Figure 4.14: Singular values of Q for the original and perturbed meshes for conformal anisotropy with constant matrix function C, (see text for C).



(a) Original mesh. (b) Perturbed mesh.

Figure 4.15: Singular values of Q for the original and perturbed meshes for conformal anisotropy with the known matrix given as a random perturbation of the identity in each tetrahedron.

4.4 Embedding a finite element discretisation in $\mathbb{R}^3$

In this section we will extend the survey problem of chapter 3 to that of embedding a finite element mesh in $\mathbb{R}^3$, such that the embedding is consistent with a known set of edge conductance values. This problem is the second part of the inverse problem that we described earlier. That is, given some edge conductance values, can we find an embedding of the mesh and a conductivity distribution defined on this mesh, that is consistent with the edge conductances?

If we were to restrict the problem to the case of finding an isotropic conductivity distribution and an embedding then it would make sense to pose the problem in terms of finding all the dihedral angles of the embedded mesh, since the dihedral angles can be used to directly compute edge conductances through the cotangent formula. However, it is our aim to investigate the anisotropic problem in which the dihedral angles cannot be used to compute the edge conductances. We assume that our boundary data is available at every boundary vertex, therefore in order to ensure that we have the correct positions for our boundary data we fix the positions of all the boundary vertices. Then our problem becomes: Given a simplicial complex, a set of edge conductance values and a known surface mesh embedded in $\mathbb{R}^3$ representing the boundary of the domain, find the interior vertex positions and piecewise constant anisotropic conductivity distribution that is consistent with the edge conductances. In this section we present some numerical results that illustrate how the non-uniqueness of EIT appears in the discretised problem.

We start with an embedded mesh and an initial conductivity distribution. We then apply a random perturbation to the conductivity and compute the edge conductances. We then attempt to fit the interior vertices and conductivities to the perturbed edge conductances. We use MATLAB's `fsolve` optimisation function to find new vertex positions and conductivity values that are consistent with the edge conductances. We use the original mesh and conductivities as the initial guess. We can apply the SVD to the Jacobian of the constraints computed at the converged solution to investigate the uniqueness of the problem.

For the numerical example we will use the mesh shown in figure 3.8. The figures will will discuss can be found at the end of this chapter. Table 4.1, shows the relative errors

Conductivity type	Vertex error $\left(\frac{\|v-v_0\|_2}{\|v_0\|_2}\right)$	Conductivity error $\left(\frac{\|\sigma-\sigma_0\|_2}{\|\sigma_0\|_2}\right)$
Isotropic	3.98×10^{-10}	1.08×10^{-7}
Anisotropic	7.36×10^{-3}	1.02
Layered	1.86×10^{-3}	1.34×10^{-1}
Conformal	8.61×10^{-10}	9.25×10^{-8}

Table 4.1: Relative errors for converged solution of embedding problem.

for the vertex coordinates and the conductivity values. For the vertex coordinates, the relative error is the norm of the difference between the converged values and the initial coordinates, divided by the norm of the initial coordinates. For the conductivity values, the error is the norm of the difference between the converged values and the perturbed values that were used to compute the edge conductances. We can see from the table that the isotropic and conformal conductivity cases have relatively small error compared to the anisotropic and layered conductivity cases. From figure 4.16 we can see that the Jacobian at the converged solution for both the isotropic and conformal cases has full rank, therefore we can expect to find a locally unique solution to the problem. This suggests that the vertex position and conductivity errors for these cases is due to compounded floating point error in calculations in the optimisation process. If the vertex positions have not changed from the initial values and the error in the conductivity values is small then this suggests that the change in edge conductance values is due only to the change in conductivity that we applied. In the cases of fully anisotropic and anisotropy due to layers, we see that the errors are much larger than we would expect from compounded floating point error. From figure 4.16, we can see that the Jacobian at the converged solution is rank-deficient, therefore there is not a unique combination of vertex coordinates and conductivities that produces the given edge conductances. The errors in the table show that the optimisation algorithm found a set of interior vertices and conductivities different to those that initially produced the edge conductances but which are still consistent with the edge conductances. This suggests that the difference between the initial vertex positions and the vertex positions from the converged solution represents a linearisation of a discrete diffeomorphism that causes the non-uniqueness in the anisotropic EIT problem.

We note here that we used an analytic expression for the Jacobian J rather than the finite difference approximation utilised by the `fsolve` function. The derivative of

the edge conductance constraints with respect to the conductivity variables is just the matrix Q. The derivative of the geometric constraints with respect to the conductivity is clearly zero. The derivative of the geometric constraints with respect to the vertex coordinates are also zero, since perturbing a vertex causes the dihedral angles in all the tetrahedra sharing the vertex to be perturbed, such that the geometric constraints are still satisfied. This is because perturbing a vertex does not change the connectivity of the mesh and so the relevant edges move with the vertex. We therefore do not need to apply the geometric constraints when we perturb the interior vertex coordinates. The geometric constraints would be important if we had parameterised the problem in terms of dihedral angles and conductivity variables, as was done in [5] for the planar mesh case.

In order to give an expression for the derivative of the edge conductances with respect to the vertex positions, we will consider a single tetrahedron $T = \{i, j, k, l\}$ and edge conductance γ_e for edge $e = \{i, j\}$. Clearly the vertex positions only appear in the edge conductances through the linear map represented by the matrix Q so the derivative of the edge conductances with respect to the vertex coordinates is $J_Q s$, where J_Q is the Jacobian of Q with respect to the vertex coordinates and $s \in \mathbb{R}^{6n_T}$ is the usual conductivity variable vector for a general anisotropic distribution. We assume that the tetrahedron is positively oriented such that

$$\det V = \det\left(x_i - x_l, x_j - x_l, x_k - x_l\right),$$

is positive. Let $\langle c_i, c_j \rangle_r$ be the component of the inner product $c_i \cdot c_j$ from (4.10) corresponding to the rth conductivity variable defined on T, where c_i, c_j are defined in (4.5). For example, for Q_{e1}, $\langle c_i, c_j \rangle_1 = c_i^1 c_j^1$. Then for the pth coordinate of the qth vertex, where $q = i, j, k, l$, we have the following expressions for the components of Q_e on tetrahedron T

$$\frac{\partial Q_{e,r}}{\partial x_q^p} = \frac{1}{6(\det V)^2}\left(\frac{\partial \langle c_i, c_j \rangle_r}{\partial x_q^p} - \frac{\det V}{6}\mathrm{tr}\left(V^{-1}\frac{\partial V}{\partial x_q^p}\right)\langle c_i, c_j \rangle_r\right). \tag{4.13}$$

Written in this form and comparing to the Lie derivative of the Hodge star, [91], [75], we can see that (4.13) is the discretisation of the Lie derivative of the conductivity with respect to the vertex positions. The last term is the derivative of the tetrahedron volume with respect to the vertex coordinate. We have the following expression for

the derivative of the volume $|T| = (\det V)/6$,

$$\frac{\partial |T|}{\partial x_q^p} = \begin{cases} -c_q^p, & \text{for } q = i, j, k, \\ c_i^p + c_j^p + c_k^p, & \text{for } q = l, \end{cases} \tag{4.14}$$

where we define $c_k = (x_i - x_l) \times (x_j - x_l)$. To compute the partial derivatives of the c_i and c_j we introduce the function P_{pq} which acting on a vector, picks out the pth component of a permutation of the vector as follows,

$$P_{pq}(x) = x^{r_1}\delta_{pr_2} - x^{r_2}\delta_{pr_1},$$

where $r = \{r_1, r_2\}$ is an even permutation of $\{1, 2, 3\}$ with q removed, x^r is the rth coordinate of the vector x and δ_{pr} is the Kronecker delta. Then for $p, q = 1, 2, 3$,

$$\begin{aligned}
&\frac{\partial c_i^q}{\partial x_i^p} = 0, && \frac{\partial c_j^q}{\partial x_i^p} = P_{pq}(x_l - x_k), \\
&\frac{\partial c_i^q}{\partial x_j^p} = P_{pq}(x_k - x_l), && \frac{\partial c_j^q}{\partial x_j^p} = 0, \\
&\frac{\partial c_i^q}{\partial x_k^p} = P_{pq}(x_l - x_j), && \frac{\partial c_j^q}{\partial x_k^p} = P_{pq}(x_i - x_l), \\
&\frac{\partial c_i^q}{\partial x_l^p} = P_{pq}((x_j - x_l) - (x_k - x_l)), && \frac{\partial c_j^q}{\partial x_l^p} = P_{pq}((x_k - x_l) - (x_i - x_l)).
\end{aligned}$$

Then inserting these expressions, along with (4.14), into (4.13) gives the Jacobian for each term with respect to the relevant vertex coordinate. For the different classes of anisotropy, we can transform the conductivity variables into the anisotropic conductivity variables for each tetrahedron by applying the relevant constraints. For example, the anisotropic conductivity variables for the isotropic case for a single tetrahedron t are simply given by $s_t = \sigma_t \, (1, 0, 0, 1, 0, 1)^T$, where σ_t is the scalar conductivity for t.

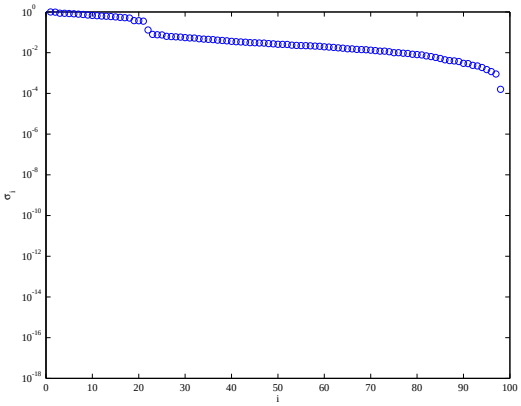
If the geometric constraints are included in the optimisation and a finite difference approximation is used to calculate the Jacobian, then the singular values exhibit odd behaviour. Figure 4.17 illustrates the effect that the step size of the finite difference approximation has on the singular values of the Jacobian. We can see that as the step size decreases the finite difference approximation actually gets worse and results in non-zero singular values that should in fact be zero. Clearly, the maximum rank of the Jacobian is the number of edges in the mesh, since we only have this many edge conductances from which to determine the interior vertex positions and conductivity

values. We can see from these plots that there are two distinct groupings of the non-zero singular values in both cases. This is due to the difference in scales between the vertex positions and the conductivity variables. Figure 4.18 shows the singular values after correcting for the difference in scaling and removing the geometric constraints from the Jacobian. We can see here that the step change in the singular values is no longer apparent when the scale correction is applied. We observe that J now has full row rank and the null space is due to the extra number of variables compared to the number of edge conductances. Figure 4.19 shows the corresponding bases for the row and null spaces of J defined by the right singular vectors. We can see that the basis vectors corresponding to the largest singular values have larger terms in the entries that correspond to the interior vertex coordinates. This suggests that the vertex coordinates are easier to find from the edge conductances than the conductivity variables. It also shows that a perturbation applied to the vertex coordinates has a larger effect on the edge conductances than a perturbation in conductivities. Figures 4.19b and 4.19d also provide evidence for this, since the entries corresponding to the conductivity variables are much larger than those corresponding to the vertex coordinates. Therefore small changes in the conductivity will have less of an effect on the edge conductance values. If the non-uniqueness was mainly due to the diffeomorphism invariance, then we would expect the components of the null space corresponding to conductivity and vertices to have roughly the same magnitude. Therefore, these results suggest that the non-uniqueness caused by the null space of the linear map $\mathcal{Q}_{X,\mathcal{K}}$ dominates the effect of the diffeomorphism invariance in the discrete anisotropic EIT problem. In order to ensure a unique solution to the discrete problem we should ensure that the total number of interior vertex coordinates and conductivity variables is less than the number of edges in the mesh.

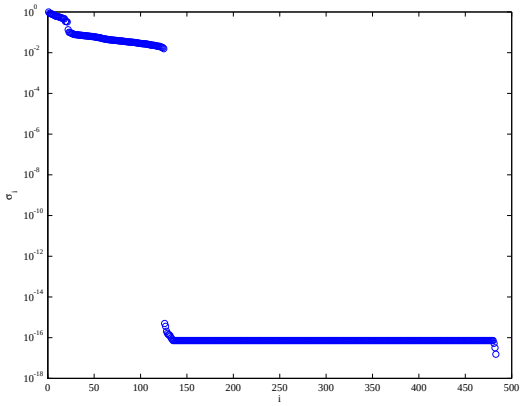
4.5 Conclusions

In this chapter we have shown that the FE system matrix is equivalent to the conductance values of a resistor network, whose resistors are positioned on the edges of the finite element mesh. We introduced the linear map $\mathcal{Q}_{X,\mathcal{K}}$ that maps conductivity values to edge conductances for a given embedding of a simplicial complex in three

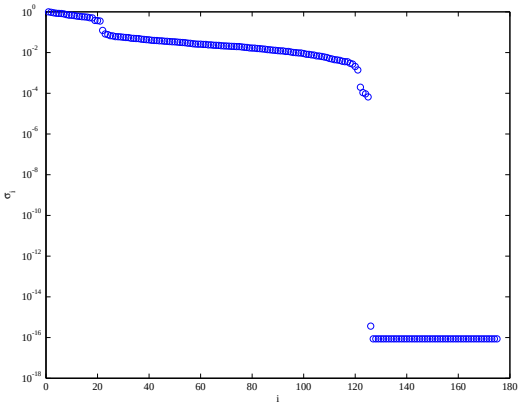
dimensions. We have shown how the conductivity can be parameterised using a priori information about the type of anisotropy that is present. For these different classes of anisotropy, we presented a small numerical example that illustrates the non-injectivity of the linear map when the mesh has certain properties. We then considered the problem of finding the interior vertex positions of the mesh and the piecewise constant conductivity distribution that is consistent with a given set of edge conductances. We have shown how using a finite difference approximation to calculate the Jacobian of the objective function can be misleading and presented analytic expressions for the derivative of the edge conductances with respect to vertex positions and conductivities. We then illustrated that the non-uniqueness of the discrete EIT problem is due mainly to the non-injectivity of the linear map $\mathcal{Q}_{X,\mathcal{K}}$ as opposed to the diffeomorphism invariance of the anisotropic EIT problem. The diffeomorphism invariance still plays a part in the anisotropic problem but is not as big an issue as the non-uniqueness caused by having more variables than edge conductances.



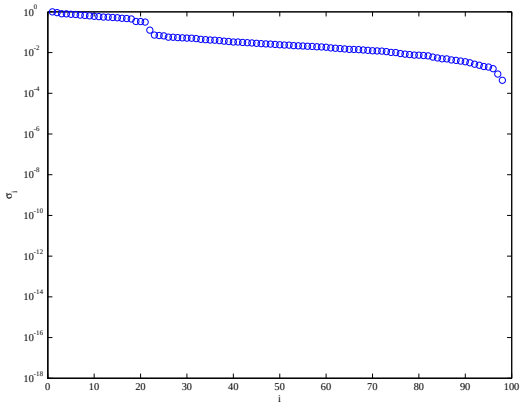
(a) Isotropic.



(b) Anisotropic.

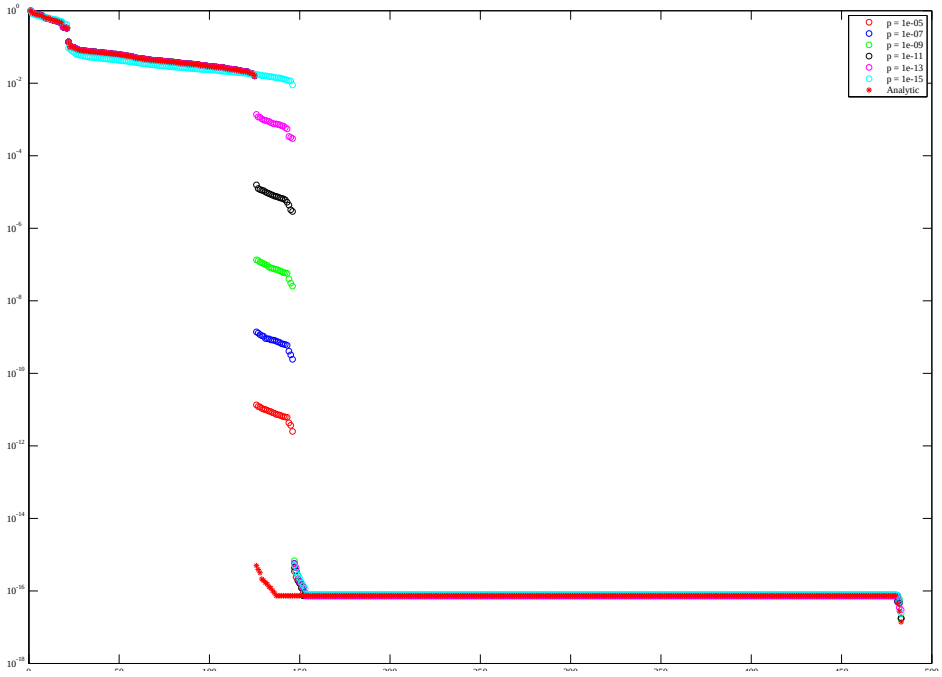


(c) Layered.

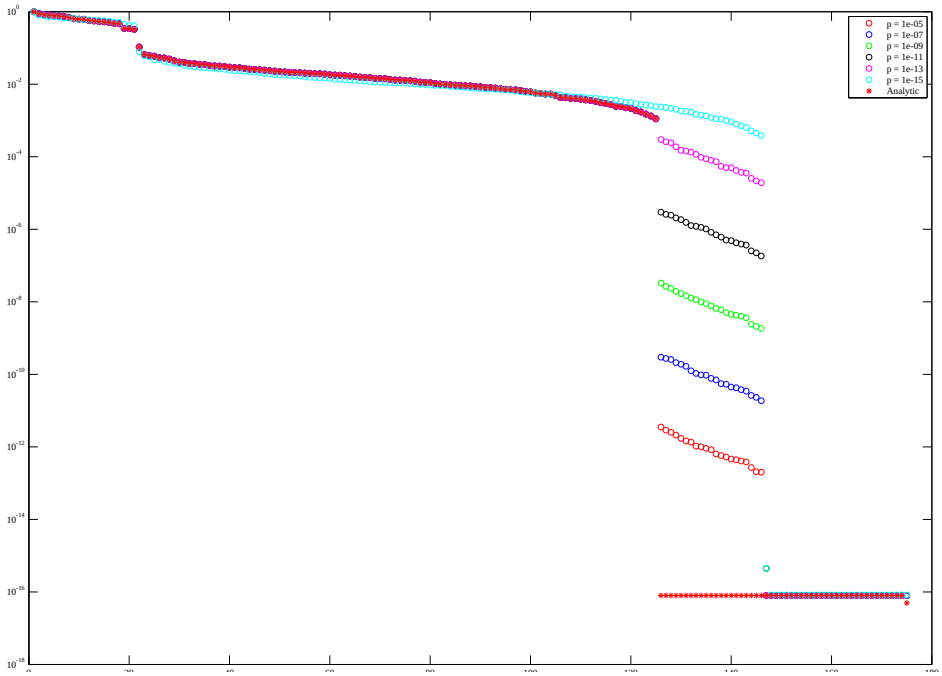


(d) Conformal.

Figure 4.16: Singular values of Jacobian at converged solution for different classes of conductivity.

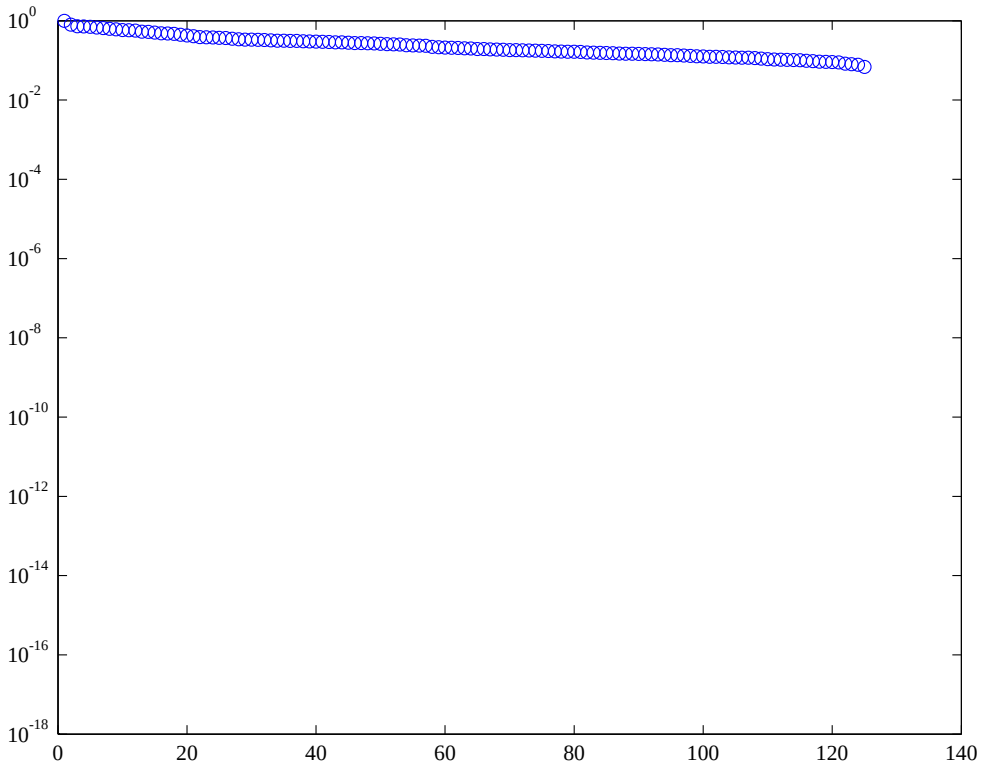


(a) Anisotropic.

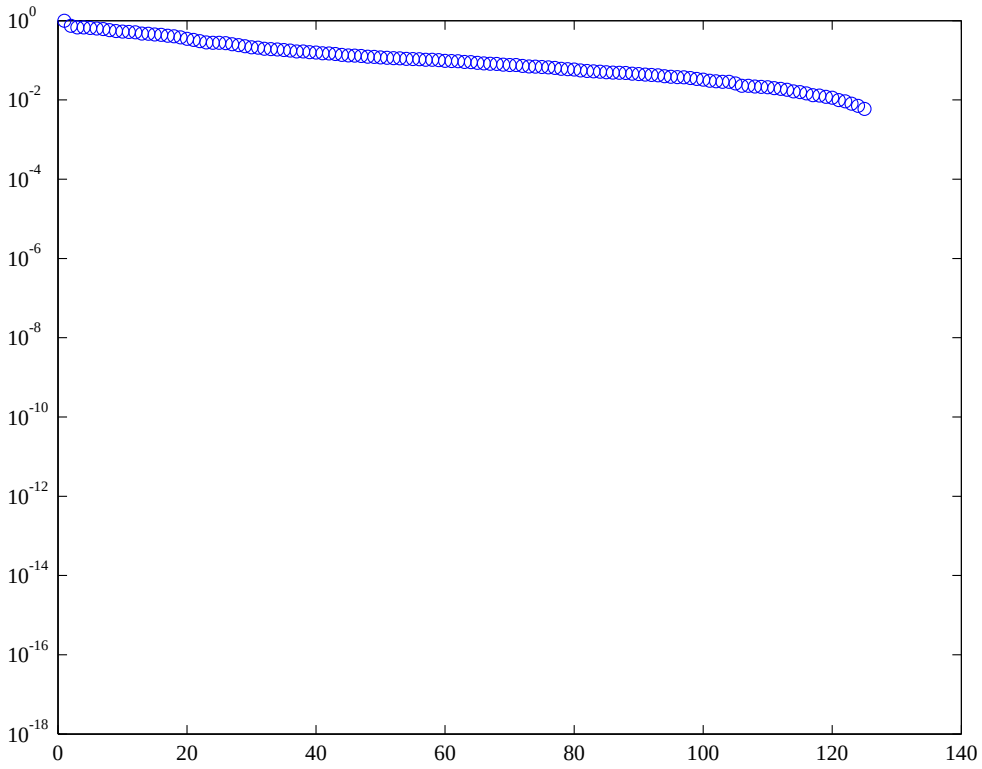


(b) Layers perpendicular to radial direction.

Figure 4.17: Comparison of analytic and finite difference approximations of the Jacobian at converged solution with various finite difference step sizes, for two classes of conductivities.

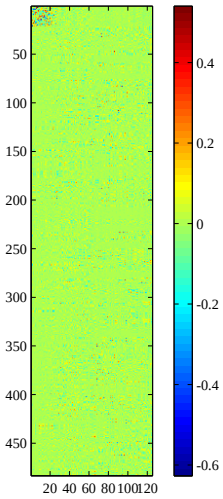


(a) Anisotropic.

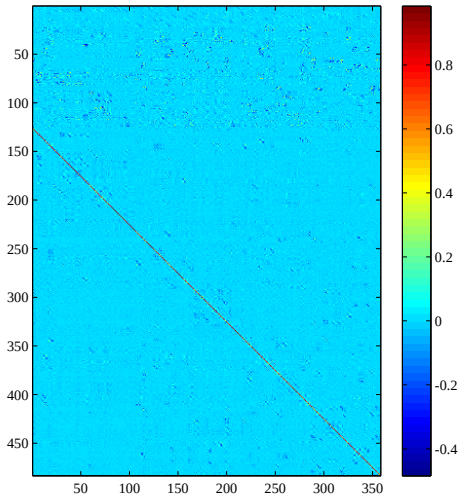


(b) Layers perpendicular to radial direction.

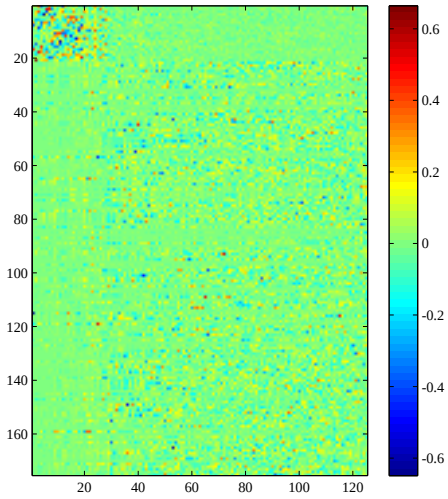
Figure 4.18: Singular values of Jacobian with difference in scaling between vertex positions and conductivities accounted for.



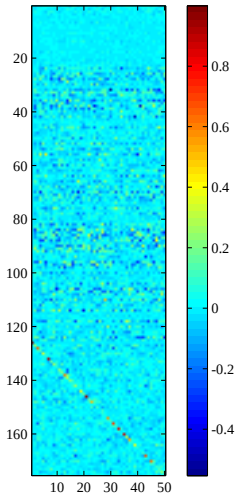
(a) Anisotropic, row J.



(b) Anisotropic, null J.



(c) Layers perpendicular to radial direction, row J.



(d) Layers perpendicular to radial direction, null J.

Figure 4.19: Singular vectors that provide bases Row and null spaces of the analytic Jacobian of the edge conductances for anisotropic (top) and layers perpendicular to the radial direction (bottom).

Chapter 5

Conclusions and future work

5.1 Summary

In this thesis we have bridged the gap between theoretical results based on the continuous EIT problem and the more practical discrete problem. Firstly, we have extended the two-dimensional work of [5] and provided a set of independent consistency conditions on the dihedral angles for a three-dimensional simplicial complex to be uniquely embedded in $\mathbb{R}^3$. The planar π constraint on triangles has been extended to the spherical Cayley-Menger condition on tetrahedra. For each interior face we have the condition that the face angles calculated from the dihedral angles of two adjacent tetrahedra must be equal. To ensure that the embedding is a flat triangulated manifold we extend the planar curvature condition to edge curvature, which is the discrete analogue of Ricci curvature. To prove that a set of dihedral angles that satisfy these constraints result in a unique embedding, we have developed a constructive procedure which utilises the spanning tree of the graph defined on the 0- and 1-cells of the dual complex. For each vertex we visit in this graph, we identify the independent dihedral angles, and the independent constraints that the other dihedral angles must satisfy. This constructive procedure provides an algorithmic way of determining the constraints we need and is implemented in a numerical example to illustrate that the local uniqueness result holds in practice.

We then consider the problem of embedding a triangulated 3-manifold with boundary in $\mathbb{R}^3$, such that the embedding is consistent with a finite element mesh with

piecewise constant conductivity. We observed that this problem is equivalent to finding an embedding of a three-dimensional resistor network whose conductances are equal to the finite element system matrix. We parameterise the problem in terms of the embedded interior vertex positions and anisotropic conductivity defined on each tetrahedron. We parameterise the conductivity depending on the a priori information available about the anisotropy and we define the linear map Q from conductivity components to edge conductances for different classes of anisotropy. We used a simplistic example to illustrate the effect that a poorly chosen mesh can have on the injectivity of this linear map. We also investigated the effect of the anisotropy parameterisation on the uniqueness of the discrete EIT problem when the interior vertex coordinates are not fixed. We found that the non-uniqueness due to the non-injectivity of Q is a bigger issue than the non-uniqueness of the anisotropic problem due to the diffeomorphism invariance inherited from the continuous problem.

5.2 Future Work

In order to extend the result of [83], a possible direction of research is the implementation of the discrete Hodge star as defined in section 2.4, in the context of the discrete EIT problem. Since the discrete Hodge star relates k-cochains to their duals, it seems plausible that a circumsphere representation similar to the circumcircle parameterisation from [83] could result in a three-dimensional Ohm-Kirchhoff network defined on the dual complex. Since this would rely on the cotangent formula for isotropic conductivity it would not be applicable to the anisotropic problem but it would at least give a theoretical result for the isotropic case.

In order to allow more conductivity variables to be defined in the anisotropic parameterisation, an extension of our work to higher order conductivity approximation would be be advantageous. In [1], a piecewise linear approximation is used which allows the conductivity to be defined at vertices, and the Jacobian of the boundary data with respect to conductivity variables is shown to be full rank for some example meshes. Since the number of vertices in a three-dimensional mesh is typically less that the number of tetrahedra, this may lead to more classes of anisotropic conductivity that can be uniquely reconstructed.

Bibliography

[1] J.-F. P. J. Abascal, W. R. B. Lionheart, S. R. Arridge, M. Schweiger, M. Atkinson, and D. S. Holder. Electrical impedance tomography in anisotropic media with known eigenvectors. *Inverse Problems*, 27(6):065004, June 2011.

[2] R. Abraham, J. E. Marsden, and T. Ratiu. *Manifolds, Tensor Analysis, and Applications*. Number 75 in Applied Mathematical Sciences. Springer New York, Jan. 1988.

[3] A. Adler, M. B. Amato, J. H. Arnold, R. Bayford, M. Bodenstein, S. H. Bhm, B. H. Brown, I. Frerichs, O. Stenqvist, N. Weiler, and G. K. Wolf. Whither lung EIT: Where are we, where do we want to go and what do we need to get there? *Physiological Measurement*, 33(5):679, May 2012.

[4] A. Adler, R. Gaburro, and W. Lionheart. Electrical impedance tomography. In O. Scherzer, editor, *Handbook of Mathematical Methods in Imaging*, pages 599–654. Springer New York, Jan. 2011.

[5] A. Al-Humaidi. *Resistor networks and finite element models*. PhD thesis, University of Manchester, 2011.

[6] G. Alessandrini. Stable determination of conductivity by boundary measurements. *Applicable Analysis*, 27(1):153–172, 1988.

[7] G. Alessandrini. Singular solutions of elliptic equations and the determination of conductivity by boundary measurements. *Journal of Differential Equations*, 84(2):252–272, Apr. 1990.

[8] G. Alessandrini. Open issues of stability for the inverse conductivity problem. *Journal of Inverse and Ill-posed Problems*, 15(5):451–460, Nov. 2007.

[9] G. Alessandrini and R. Gaburro. Determining conductivity with special

anisotropy by boundary measurements. *SIAM Journal on Mathematical Analysis*, 33(1):153, 2001.

[10] G. Alessandrini, V. Isakov, and J. Powell. Local uniqueness in the inverse conductivity problem with one measurement. *Transactions of the American Mathematical Society*, 347(8):3031–3041, 1995. ArticleType: research-article / Full publication date: Aug., 1995 / Copyright 1995 American Mathematical Society.

[11] Allen Hatcher. *Algebraic topology / Allen Hatcher.* Cambridge University Press, Cambridge, 2002.

[12] K. Astala and L. Päivärinta. Calderón's inverse conductivity problem in the plane. *Annals of Mathematics-Second Series*, 163(1):265300, 2006.

[13] K. Astala, L. Päivärinta, and M. Lassas. Calderón's inverse problem for anisotropic conductivity in the plane. *Communications in Partial Differential Equations*, 30(1-2):207–224, 2005.

[14] D. Audet. Déterminants sphérique et hyperbolique de cayley-menger. *Bulletin AMQ*, 51(2):45–52, 2011.

[15] D. C. Barber and B. H. Brown. Applied potential tomography. *Journal of Physics E: Scientific Instruments*, 17(9):723, Sept. 1984.

[16] R. D. Barker. Depth of investigation of collinear symmetrical four-electrode arrays. *Geophysics*, 54(8):1031, Aug. 1989.

[17] B. Bollobás. *Modern Graph Theory.* Number 184 in Graduate Texts in Mathematics. Springer New York, Jan. 1998.

[18] L. Borcea, V. Druskin, and A. V. Mamonov. Circular resistor networks for electrical impedance tomography with partial boundary measurements. *Inverse Problems*, 26(4):045010, Apr. 2010.

[19] L. Borcea, V. Druskin, A. V. Mamonov, and F. G. Vasquez. Pyramidal resistor networks for electrical impedance tomography with partial boundary measurements. *Inverse Problems*, 26(10):105009, Oct. 2010.

[20] L. Borcea, V. Druskin, and F. G. Vasquez. Electrical impedance tomography with resistor networks. *Inverse Problems*, 24(3):035013, June 2008.

[21] A. Bossavit. *Computational Electromagnetism: Variational Formulations, Complementarity, Edge Elements*. Academic Press, Feb. 1998.

[22] G. E. Bredon. *Topology and Geometry*. Number 139 in Graduate Texts in Mathematics. Springer New York, Jan. 1993.

[23] B. Brown and D. Barber. Applied potential tomography-a new in vivo medical imaging technique. *Clin. Phys. Physiol. Meas*, 4(1), 1982.

[24] R. M. Brown. Global uniqueness in the impedance-imaging problem for less regular conductivities. *SIAM Journal on Mathematical Analysis*, 27(4):1049, 1996.

[25] R. M. Brown and G. A. Uhlmann. Uniqueness in the inverse conductivity problem for nonsmooth conductivities in two dimensions. *Communications in Partial Differential Equations*, 22(5):1009–1027, 1997.

[26] M. Brühl. Explicit characterization of inclusions in electrical impedance tomography. *SIAM Journal on Mathematical Analysis*, 32(6):1327, 2001.

[27] A. P. Calderón. On an inverse boundary value problem. In *Seminar on numerical analysis and its applications to continuum mechanics*, pages 65–73, Rio de Janeiro: Sociedad Brasileiria de Matematica, 1980.

[28] A. Cayley. A theorem in the geometry of position. *Cambridge Mathematical Journal*, 2:267–271, 1841.

[29] H. Chauris, W. Lionheart, and A. Adler. *100 Years of Electrical Imaging*. Presses des MINES, 2012.

[30] J. Cheeger, W. Mller, and R. Schrader. On the curvature of piecewise flat spaces. *Communications in Mathematical Physics*, 92(3):405–454, Sept. 1984.

[31] K.-S. Cheng, D. Isaacson, J. C. Newell, and D. Gisser. Electrode models for electric current computed tomography. *IEEE Transactions on Biomedical Engineering*, 36(9):918–924, Sept. 1989.

[32] M. H. Choi, T.-J. Kao, D. Isaacson, G. Saulnier, and J. Newell. A reconstruction algorithm for breast cancer imaging with electrical impedance tomography in mammography geometry. *IEEE Transactions on Biomedical Engineering*, 54(4):700–710, Apr. 2007.

[33] G. M. Crippen. A novel approach to calculation of conformation: Distance geometry. *Journal of Computational Physics*, 24(1):96–107, May 1977.

[34] E. Curtis and J. Morrow. Determining the resistors in a network. *SIAM Journal on Applied Mathematics*, 50(3):918–930, June 1990.

[35] E. Curtis and J. Morrow. The dirichlet to neumann map for a resistor network. *SIAM Journal on Applied Mathematics*, 51(4):1011–1029, Aug. 1991.

[36] E. B. Curtis. *Inverse Problems for Electrical Networks*. World Scientific Publishing Co Pte Ltd, Singapore ; River Edge, NJ, Mar. 2000.

[37] E. B. Curtis, D. Ingerman, and J. A. Morrow. Circular planar graphs and resistor networks. *Linear Algebra and its Applications*, 283(13):115–150, Nov. 1998.

[38] A. J. Delaney, P. R. Peapples, and S. A. Arcone. Electrical resistivity of frozen and petroleum-contaminated fine-grained soil. *Cold Regions Science and Technology*, 32(23):107–119, Sept. 2001.

[39] M. Desbrun, E. Kanso, and Y. Tong. Discrete differential forms for computational modeling. In A. I. Bobenko, J. M. Sullivan, P. Schrder, and G. M. Ziegler, editors, *Discrete Differential Geometry*, number 38 in Oberwolfach Seminars, pages 287–324. Birkhuser Basel, Jan. 2008.

[40] A. Dey and H. F. Morrison. Resistivity modelling for arbitrarily shaped two-dimensional structures. *Geophysical Prospecting*, 27(1):106136, 1979.

[41] M. P. do Carmo. *Riemannian Geometry*. Mathematics: Theory and Applications. Birkhuser Basel, 1992.

[42] A. W. M. Dress and T. F. Havel. Distance geometry and geometric algebra. *Foundations of Physics*, 23(10):1357–1374, Oct. 1993.

[43] R. J. Duffin. Nonlinear networks. i. *Bulletin of the American Mathematical Society*, 52(10):833–838, 1946.

[44] H. C. Elman, D. Silvester, and A. Wathen. *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*. Oxford University Press, 2014.

[45] J. Forsyth, A. Borsic, R. J. Halter, A. Hartov, and K. D. Paulsen. Optical

breast shape capture and finite-element mesh generation for electrical impedance tomography. *Physiological Measurement*, 32(7):797, July 2011.

[46] T. Frankel. *The Geometry of Physics: An Introduction. 1997*. Cambridge, UK: Cambridge University Press, 1999.

[47] D. Glickenstein. Discrete conformal variations and scalar curvature on piecewise flat two- and three-dimensional manifolds. *Journal of Differential Geometry*, 87(2):201–238, Feb. 2011.

[48] G. H. Golub and C. F. Van Loan. *Matrix Computations*. JHU Press, Dec. 2012.

[49] P. W. Gross and P. R. Kotiuga. *Electromagnetic Theory and Computation - A Topological Approach*. Number 48 in MSRI Monograph Series. Cambridge University Press, 2004.

[50] B. Haberman and D. Tataru. Uniqueness in calderóns problem with lipschitz conductivities. *Duke Mathematical Journal*, 162(3):497–516, 2013.

[51] G. W. Hart, S. Levy, and R. McLenaghan. Spherical geometry and trigonometry. In D. Zwillinger, editor, *CRC standard mathematical tables and formulae*. Chapman & Hall/CRC, Boca Raton, Fla; London, 31st ed. edition, 2003.

[52] T. F. Havel. Distance geometry: Theory, algorithms, and chemical applications. In *Encyclopedia of Computational Chemistry*. John Wiley & Sons, Ltd, 1998.

[53] H. Heck and J.-N. Wang. Stability estimates for the inverse boundary value problem by partial cauchy data. *Inverse Problems*, 22(5):1787–1796, Oct. 2006.

[54] Herbert Goldstein. *Classical mechanics / Herbert Goldstein.* Addison-Wesley, Reading, Mass; London, 2nd ed. edition, 1980.

[55] A. N. Hirani. *Discrete exterior calculus*. phd, California Institute of Technology, 2003.

[56] D. Holder. *Electrical impedance tomography methods, history, and applications*. Institute of Physics Pub., Bristol, Philadelphia, 2005.

[57] S. Iglin. http://iglin.exponenta.ru/gr.html. Accessed November 2010.

[58] V. Isakov. On uniqueness of recovery of a discontinuous conductivity coefficient. *Communications on Pure and Applied Mathematics*, 41(7):865–877, Oct. 1988.

[59] V. Isakov. On uniqueness in the inverse conductivity problem with local data. *Inverse Problems and Imaging*, 1(1):95, 2007.

[60] V. Isakov and J. Powell. On the inverse conductivity problem with one measurement. *Inverse Problems*, 6(2):311–318, Apr. 1990.

[61] D. Jungnickel. *Graphs, Networks and Algorithms*. Number 5 in Algorithms and Computation in Mathematics. Springer Berlin Heidelberg, Jan. 2013.

[62] C. Kenig, G. Uhlmann, and J. Sjstrand. The calderón problem with partial data. *Annals of Mathematics*, 165(2):567–591, Mar. 2007.

[63] R. Kohn and M. Vogelius. Determining conductivity by boundary measurements. *Communications on Pure and Applied Mathematics*, 37(3):289–298, May 1984.

[64] R. V. Kohn and M. Vogelius. Identification of an unknown conductivity by means of measurements at the boundary. In *Proceedings SIAM-AMS Symposium on Inverse Problems*, pages 113—123, New York, 1983.

[65] R. V. Kohn and M. Vogelius. Determining conductivity by boundary measurements II. interior results. *Communications on Pure and Applied Mathematics*, 38(5):643–667, Sept. 1985.

[66] M. Lassas, M. Salo, and L. Tzou. Inverse problems and invisibility cloaking for FEM models and resistor networks. *arXiv:1307.1539 [math]*, July 2013. arXiv: 1307.1539.

[67] M. Lassas, M. Taylor, and G. Uhlmann. The dirchlet-to-neumann map for complete riemannian manifolds with boundary. *Communications in Analysis and Geometry*, 11(2):207222, 2003.

[68] M. Lassas and G. Uhlmann. On determining a riemannian manifold from the dirichlet-to-neumann map. *Annales Scientifiques de l'école Normale Supérieure*, 34(5):771–787, Sept. 2001.

[69] C. Lavor, L. Liberti, N. Maculan, and A. Mucherino. Recent advances on the discretizable molecular distance geometry problem. *European Journal of Operational Research*, 219(3):698–706, June 2012.

[70] G. Lawler and J. Sylvester. Determining resistances from boundary measurements

in finite networks. *SIAM Journal on Discrete Mathematics*, 2(2):231–239, May 1989.

[71] J. M. Lee and G. Uhlmann. Determining anisotropic real-analytic conductivities by boundary measurements. *Communications on Pure and Applied Mathematics*, 42(8):10971112, 1989.

[72] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino. Euclidean distance geometry and applications. *SIAM Review*, 56(1):3–69, Jan. 2014.

[73] W. R. B. Lionheart. Conformal uniqueness results in anisotropic electrical impedance imaging. *Inverse Problems*, 13(1):125, Feb. 1997.

[74] W. R. B. Lionheart. EIT reconstruction algorithms: pitfalls, challenges and recent developments. *Physiological Measurement*, 25(1):125, Feb. 2004.

[75] W. R. B. Lionheart. Geometric methods for anisotopic inverse boundary value problems. In K. Bingham, Y. V. Kurylev, and E. Somersalo, editors, *New Analytic and Geometric Methods in Inverse Problems*, pages 337–351. Springer Berlin Heidelberg, Jan. 2004.

[76] M. H. Loke and R. D. Barker. Least-squares deconvolution of apparent resistivity pseudosections. *Geophysics*, 60(6):1682–1690, Dec. 1995.

[77] MATLAB and Optimization Toolbox R2011a. The MathWorks, Inc., Natick, Massachusetts, United States.

[78] K. Menger. Untersuchungen ber allgemeine metrik. *Mathematische Annalen*, 100(1):75–163, Dec. 1928.

[79] K. Menger. New foundation of euclidean geometry. *American Journal of Mathematics*, 53(4):721–745, Oct. 1931.

[80] P. Monk. *Finite Element Methods for Maxwell's Equations*. Clarendon Press, Apr. 2003.

[81] A. I. Nachman. Global uniqueness for a two-dimensional inverse boundary value problem. *The Annals of Mathematics*, 143(1):71, Jan. 1996.

[82] A. Panchenko, L. Päivärinta, and G. Uhlmann. Complex geometrical optics solutions for lipschitz conductivities. *Revista Matemtica Iberoamericana*, 19(1):57–72, 2003.

[83] K. Paridis. *The inverse conductivity problem: Anisotropy, finite elements and resistor networks*. PhD thesis, University of Manchester, 2013.

[84] A. Romsauerova, A. McEwan, L. Horesh, R. Yerworth, R. H. Bayford, and D. S. Holder. Multi-frequency electrical impedance tomography (EIT) of the adult human head: initial findings in brain tumours, arteriovenous malformations and chronic stroke, development of an analysis method and calibration. *Physiological Measurement*, 27(5):S147, May 2006.

[85] J. Schberl. NETGEN an advancing front 2d/3d-mesh generator based on abstract rules. *Computing and Visualization in Science*, 1(1):41–52, July 1997.

[86] P. P. Silvester and R. L. Ferrari. *Finite Elements for Electrical Engineers*. Cambridge University Press, Sept. 1996.

[87] J. Sylvester and G. Uhlmann. A uniqueness theorem for an inverse boundary value problem in electrical prospection. *Communications on Pure and Applied Mathematics*, 39(1):91–112, Jan. 1986.

[88] J. Sylvester and G. Uhlmann. A global uniqueness theorem for an inverse boundary value problem. *The Annals of Mathematics*, 125(1):153–169, 1987.

[89] J. Sylvester and G. Uhlmann. Inverse boundary value problems at the boundary-continuous dependence. *Communications on Pure and Applied Mathematics*, 41(2):197–219, Mar. 1988.

[90] M. E. Taylor. *Partial Differential Equations I*. Number 115 in Applied Mathematical Sciences. Springer New York, Jan. 2011.

[91] A. Trautman. Deformations of the hodge map and optical geometry. *Journal of Geometry and Physics*, 1(2):85–95, 1984.

[92] G. Uhlmann. Electrical impedance tomography and calderón's problem. *Inverse Problems*, 25(12):123011, Dec. 2009.

[93] G. Uhlmann and A. Bukhgeim. Recovering a potential from partial cauchy data. *Communications in Partial Differential Equations*, 27(3-4):653–668, Jan. 2002.

[94] E. VanderZee, A. N. Hirani, D. Guoy, and E. Ramos. Well-centered triangulation. *SIAM Journal on Scientific Computing*, 31(6):4497–4523, Jan. 2010. arXiv: 0802.2108.

[95] Y. Wan, R. Halter, A. Borsic, P. Manwaring, A. Hartov, and K. Paulsen. Sensitivity study of an ultrasound coupled transrectal electrical impedance tomography system for prostate imaging. *Physiological Measurement*, 31(8):S17, Aug. 2010.

[96] P. Wilkinson, P. Meldrum, O. Kuras, J. Chambers, S. Holyoake, and R. Ogilvy. High-resolution electrical resistivity tomography monitoring of a tracer test in a confined aquifer. *Journal of Applied Geophysics*, 70(4):268–276, Apr. 2010.

[97] Y. Yemini. The positioning problem - a draft of an intermediate summary. Technical report, Dec. 1978.

Appendix A

Triangulation Problem Code

```
% dihedral_survey_solve.m

clear
%% Load netgen mesh
meshdir = './NetgenMeshes/';
meshfile = 'sphere_very_coarse';
[~,H, V] = mesh_reader([meshdir,meshfile,'.vol']);

%% Add a few more interior verts and remesh
V_extra = [-0.45,0.1,0.08; 0.46,0.04,-0.1; 0.03,-0.44,0.04; ...
    0.02,0.42,0.09;0.01,-0.02,-0.46; -0.04,0.06,0.43];
V = [V; V_extra];
H = delaunay(V);

get_mesh_parameters

%% Plot:
figure(1);clf; tetramesh(H,V,'FaceAlpha',0); axis off

%% Go through tree and solve constraints then do survey solve by
% converting dihedrals to face angles
tree = grMinSpanTree(D);
rt = 1;
Dm=treesort(D(tree,:),rt); % easier if a directed tree.
```

```
th0 = th;
options = optimset('Display','iter');
%% Perturb dihedrals and solve survey problem
pp = 1:9;
ep = [1e-2,1e-1];

for p = kron(ep,pp)
Vnew = [];
cnt = 0;
while isempty(Vnew) && cnt<5
cnt = cnt+1;
fprintf('Perturbation: %.0e, Attempt: %i\n',p,cnt);
thp = th + p*randn(size(th));
[thnew,dep_idx,TF,J,remove_match] = ...
    apply_dihedral_constraints(H,F,E,find(~isbnde),Dm,thp,th0,options);

%% Recalculate face angles
aa = all_dihedral2face_angles(thnew,H,F,TF);
anew = zeros(nf,3);
for f=1:nf
    [i,j] = find(TF==f);
    anew(f,:) = aa(i(1),(j(1)-1)*3+(1:3));
end

%% Survey solve
facein = get_first_face(F,H,Eb,V);
fh=2; figure(fh); clf(fh); figure(fh);
Vnew = survey_solve3d(H,anew,facein,V(facein(1:2),:),V(facein(3),3),fh);
figure(fh); axis equal, axis off
end
if cnt==5 && isempty(Vnew), break, end

%% Plot svd of variable vertex 2 independent dihedral Jacobian
vi = 1:3*nv;
vi([(facein(1)-1)*3+(1:3),(facein(2)-1)*3+(1:3),(facein(3)-1)*3+1]) = [];
% New
Jhh = vert2dihedral_jacobian(Vnew,H);
Jh = Jhh(reshape(~dep_idx',[],1),vi);
figure(4); semilogy(svd(Jh),'o');
```

```
xlabel('i'); ylabel('\sigma_i')
% Independent to dependent Jacobian (from fsolve)
figure(3); clf; semilogy(svd(J),'o');
xlabel('i'); ylabel('\sigma_i')

end
```

```
% get_mesh_parameters.m

%% Find dual, etc
nt = size(H,1); nv = size(V,1);
H = orientH3d(H,V);
[E,F,D,Centroids,FC,FD]=meshtograph3d(H,V);
nf = size(F,1); nf0 = size(D,1); nfb = nf-nf0; ne = size(E,1);

%% Compute face angles
a = calc_face_angles(V,F);

%% Compute dihedral angles for each tet
th = zeros(nt,6);
for k=1:nt
    th(k,:) = calc_dihedrals(V(H(k,:),:));
end

%% Find interior edges
isbnde = find_interior_edges(E,H,th);
neb = sum(isbnde); ne0=ne-neb;
E0 = E(~isbnde,:);
Eb = E(isbnde,:);
e1 = find(all(E==Eb(1,1) | E==Eb(1,2),2));

%% Find interior verts:
vlist = 1:nv;
vb_idx = ismember(vlist,Eb(:)); vb = vlist(vb_idx);
vi = vlist(~vb_idx);
nvi = length(vi); nvb = length(vb);
```

```
%% Find interior faces
Fi = get_interior_faces(F,H);
nf0 = size(Fi,1);

%% Edge lengths:
l = calc_edge_lengths(E,V);
```

```
function [E,F,D,C,FC,FD]=meshtograph3d(H,V)

% [E,F,D,C,FC,FD]=meshtograph3d(H,V)
%
% Return the graph from a mesh.
%
% Input:
%   H - topology matrix
%   V - geometry (vertex positions)
%
% Output:
%   E - edges (1-faces of simplicial complex)
%   F - faces of tets (2-faces of simplicial complex)
%   D - edges of dual mesh
%   C - tetrahedron centroids (for plotting dual)
%   FC - Face centroids
%   FD - Dual of 2-skeleton
%
% Modification of Al Humaidi's code.

%% Edges and faces
nt = size(H,1);
E=[]; F=[];
% This will produce interior edges multiple times
e_idx = flipud(combnk(1:4,2)); nei = size(e_idx);
f_idx = flipud(combnk(1:4,3)); nfi = size(f_idx);
for it=1:nt
    E=[E;reshape(H(it,e_idx),nei)];
    F=[F;reshape(H(it,f_idx),nfi)];
end
```

```
% Sorting makes it easier to spot duplicates
E=sort(E,2); F=sort(F,2);
%Remove duplicates
E=unique(E,'rows'); F=unique(F,'rows');

%% Dual and centroids
if nargout>2

nf=size(F,1);
D=[];
% Find the faces in two tets
for i=1:nf
    [ts1,j] = find(H==F(i,1));
    [ts2,j] = find(H==F(i,2));
    [ts3,j] = find(H==F(i,3));
    ts = intersect(ts1',ts2');
    ts = intersect(ts,ts3');
    if length(ts)==2
        D = [D;ts];
    end
end

if nargout>3 && nargin==2
    % and centroid for plotting
    C = zeros(nt,3);
    for t=1:nt
        C(t,:) = mean(V(H(t,:),:));
    end

    % Face centroids
    if nargout>4
        FC = zeros(nf,3);
        for t=1:nf
            FC(t,:) = mean(V(F(t,:),:));
        end
    end

    % Two faces share an edge then they are in dual of 2-skeleton
    if nargout>5
```

```
        FD = [];
        ne = size(E,1);
        % For each edge, find faces that share it
        for e=1:ne
            faces = find(sum(F==E(e,1) | F==E(e,2),2)==2);
            FD = [FD; combnk(faces,2)];
        end
    end

end  % nargout>3

end  % nargout>2
```

```
function Hn = orientH3d(H,V)

nt=size(H,1);

for it = 1:nt
    v1 = V(H(it,1),:);
    v2 = V(H(it,2),:);
    v3 = V(H(it,3),:);
    v4 = V(H(it,4),:);
    s = det( [v2-v1;v3-v1;v4-v1]);
    if s<0
        H(it,:)=H(it,[2,1,3,4]);
    end
end
Hn=H;
```

```
function a = calc_face_angles(V,F)

% a = calc_face_angles(V,F)
%
% Calculate face angles a from vertex positions V.

nf = size(F,1);
```

```
a = zeros(nf,3);
for k=1:nf
    v1 = V(F(k,1),:); v2 = V(F(k,2),:); v3 = V(F(k,3),:);
    a(k,1) = acos( (v3-v1)*(v2-v1)'/(norm(v3-v1)*norm(v2-v1)) );
    a(k,2) = acos( (v1-v2)*(v3-v2)'/(norm(v1-v2)*norm(v3-v2)) );
    a(k,3) = acos( (v2-v3)*(v1-v3)'/(norm(v2-v3)*norm(v1-v3)) );
end
```

```
function th = calc_dihedrals(V)

% th = calc_dihedrals(V)
%
% Calculate dihedral angles for a tetrahedron.
% th(i) is angle at ith edge in tet using following convention
%   1: edge (1,2)
%   2: edge (1,3)
%   3: edge (1,4)
%   4: edge (2,3)
%   5: edge (2,4)
%   6: edge (3,4)

th = zeros(1,6);
v1 = V(1,:); v2 = V(2,:); v3 = V(3,:); v4 = V(4,:);
% Outer normals if tet is positively oriented
n1 = cross( v3-v2, v4-v2 ); n1=n1/norm(n1);
n2 = cross( v1-v3, v4-v3 ); n2=n2/norm(n2);
n3 = cross( v1-v4, v2-v4 ); n3=n3/norm(n3);
n4 = cross( v3-v1, v2-v1 ); n4=n4/norm(n4);

th(1) = acos(-n3*n4'); th(2) = acos(-n2*n4'); th(3) = acos(-n3*n2');
th(4) = acos(-n4*n1'); th(5) = acos(-n1*n3'); th(6) = acos(-n1*n2');
```

```
function isbnde = find_interior_edges(E,H,th)

% isbnde = find_interior_edges(E,H,th)
%
```

```
% Find the interior edges of a mesh using dihedral angles.

ne = size(E,1);
isbnde = false(1,ne);
edge_order = flipud(combnk(1:4,2));
for k=1:ne
    % find tets for which edge belongs
    tets = find(sum( H==E(k,1) | H==E(k,2),2 )==2);
    ang=0;
    % Find edge verts in each tet
    for l=tets'
        v1 = find(H(l,:)==E(k,1));
        v2 = find(H(l,:)==E(k,2));
        ang = ang+th(l,all( edge_order==v1 | edge_order==v2, 2 ));
    end
    isbnde(k) = (2*pi-ang > 1e-12);
end
```

```
function Fi = get_interior_faces(F,H,TF)

if nargin<3
    TF = find_all_tet_faces(H,F);
end

nf = size(F,1);
Fi = zeros(nf,1);
nfi = 0;
for f=1:nf
    [i,~] = find(TF==f);
    if length(i)==1, continue, end
    nfi=nfi+1;
    Fi(nfi) = f;
end
Fi(nfi+1:end) = [];
```

```
function TF = find_all_tet_faces(H,F)
```

```
% TF = find_all_tet_faces(H,F)

nt = size(H,1);
TF = zeros(nt,4);
for t=1:nt
    f_combo = combnk(H(t,:),3);
    for k=1:4
        TF(t,k) = find(all(F==f_combo(k,1) | F==f_combo(k,2) | ...
            F==f_combo(k,3),2));
    end
end
```

```
function l = calc_edge_lengths(edges,v)

l = sqrt(sum( (v(edges(:,1),:) - v(edges(:,2),:)).^2 , 2) );
```

```
function [thnew,dep_idx,TF,J,remove_match,exitflag] = ...
    apply_dihedral_constraints(H,F,E,E0,D,th,th0,options,varargin)

%% Initialise some variables
if nargin<7
    options = optimset('Display','iter');
end
nf = size(F,1);
nt = size(H,1);
nv = max(max(H));
TF = find_all_tet_faces(H,F);
Fi = get_interior_faces(F,H,TF);
Fb = ~ismember(1:nf,Fi);
Vb = unique(F(Fb,:));
Vi = find(~ismember(1:nv,Vb));
fk_idx = false(nf,1);   % has face been visited already?
vk_idx = false(nv,1);
rt_idx = [];
dep_idx = false(nt,6);   % list of dependent angles
```

```
tet_type = zeros(nt,1);

%% First face only has Cayley-Menger constraint
rt = D(1);
dep_idx(rt,6) = true;     % arbitrarily choose last edge as dependent
vk_idx(H(rt,:)) = true;
fk_idx(TF(rt,:)) = true;

%% Compute other constraint
ne = size(E,1);
curva_idx = cell(1,ne);
curvt_idx = cell(1,ne);
face_matched = false(nf,1);
facein = intersect(TF(rt,:),TF(D(1,2),:));
opp_idx1 = zeros(nf,3); opp_idx2 = zeros(nf,3);
th_idxa1 = zeros(nf,3); th_idxa2 = zeros(nf,3);
th_idxb1 = zeros(nf,3); th_idxb2 = zeros(nf,3);
tet_idx1 = zeros(nf,1); tet_idx2 = zeros(nf,1);
remove_match = false(nf,3);
v_removed = false(nv,1);
tet_removed = false(nt,1);

%% Get dependent variables
[dep_idx,tet_type] = trav_tree_dep_var(H,F,D,TF,rt,rt,facein,...
    fk_idx,vk_idx,dep_idx,tet_type);

%% Check the independent variables are within the required range:
if any(any(th(~dep_idx)<=0 | th(~dep_idx)>=pi))
    warning('Some of the independent angles are inconsistent.');
    thnew = [];
    J = [];
    return
end

%% Find constraint indices
fk_idx = false(nf,1); fk_idx(TF(rt,:)) = true;
[curva_idx,curvt_idx,opp_idx1,opp_idx2,th_idxa1,th_idxa2,...
    th_idxb1,th_idxb2,tet_idx1,tet_idx2,remove_match,...
    face_matched,fk_idx,rt_idx,v_removed,tet_removed] = ...
```

```
    trav_tree_setup_const(H,F,Fi,Vi,E,E0,TF,D,rt,tet_type,...
    curva_idx,curvt_idx,opp_idx1,opp_idx2,th_idxa1,th_idxa2,th_idxb1,...
    th_idxb2,tet_idx1,tet_idx2,fk_idx,rt_idx,...
    remove_match,face_matched,v_removed,tet_removed,false,varargin{:});

%% Remove one interior angle from each face:
for f = Fi'
    v = 1:3;
    if sum(remove_match(f,:),2)==2, continue, end
    for i = v
        if remove_match(f,i), continue, end
%         if ~ismember(F(f,i),
        remove_match(f,i) = true;
        break   % only remove one for each face
    end
end

%% Apply constraints
fun = @(TH) apply_survey_constraints(TH,th,Fi,dep_idx,...
    curva_idx,curvt_idx,opp_idx1,opp_idx2,th_idxa1,th_idxa2,...
    th_idxb1,th_idxb2,tet_idx1,tet_idx2,remove_match);
if size(th0,2)>1
    th0 = th0(dep_idx);
end
fprintf('Running fsolve...');
[th_dep,~,exitflag,~,J] = fsolve(fun,th0,options);
cnt = 0;
while exitflag>1 && cnt<5
    fprintf('Trying again from last point...');
    [th_dep,~,exitflag,~,J] = fsolve(fun,th_dep,options);
    cnt = cnt+1;
end
thnew = th;
thnew(dep_idx) = th_dep;
```

```
function [dep_idx,tet_type,fk_idx,vk_idx] = trav_tree_dep_var(H,F,D,...
    TF,rt,rrt,facein,fk_idx,vk_idx,dep_idx,tet_type,varargin)
```

```
% Traverse tree to get dependent variables

num_known = sum(fk_idx(TF(rt,:)));
v_known = sum(vk_idx(H(rt,:)));
if num_known==1 && v_known==3
    tet_type(rt) = 1;

    % 3 dependents are the 3 edges not on facein
    FF = F(facein,:);
    e_idx = flipud(combnk(H(rt,:),2));
    V_diff = setdiff(H(rt,:),FF);
    dep_idx(rt,any(e_idx==V_diff,2)) = true;

elseif num_known==1 && v_known==4
    tet_type(rt) = 2;
    % 6 dependents are all edges
    dep_idx(rt,:) = true;

elseif num_known==2
    tet_type(rt) = 3;
    % 6 dependents are all edges
    dep_idx(rt,:) = true;

elseif num_known==3
    tet_type(rt) = 4;
    % 6 dependents are all edges
    dep_idx(rt,:) = true;

elseif num_known==4 && rt~=rrt
    tet_type(rt) = 5;
    % 6 dependents are all edges
    dep_idx(rt,:) = true;

end
fk_idx(TF(rt,:)) = true; vk_idx(H(rt,:)) = true;

% Get the next tet
r=find(D(:,1)==rt);
```

```
for ir =r'
    nexttet=D(ir,2); % The next tet
    nextface=intersect(TF(rt,:),TF(nexttet, :));
    [dep_idx,tet_type,fk_idx,vk_idx] = trav_tree_dep_var(H,F,D,TF,...
        nexttet,rrt,nextface,fk_idx,vk_idx,dep_idx,tet_type,varargin{:});
end
```

```
function [curva_idx,curvt_idx,opp_idx1,opp_idx2,th_idxa1,th_idxa2,...
    th_idxb1,th_idxb2,tet_idx1,tet_idx2,remove_match,...
    face_matched,fk_idx,rt_idx,v_removed,tet_removed] = ...
    trav_tree_setup_const(H,F,Fi,Vi,E,E0,TF,D,rt,tet_type,...
    curva_idx,curvt_idx,opp_idx1,opp_idx2,th_idxa1,th_idxa2,...
    th_idxb1,th_idxb2,tet_idx1,tet_idx2,fk_idx,rt_idx,remove_match,...
    face_matched,v_removed,tet_removed,edge_cond_flag,varargin)

% Traverse tree to setup constraints

% Add all interior faces of this tet to match list:
faces = TF(rt,:);
faces(face_matched(faces)) = [];   % Remove faces already done
faces(~ismember(faces,Fi)) = [];    % Remove boundary faces
for f=faces
    [rtf_idx,stf_idx] = find_face_match_idx(TF,H,F,f,1:3);
    opp_idx1(f,:) = rtf_idx(:,1)';
    opp_idx2(f,:) = stf_idx(:,1)';
    th_idxa1(f,:) = rtf_idx(:,2)';
    th_idxa2(f,:) = stf_idx(:,2)';
    th_idxb1(f,:) = rtf_idx(:,3)';
    th_idxb2(f,:) = stf_idx(:,3)';
    tet_idx1(f) = rtf_idx(1,4);
    tet_idx2(f) = stf_idx(1,4);
end
face_matched(faces) = true;
% rt_idx(rt) = true;
rt_idx = [rt_idx,rt];

if tet_type(rt)==3      % 2 faces known
```

```
    % 1st constraint is curvature so find edge shared by both known faces
    [curva_idx,curvt_idx] = curv_cond(H,F,E,E0,TF,...
        rt,fk_idx,rt_idx,curva_idx,curvt_idx,2);

elseif tet_type(rt)==4 || tet_type(rt)==5   % 3 or 4 faces known
    % curvature constraints which all tets surrounding have been visited
    [curva_idx,curvt_idx] = curv_cond(H,F,E,E0,TF,...
        rt,fk_idx,rt_idx,curva_idx,curvt_idx,tet_type(rt)-1);

    % Remove interior angle if tet encloses an interior vertex
    V = H(rt,:);
    V(~ismember(V,Vi)) = [];    % only interior vertices
    V(v_removed(V)) = [];
    vlist = [];
    for v=V
        [t,~] = find(H==v);     % Find the tets this vertex is in
%         if all(rt_idx(t))       % have we visited all these tets?
        if all(ismember(t,rt_idx))
            vlist = [vlist,v];
        end
    end
    if ~isempty(vlist)
        ntets = zeros(size(vlist));
        for v=vlist
            [t,~] = find(H==v); t(t==rt) = [];
            ntets(vlist==v) = sum(~tet_removed(t));
        end
        [~,ti] = sort(ntets);
        vlist = vlist(ti);
        for v=vlist
            [tets,~] = find(H==v);
            tets(tet_removed(tets)) = [];
            % reverse order we visited the tets:
            tets = fliplr(rt_idx(ismember(rt_idx,tets)));
            for t = tets
                faces = TF(t,:);
                [ii,~] = find(F(faces,:)==v);
                if any(any(remove_match(faces(ii),:))), continue, end
                EE = find_edges3d(H(t,:),E);    % edges in this tet
```

```
                for e=EE
                    if isempty(curva_idx{e}), EE(EE==e) = []; end
                end
                EE = EE(any(E(EE,:)==v,2));
                if length(EE)<3, continue, end
                % Remove one of these edges
                reidx = EE(1);
                curva_idx{reidx} = [];
                curvt_idx{reidx} = [];
                % Faces sharing removed edge:
                i = sum(F(faces,:)==E(reidx,1) | ...
                            F(faces,:)==E(reidx,2),2)==2;
                faces = faces(i);
                if ~edge_cond_flag
                    % Remove angles on these faces at interior vertex
                    [i,j] = find(F(faces,:)==v);
                    for k=1:length(i)
                        remove_match(faces(i(k)),j(k)) = true;
                    end
                    % Remove angles not on removed edge
                    [i,j] = find(F(faces,:)~=E(reidx,1) & ...
                                    F(faces,:)~=E(reidx,2));
                    for k=1:length(i)
                        remove_match(faces(i(k)),j(k)) = true;
                    end
                end
                % Other face containing interior vertex:
                face = TF(t,:);
                face(all(F(face,:)~=v,2)) = [];
                face(sum(F(face,:)==E(reidx,1) | ...
                            F(face,:)==E(reidx,2),2)==2) = [];
                remove_match(face,F(face,:)~=v) = true;
                tet_removed(t) = true;
                v_removed(v) = true;
                break
            end
        end
    end
end
```

```
fk_idx(TF(rt,:)) = true;

% Get the next tet
r=find(D(:,1)==rt);
for ir =r'
    nexttet=D(ir,2); % The next tet
    if ~isempty(varargin)
        figure(varargin{1}); hold on;
        C = varargin{2};
        plot3(C(D(ir,:),1),C(D(ir,:),2),C(D(ir,:),3),'LineWidth',1.5);
    end
    [curva_idx,curvt_idx,opp_idx1,opp_idx2,th_idxa1,th_idxa2,...
    th_idxb1,th_idxb2,tet_idx1,tet_idx2,remove_match,face_matched,...
    fk_idx,rt_idx,v_removed,tet_removed] = ...
    trav_tree_setup_const(H,F,Fi,Vi,E,E0,TF,D,nexttet,tet_type,...
    curva_idx,curvt_idx,opp_idx1,opp_idx2,th_idxa1,th_idxa2,...
    th_idxb1,th_idxb2,tet_idx1,tet_idx2,fk_idx,rt_idx,remove_match,...
    face_matched,v_removed,tet_removed,edge_cond_flag,varargin{:});
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [curva_idx,curvt_idx,curv_used,opp_edge] = ...
    curv_cond(H,F,E,E0,TF,rt,fk_idx,rt_idx,curva_idx,curvt_idx,nface)

faces = TF(rt,fk_idx(TF(rt,:)));
f_idx = combnk(1:nface,2);
if nface==4, f_idx = flipud(f_idx); end
ncurv = size(f_idx,1);
curv_cnt=0;
curv_used = zeros(1,ncurv);
opp_edge = zeros(1,ncurv);
for c=1:ncurv   % max number of unique interior edges
    se = intersect(F(faces(f_idx(c,1)),:),F(faces(f_idx(c,2)),:));
    e_idx = all(E==se(1) | E==se(2),2);
    if ~ismember(find(e_idx),E0), continue, end   % skip boundary edges
    [cidx,tidx] = find_curv_idx(H,se,rt_idx);
    if isempty(cidx), continue, end
    curva_idx{e_idx} = cidx;
    curvt_idx{e_idx} = tidx;
```

```
    curv_cnt=curv_cnt+1;
    curv_used(curv_cnt) = find(e_idx);  % Keep track of edges used here
    oppv = H(rt,H(rt,:)~=E(e_idx,1) & H(rt,:)~=E(e_idx,2));
    opp_edge(curv_cnt) = find(all(E==oppv(1) | E==oppv(2),2));
end
curv_used(curv_used==0) = [];
opp_edge(opp_edge==0) = [];
```

```
function fun = apply_survey_constraints(TH,th,Fi,dep_idx,...
    curva_idx,curvt_idx,opp_idx1,opp_idx2,th_idxa1,th_idxa2,...
    th_idxb1,th_idxb2,tet_idx1,tet_idx2,remove_match,flag)

th(dep_idx) = TH;
fun = [];

% All tets have Cayley-Menger constraint
% if nargin<16 || ~flag
    fun = spherical_cayley_menger(th);
% end

for e=1:length(curva_idx)
    if isempty(curva_idx{e}), continue, end
    curv_sum = curv_constraint(th,curva_idx{e},curvt_idx{e});
    fun = [fun; abs(2*pi - curv_sum)];
end

% Others are the shared face angle matchings
for f=1:size(opp_idx1,1);
    if ~ismember(f,Fi), continue, end
    for v=1:3
        if remove_match(f,v),continue,end
%         if opp_idx1(f,v)==0, continue, end
        a_rt = dihedrals2face(th(tet_idx1(f),...
            [opp_idx1(f,v),th_idxa1(f,v),th_idxb1(f,v)]));
        as = dihedrals2face(th(tet_idx2(f),...
            [opp_idx2(f,v),th_idxa2(f,v),th_idxb2(f,v)]));
        fun = [fun; abs(as-a_rt)];
```

```
    end
end
```

```
function dcm = spherical_cayley_menger(th)

% dcm = spherical_cayley_menger(th)

nt = size(th,1);
dcm = zeros(nt,1);
for k=1:nt
    CM = [-1, cos(th(k,[6,5,4])); ...
          cos(th(k,6)),-1,cos(th(k,[3,2]));...
          cos(th(k,[5,3])),-1,cos(th(k,1));...
          cos(th(k,[4,2,1])),-1];
    dcm(k) = det(CM);
end
```

```
function curv_sum = curv_constraint(TH,cidx,tidx)

% Find all tets for which this is an edge
lt = length(tidx);
curv_sum = 0;
for t=1:lt
    curv_sum = curv_sum+TH(tidx(t),cidx(t));
end
```

```
function a = dihedrals2face(th)

a = acos((cos(th(1)) + cos(th(2))*cos(th(3)))./(sin(th(2))*sin(th(3))));
```

```
function edge_idx = find_edges3d(elem,edges)

% Find the edges that belong to elem.
```

```
edge_idx(1) = find( all(edges==elem(1) | edges==elem(2) , 2) );
edge_idx(2) = find( all(edges==elem(1) | edges==elem(3) , 2) );
edge_idx(3) = find( all(edges==elem(1) | edges==elem(4) , 2) );
edge_idx(4) = find( all(edges==elem(2) | edges==elem(3) , 2) );
edge_idx(5) = find( all(edges==elem(2) | edges==elem(4) , 2) );
edge_idx(6) = find( all(edges==elem(3) | edges==elem(4) , 2) );
```

```
function facein = get_first_face(F,H,Eb,V)

ff = find( sum(F==Eb(1,1) | F==Eb(1,2),2)==2);
for i=1:length(ff)
    f1 = ff(i);
    facein = F(f1,:);
    tetin = sum( H==facein(1) | H==facein(2) | H==facein(3),2)==3;
    if sum(tetin)==1, break, end
    if i==length(ff), error('No boundary faces for this edge.'); end
end
fo = setdiff(H(tetin,:),facein);
if isinorder3d(V(facein,:),V(fo,:))
    facein=facein([2,1,3]);
end
```

```
function isit=isinorder3d(face_v,opp_v)

% isit=isinorder3d(face_v,opp_v)
%
% Test orientation of next face w.r.t node opposite.

v1 = face_v(1,:);
v2 = face_v(2,:);
v3 = face_v(3,:);
isit = det( [v1-opp_v;v2-opp_v;v3-opp_v])>0;
```

```
function V = survey_solve3d(H,a,f1,v,z,fh,F,D,tree)
```

```
% V = survey_solve3d(H,a,f1,v,fh)
%
% Extension of Al Humaidi survey solve code to 3D.
% Inputs:
%   H - element topology matrix [n_elements x 4]
%   a - face angles for each face [n_face x 3]
%   f1 - vertex numbers of first face [1 x 3] (Note: This face must be
%        ordered clockwise from the outside of the tet.
%   v - coordinates of first edge [2 x 3]
%   x - x-coordinate of 3rd vertex of first face
%   fh - (optional) figure handle for plotting (if not given then no
%        plotting, if gcf then use current window
%
% Output:
%   V - coordinates of all vertices in mesh [n_v x 3]
%

global newV cnt

% Get edges, faces, dual graph and min spanning tree of dual
if nargin<6
    fh = [];
end
if nargin<7
    [~,F,D]=meshtograph3d(H);
end
if nargin<9
    tree = grMinSpanTree(D);
end

% Edges of min spanning tree of dual graph
Dm=D(tree,:);
% The root of the tree is face f1, find this in face in F
r = find(all( (F==f1(1) | F==f1(2)) | F==f1(3) , 2 ));
if length(r)~=1
    error('Not found face [%d, %d, %d]',f1);
end
```

```
% Which tet is this face in?
root_tet = find(sum( (H==f1(1) | H==f1(2)) | H==f1(3) , 2 )==3);
Dm=treesort(Dm,root_tet); % easier if a directed tree.

% Compute the y and z position of 3rd vertex:
a1 = a(r,F(r,:)==f1(1));     % angle at first vertex
a2 = a(r,F(r,:)==f1(2));
a3 = a(r,F(r,:)==f1(3));
idx = combnk(1:3,2);
for k=1:3
    func = @(vv) third_vertex(vv,v(1,:),v(2,:),z,a1,a2,a3,idx(k,:));
    options = optimset('Display','notify');
    [v3,~,exitflag] = fsolve(func,[1,1],options);
    if exitflag==1, break, end
end
if exitflag~=1
    V = [];
    fprintf('3rd vertex could not be found.\n\n');
    return
end
v(3,:) = [v3,z];

nv = max(max(H));% number of vertices
newV= zeros(nv,3);
newV(f1(1),:)=v(1,:);
newV(f1(2),:)=v(2,:);
newV(f1(3),:)=v(3,:);

cnt=0;
if ~isempty(fh)
    figure(fh(1)); hold on;
    plot3(v([1:3,1],1),v([1:3,1],2),v([1:3,1],3),'o-'); view(3);
    pause(1);
    traverse_tets(H,Dm,F,a,root_tet,f1,v,fh)
else
    traverse_tets(H,Dm,F,a,root_tet,f1,v)
end
V=newV;
```

```
function f = third_vertex(v3,v1,v2,z,a1,a2,a3,idx)

v3 = [v3,z];
f = [cos(a1) - ( (v3-v1)*(v2-v1)' )/( norm(v3-v1)*norm(v2-v1) );...
    cos(a2) - ( (v3-v2)*(v1-v2)' )/( norm(v3-v2)*norm(v1-v2) );...
    cos(a3) - ( (v1-v3)*(v2-v3)' )/( norm(v1-v3)*norm(v2-v3) )];

f = f(idx);
```

```
function traverse_tets(H,Dm,F,a,troot,facein,v,varargin)

global newV cnt
cnt=cnt+1;

% we are in tet troot and entered it from facein
nextvertex = setdiff(H(troot,:),facein);
if length(nextvertex)~=1
    error('Next vertex not unique!');
end

% Find vertex numbers in the face opposite new vertex
fo4 = find( all(F==facein(1) | F==facein(2) | F==facein(3),2) );
id42 = F(fo4,:)==facein(2);

% Face opposite vertex 1:
fo1 = find( all(F==facein(2) | F==facein(3) | F==nextvertex,2) );
id12 = F(fo1,:)==facein(2);
id13 = F(fo1,:)==facein(3);
id14 = F(fo1,:)==nextvertex;

% Face opposite vertex 3:
fo3 = find( all(F==facein(1) | F==facein(2) | F==nextvertex,2) );
id32 = F(fo3,:)==facein(2);

newV(nextvertex,:) = find_missing_vertex(v(1,:),v(2,:),v(3,:),...
    a(fo1,id12),a(fo1,id13),a(fo1,id14),a(fo3,id32),a(fo4,id42));
```

```
if nargin>7
    figure(varargin{1}); hold on
    for k=1:3
        plot3(newV([facein(k),nextvertex],1),...
            newV([facein(k),nextvertex],2),...
            newV([facein(k),nextvertex],3),'-o'); view(3);
    end
    hold off
    pause(0.1)
end

%Get the next tet
[r,~]=find(Dm(:,1)==troot);
for ir =r'
    nexttet=Dm(ir,2); %The next tet
    nextface=intersect(H(troot,:),H(nexttet, :));
    nextv = zeros(3,3);
    nextv(1,:)= newV(nextface(1),:);
    nextv(2,:)= newV(nextface(2),:);
    nextv(3,:)= newV(nextface(3),:);
    oldv = setdiff(H(troot,:),nextface);
    if ~isinorder3d(newV(nextface,:),newV(oldv,:))
        nextface=nextface([2,1,3]);
        nextv=nextv([2,1,3],:);
    end
    traverse_tets(H,Dm,F,a,nexttet,nextface,nextv,varargin{:});
end
```

```
function v4 = find_missing_vertex(v1,v2,v3,a12,a13,a14,a32,a42)

% v4 = find_missing_vertex(v1,v2,v3,a12,a13,a14,a32,a42)
%
% Computes the position of the 4th vertex of a tetrahedron given the face
% angles aij where aij is the angle for the face opposite vertex i and
% at vertex j.

v12 = v1-v2;
```

```
v23 = v3-v2; l23 = norm(v23);
l24 = l23*sin(a13)/sin(a14); % distance along an edge to new vertex
th = acos(cos(a32)/(sin(a12)*sin(a42)) - cot(a12)*cot(a42));  % dihedral
n4 = -cross(v23,v12); n4 = n4/norm(n4); % Outward normal

v4 = rotate_vector3d(l24*v23/l23,n4,a12)+v2;
v4 = rotate_vector3d((v4-v2),-v23/l23,pi-th)+v2;
```

```
function p = rotate_vector3d(v,u,a)

% p = rotate_vector3d(v,u,a)
%
% Rotate point v by angle a about the axis defined by the unit vector u.

p = cos(a)*v + sin(a)*cross(u,v) + (1-cos(a))*dot(u,v)*u;
if size(p,1)~=size(v,1)
    p = p';
end
```

Appendix B

Discrete EIT Code

```
% edgecondmap_nonuniqueness.m

%% Create simple 8-tet mesh:
H = [1,2,3,4; 1,2,4,5; 1,2,5,6; 1,2,6,3; ...
    7,2,3,4; 7,2,4,5; 7,2,5,6; 7,2,6,3];
th = linspace(0,2*pi,5); th(end)=[];
V = [0,0,1; 0,0,0; [cos(th(:)),sin(th(:)),zeros(4,1)]; 0,0,-1];
pert = true;
if pert
    V(2,:) = V(2,:)+0.1;
    p_str = '_pert';
else
    p_str = '';
end
get_mesh_parameters;

%% Random number stream setup
if ~exist('defaultStream','var')
    defaultStream = RandStream.getDefaultStream;
    savedState = defaultStream.State;
else
    defaultStream.State = savedState;
end

%% Define conduductivity type
```

```
% % Anisotropic
% cond_type.string = 'anisotropic';

% % Layers:
% cond_type.string = 'layered';
% % cond_type.normals = Centroids;    % radial
% % cond_type.normals = repmat([0,0,1],nt,1);
% r = 0.1*randn(nt,3);
% cond_type.normals = repmat([0,0,1],nt,1) + r;
%
% cond_type.normals = cond_type.normals./repmat(sqrt(sum(...
%   cond_type.normals.^2,2)),1,3);

% % Isotropic:
% cond_type.string = 'isotropic';

% Multiplicative constant:
cond_type.string = 'conformal';
cond_type.constraints = repmat([1,0,0,1,0,1]',nt,1);
r = 0.1*randn(6*nt,1);
cond_type.constraints = cond_type.constraints + r;  % random
constraint_str = '_random';

%% Get conductivity to edge conductance map
Q = cond2edgecondmap(H,E,V,cond_type);

%% Plot singular values
[QQ,RR] = qr(Q);
[UU,SS,VV] = svd(QQ*RR);
sing = diag(SS);
sing = sing./sing(1);   % scale on largest sing value

figure(1); clf; semilogy(sing,'o');
ax = axis; ax([3,4]) = [1e-18,1]; axis(ax);

Z = null(Q,'r');
Z = Z./repmat(max(abs(Z),[],1),size(Z,1),1);    % normalise
figure(3); clf; imagesc(Z); axis image; colorbar; % rational null
```

```
function [A,Z] = cond2edgecondmap(H,E,V,cond_type)

% [A,Z] = cond2edgecondmap(H,E,V,cond_type)
%
% Calculate the linear map between anisotropic conductivity s and the
% edge conductance. If Z is requested then the rational null space is
% also returned. n is the layer normal vector for each tet.

nt = size(H,1);
ne = size(E,1);
type_flag = 1;
A = zeros(ne,6*nt);
iv1=1; iv2=2;
if nargin>3
    switch cond_type.string
        case 'anisotropic'
            type_flag = 1;
        case 'layered'
            type_flag = 2;
            n = cond_type.normals;
            A = zeros(ne,2*nt);
        case 'isotropic'
            type_flag = 3;
            A = zeros(ne,nt);
        case 'conformal'
            type_flag = 4;
            c = cond_type.constraints;
            A = zeros(ne,6*nt);
        otherwise
            error('%s not recognised as a possible conductivity type.'...
                ,cond_type.string);
    end
end

% Compute volume of each tet:
v1 = V(H(:,1),:); v2 = V(H(:,2),:); v3 = V(H(:,3),:); v4 = V(H(:,4),:);
vol = abs( sum( (v1-v4).*cross(v2-v4,v3-v4,2),2) )/6;
const = 36;
```

```
for e=1:ne
    [tets1,ni] = find(H==E(e,1));
    [tets2,nj] = find(H==E(e,2));
    [tets,ii,jj] = intersect(tets1,tets2);
    ni=ni(ii); nj=nj(jj);   % nodes in tet on this edge
    for tk = 1:length(tets)
        t = tets(tk);
        % Need to get opposite nodes in correct order
        no = find( H(t,:)~=H(t,ni(tk)) & H(t,:)~=H(t,nj(tk)) );
        i = H(t,ni(tk)); j = H(t,nj(tk));
        k = H(t,no(iv1)); l = H(t,no(iv2));
        vi=V(i,:); vj=V(j,:);
        vk=V(k,:); vl=V(l,:);
        if det([(vi-vl)',(vj-vl)',(vk-vl)']) < 0
            tmp = k;
            k = l;
            l = tmp;
            vk=V(k,:); vl=V(l,:);
        end
        c1 = cross(vk-vl,vj-vl);  c2 = cross(vi-vl,vk-vl);
        if type_flag==1 || type_flag==4
            A(e,(t-1)*6 + (1:6)) = [c1(1)*c2(1), ...
                c1(1)*c2(2)+c1(2)*c2(1), c1(1)*c2(3)+c1(3)*c2(1),...
                c1(2)*c2(2), c1(2)*c2(3)+c1(3)*c2(2), c1(3)*c2(3)]./...
                (const*vol(t));
        elseif type_flag==2
            idx = (t-1)*2;
            A(e,idx+1) = c1*(n(t,:)'*n(t,:))*c2';
            A(e,idx+2) = c1*c2' - A(e,idx+1);
            A(e,idx+(1:2)) = A(e,idx+(1:2))/(const*vol(t));
        elseif type_flag==3
            A(e,t) = c1*c2'/(const*vol(t));
        end
    end
end
A = -A;

if type_flag==4
```

```
    A = condmap2multconst(A,c);
end

if nargout>1
    Z = null(A,'r');
end
```

```
function B = condmap2multconst(A,s)

% B = condmap2multconst(A,s)
%
% Convert conductivity-to-edge-conductance map to constrained matrix for
% conductivity known up to multiplicative constant.

[ne,ntt] = size(A); nt = ntt/6;
for t=1:nt
    idx = (t-1)*6 + (1:6);
    B(:,t) = sum(A(:,idx).*repmat(s(idx)',ne,1),2);
end
```

```
% test_edgecond2vertices_dihedrals.m

clear

%% Load netgen mesh
meshdir = './NetgenMeshes/';
meshfile = 'sphere_very_coarse';
% meshfile = 'cylinder_v_coarse';
[~,H, V] = mesh_reader([meshdir,meshfile,'.vol']);

%% Add a few more interior verts and remesh
if strcmp(meshfile,'sphere_very_coarse');
    V_extra = [-0.45,0.1,0.08; 0.46,0.04,-0.1; 0.03,-0.44,0.04; ...
        0.02,0.42,0.09;0.01,-0.02,-0.46; -0.04,0.06,0.43];
    V = [V; V_extra];
    H = delaunay(V);
```

```
end

get_mesh_parameters

%% Tree:
tree = grMinSpanTree(D);
rt = 1;
Dm=treesort(D(tree,:),rt); % easier if a directed tree

%% Random number stream setup
defaultStream = RandStream.getDefaultStream;
load randstream;
defaultStream.State = savedState;

%% Define type of conductivity perturbation
% Anisotropic
sv = repmat([1,0,0,1,0,1],nv,1);
s = repmat([1,0,0,1,0,1],nt,1);    % tets
cond_type_test.string = 'anisotropic';

% % Layers:
% cond_type_test.string = 'layered';
% cond_type_test.normals = Centroids; % spherical
% % cond_type_test.normals = repmat([0,0,1],nt,1);
% s = repmat([1,1]',nt,1);
% sv = repmat([1,1],nv,1);

% % Isotropic:
% cond_type_test.string = 'isotropic';
% sv = ones(nv,1);
% s = ones(nt,1);

% % Multiplicative constant:
% cond_type_test.string = 'conformal';
% cond_type_test.constraints = repmat([1,0,0,1,0,1]',nt,1);
% sv = ones(nv,1);
% s = ones(nt,1);

pp = 1; ep = [1e-1];
```

```
%% Options
options = optimset(@fsolve);
options.Display = 'iter';
options.Algorithm = 'levenberg-marquardt';
options.fix_boundary = true;
options.fix_cond_boundary = false;
options.conductivity = 'tets';
options.Jacobian = 'on';

%% Run simulations
spp = ep*pp
fprintf('Perturbation: %.0e\n',spp);

while 1
    sp = s + spp*randn(size(s));
    flag = false;
    if strcmp(cond_type_test.string,'anisotropic')
        % Check pos def
        sss = unstack_conductivity(reshape(sp',[],1));
        for k=1:length(sss)
            if any(eig(sss{k})<=0), flag = true; break, end
        end
    else
        % Check both eigenvalues positive
        if any(any(sp<=0)), flag=true; end
    end
    if flag, continue, end
    break
end
sp = reshape(sp',[],1);
s0 = s;

%% Compute edge conductances
if isfield(cond_type_test,'constraints')
    r = 0.1*randn(6*nt,1);
    cond_type_test.constraints = cond_type_test.constraints+r;
end
Q = cond2edgecondmap(H,E,V,cond_type_test);
```

```
C = Q*sp;

%% Initial point:
v0 = V;
cond_type = cond_type_test;     % Assume a-priori knowledge is correct

%% Find mesh perturbation
[vnew,snew,J,exitflag,fval] = ...
    edgecond2vertices_dihedrals(H,F,E,find(~isbnde),Dm,v0,C,s0,...
    cond_type,options);

%% Errors
v_err = norm(V(:)-vnew(:))./norm(V(:))
s_err = norm(snew(:)-sp(:))./norm(sp(:))
```

```
function [vnew,snew,Jexitflag,fval] = ...
    edgecond2vertices_dihedrals(H,F,E,E0,D,v0,C,s0,cond_type,options)

%% Initialise some variables
nf = size(F,1);
nt = size(H,1);
nv = max(max(H));
TF = find_all_tet_faces(H,F);
Fi = get_interior_faces(F,H,TF);
Fb = ~ismember(1:nf,Fi);
Vb = unique(F(Fb,:));
Vi = find(~ismember(1:nv,Vb));

%% Apply edge conductance constraints
if isfield(options,'fix_boundary') && options.fix_boundary
    fix_boundary = true;
    options = rmfield(options,'fix_boundary');
    vb_pos = v0(Vb,:);
    v00 = reshape(v0(Vi,:)',[],1);
else
    fix_boundary = false;
    vb_pos = [];
```

```
    v00 = reshape(v0',[],1);
end
s00 = reshape(s0',[],1);
sb = [];
fix_cond_boundary = false;
cond_flag = false;
if isfield(options,'conductivity')&& strcmp(options.conductivity,'verts')
    if isfield(options,'fix_cond_boundary') && ...
            options.fix_cond_boundary
        fix_cond_boundary = true;
        options = rmfield(options,'fix_cond_boundary');
        sb = s0(Vb,:);
        s00 = reshape(s0(Vi,:)',[],1);
    end
    cond_flag = true;
    options = rmfield(options,'conductivity');
end

fun = @(x) apply_edgecond2vertices_dihedral_constraints(x,nv,ne,C,E,H,...
    Vb_idx,Vi,Vb,sb,cond_type,fix_boundary,fix_cond_boundary,cond_flag);

fprintf('Running fsolve...');
[x,fval,exitflag,~,J] = fsolve(fun,[v00;s00],options);
cnt = 0;
while exitflag>1 && cnt<5
    fprintf('Trying again from last point...');
    [x,fval,exitflag,~,J] = fsolve(fun,x,options);
    cnt = cnt+1;
end
if fix_boundary
    vnew = zeros(nv,3);
    vnew(Vb,:) = v0(Vb,:);
    n_vert_var = 3*length(Vi);
    vnew(Vi,:) = reshape(x(1:n_vert_var),3,[])';
else
    n_vert_var = 3*nv;
    vnew = reshape(x(1:n_vert_var),3,nv)';
end
if cond_flag
```

```
    if fix_cond_boundary
        snew = reshape(x(n_vert_var+1:end)',[],length(Vi))';
    else
        snew = reshape(x(n_vert_var+1:end)',[],nv)';
    end
else
    snew = reshape(x(n_vert_var+1:end)',[],nt)';
end
```

```
function [fun,J] = apply_edgecond2vertices_dihedral_constraints(x,nv,...
    ne,C,E,H,Vb_idx,Vi,Vb,sb,cond_type,fix_boundary,...
    fix_cond_boundary,cond_flag)

%% Vertex input
if fix_boundary
    V = zeros(nv,3);
    V(Vb_idx,:) = Vb;
    n_vert_var = 3*length(Vi);
    V(Vi,:) = reshape(x(1:n_vert_var),3,[])';
else
    n_vert_var = 3*nv;
    V = reshape(x(1:n_vert_var),3,nv)';
end

%% Conductivity input
if cond_flag
    if fix_cond_boundary
        % conductivity on interior vertices
        sv = zeros(length(Vi),size(sb,2));
        sv(Vi,:) = reshape(x(n_vert_var+1:end),[],length(Vi))';
        sv(Vb_idx,:) = sb;
    else
        sv = reshape(x(n_vert_var+1:end),[],nv)';
    end
    s = vertcond2tetcond(sv,H);
    s = reshape(s',[],1);
else
```

```
    % conductivities on tets
    s = x(n_vert_var+1:end);
end

%% Edge conductances
Q = cond2edgecondmap(H,E,V,cond_type);
Ce = Q*s;
eidx = 1:ne;
fun = Ce(eidx)-C(eidx);

%% Analytic Jacobian
if nargout>1
    J = zeros(ne,size(x,1));
    J(:,1:3*length(Vi)) = edgecond2vertices_analytic_jacobian(...
        H,E,V,Vi,s,cond_type);
    J(:,3*length(Vi)+1:end) = Q;
end
```

```
function J = edgecond2vertices_analytic_jacobian(H,E,V,vi,s0,cond_type)

%% Initialise arrays
nt = size(H,1);
ne = size(E,1);
nvi = length(vi);
J = zeros(ne,3*nvi);

%% Put in correct form for fully anisotropic
s = change_cond2anisotropic(cond_type,s0,nt);

%% Compute volume on each tet
v1 = V(H(:,1),:); v2 = V(H(:,2),:); v3 = V(H(:,3),:); v4 = V(H(:,4),:);
vol = abs( sum( (v1-v4).*cross(v2-v4,v3-v4,2),2) )/6;
const = 36;

%% For each edge differentiate wrt every vertex position
for e=1:ne
    % Reset dQ
```

```
dQ = zeros(3*nvi,6*nt);  % General anisotropic for now
% Q only non-zero on tets that edge is adjacent to
[tets1,ni] = find(H==E(e,1));
[tets2,nj] = find(H==E(e,2));
[tets,ii,jj] = intersect(tets1,tets2);
ni=ni(ii); nj=nj(jj);   % nodes in tet on this edge
for tk = 1:length(tets)
    t = tets(tk);
    % Need to get opposite nodes in correct order
    no = find( H(t,:)~=H(t,ni(tk)) & H(t,:)~=H(t,nj(tk)) );
    % These are local nodes for this edge on this tet
    i = H(t,ni(tk)); j = H(t,nj(tk));
    k = H(t,no(1)); l = H(t,no(2));
    % Which ones are interior vertices?
    i_idx = find(vi==i); j_idx = find(vi==j);
    k_idx = find(vi==k); l_idx = find(vi==l);
    xi=V(i,:); xj=V(j,:);
    xk=V(k,:); xl=V(l,:);
    % Test correct way around:
    if det([(xi-xl)',(xj-xl)',(xk-xl)']) < 0
        tmp = k;
        k = l;
        l = tmp;
        xk=V(k,:); xl=V(l,:);
        k_idx = find(vi==k); l_idx = find(vi==l);
    end
    % Calcluate ci's and cj's and their derivs:
    ci = cross(xk-xl,xj-xl);  cj = cross(xi-xl,xk-xl);
    dA = d_detA(xi,xj,xk,xl);
    dV = dA/(6*vol(t));     % extra scaling factor
    cstr = 'ijkl';
    % These local indices are only non-zeros in dQ
    for p=1:3
        dci = calc_dci(xj,xk,xl,p);
        dcj = calc_dcj(xi,xk,xl,p);
        for cc=1:length(cstr)
            idx = eval([cstr(cc),'_idx']);
            if ~isempty(idx)
                dQ((idx-1)*3+p,(t-1)*6+(1:6)) = ...
```

```
                        calc_dQ(ci,cj,dci,dcj,p,dV,cc);
                end
            end
        end
        % Divide by volume part:
        dQ(:,(t-1)*6+(1:6)) = -dQ(:,(t-1)*6+(1:6))/(const*vol(t));
    end
    J(e,:) = (dQ*s)';
end


%% Subfunctions
function dQ = calc_dQ(ci,cj,dci,dcj,p,dA,idx)
dQ = zeros(1,6);
% Diagonal components
dQ(1) = dci(1,idx)*cj(1) + ci(1)*dcj(1,idx) - ci(1)*cj(1)*dA(p,idx);
dQ(4) = dci(2,idx)*cj(2) + ci(2)*dcj(2,idx) - ci(2)*cj(2)*dA(p,idx);
dQ(6) = dci(3,idx)*cj(3) + ci(3)*dcj(3,idx) - ci(3)*cj(3)*dA(p,idx);
% Off-diagonals
dQ(2) = dci(1,idx)*cj(2) + ci(1)*dcj(2,idx) + ...
        dci(2,idx)*cj(1) + ci(2)*dcj(1,idx) - ...
        (ci(1)*cj(2) + ci(2)*cj(1))*dA(p,idx);
dQ(3) = dci(1,idx)*cj(3) + ci(1)*dcj(3,idx) + ...
        dci(3,idx)*cj(1) + ci(3)*dcj(1,idx) - ...
        (ci(1)*cj(3) + ci(3)*cj(1))*dA(p,idx);
dQ(5) = dci(2,idx)*cj(3) + ci(2)*dcj(3,idx) + ...
        dci(3,idx)*cj(2) + ci(3)*dcj(2,idx) - ...
        (ci(2)*cj(3) + ci(3)*cj(2))*dA(p,idx);

function dci = calc_dci(xj,xk,xl,p)
dci = zeros(3,4);
for q=1:3
    dci(q,2) = delta_perm(xk,xl,p,q);   % dci_q/dxj_p
    dci(q,3) = delta_perm(xl,xj,p,q);   % dci_q/dxk_p
    dci(q,4) = delta_perm(xj-xl,xk-xl,p,q);   % dci_q/dxl_p
end

function dcj = calc_dcj(xi,xk,xl,p)
dcj = zeros(3,4);   % d x n_v
```

```
for q=1:3
    dcj(q,1) = delta_perm(xl,xk,p,q);   % dcj_q/dxi_p
    dcj(q,3) = delta_perm(xi,xl,p,q);   % dcj_q/dxk_p
    dcj(q,4) = delta_perm(xk-xl,xi-xl,p,q);   % dcj_q/dxl_p
end

function v = delta_perm(x,y,p,q)
kd = eye(3);
e = [2,3; 3,1; 1,2];
r1 = e(q,1); r2 = e(q,2);
v = (x(r1)-y(r1))*kd(p,r2) - (x(r2)-y(r2))*kd(p,r1);

function dA = d_detA(xi,xj,xk,xl)
dA = zeros(3,4);
dA(:,1) = cross(xj-xl,xk-xl)';
dA(:,2) = cross(xk-xl,xi-xl)';
dA(:,3) = cross(xi-xl,xj-xl)';
dA(:,4) = -sum(dA(:,1:3),2);

function s = change_cond2anisotropic(cond_type,s0,nt)

switch cond_type.string
    case 'anisotropic'
        s = s0;
    case 'layered'
        n = cond_type.normals;
        s = zeros(6*nt,1);
        I = eye(3);
        for t = 1:nt
            idx = (t-1)*2;
            ss = (s0(idx+1)-s0(idx+2))*(n(t,:)'*n(t,:)) + s0(idx+2)*I;
            idx = (t-1)*6+(1:6);
            s(idx) = [ss(1,1),ss(1,2),ss(1,3),ss(2,2),ss(2,3),ss(3,3)]';
        end
    case 'isotropic'
        s = zeros(6*nt,1);
        for t=1:nt
            idx = (t-1)*6+(1:6);
            s(idx) = [s0(t),0,0,s0(t),0,s0(t)]';
```

```
        end
    case 'conformal'
        A = cond_type.constraints;
        s = zeros(6*nt,1);
        for t=1:nt
            idx = (t-1)*6+(1:6);
            s(idx) = s0(t)*A(idx);
        end
    otherwise
        error('%s not recongnised as a possible conductivity type.',...
            cond_type.string);
end
```

```
% plot_edgecond2vertpos_jacobian.m

%% Load netgen mesh
meshdir = './NetgenMeshes/';
meshfile = 'sphere_very_coarse';
% meshfile = 'cylinder_v_coarse';
[~,H, V] = mesh_reader([meshdir,meshfile,'.vol']);

%% Add a few more interior verts and remesh
if strcmp(meshfile,'sphere_very_coarse');
    V_extra = [-0.45,0.1,0.08; 0.46,0.04,-0.1; 0.03,-0.44,0.04; ...
        0.02,0.42,0.09;0.01,-0.02,-0.46; -0.04,0.06,0.43];
    V = [V; V_extra];
    H = delaunay(V);
end

%% Get parameters
get_mesh_parameters

%% Tree:
tree = grMinSpanTree(D);
rt = 1;
Dm=treesort(D(tree,:),rt); % easier if a directed tree
```

```
%% Random number stream setup
defaultStream = RandStream.getDefaultStream;
load randstream;
defaultStream.State = savedState;

%% Define type of conductivity perturbation
% % Anisotropic
% sv = repmat([1,0,0,1,0,1],nv,1);
% s = repmat([1,0,0,1,0,1]',nt,1);    % tets
% cond_type_test.string = 'anisotropic';

% Layers:
cond_type_test.string = 'layered';
cond_type_test.normals = Centroids; % spherical
% cond_type_test.normals = repmat([0,0,1],nt,1);
% r = 0.1*randn(nt,3);
% cond_type_test.normals = cond_type_test.normals + r;
s = repmat([1,1]',nt,1);
sv = repmat([1,1],nv,1);

% % Isotropic:
% cond_type_test.string = 'isotropic';
% sv = ones(nv,1);
% s = ones(nt,1);

% % Multiplicative constant:
% cond_type_test.string = 'mult_const';
% cond_type_test.constraints = repmat([1.2,-0.2,-0.1,1.1,0,1]',nt,1);
% sv = ones(nv,1);
% s = ones(nt,1);

Q = cond2edgecondmap(H,E,V,cond_type_test);
C = Q*s;

%% Options
options.fix_boundary = true;
options.fix_cond_boundary = false;
options.conductivity = 'tets';
```

```
%% Jacobian at a point where constraints are met (i.e. initial mesh)
% p = [1e-5,1e-7,1e-9,1e-11,1e-13,1e-15];
% % p = [1e-8,1e-15];
% J = cell(length(p),1);
% for i=1:length(p)
%     fprintf(1,'Perturbation: %.0e ...\n',p(i));
%     options.perturbation = p(i);
%     J{i} = edgecond2vertices_jacobian(H,F,E,find(~isbnde),...
%         Dm,V,C,s,cond_type_test,options);
% end

%% Analytic:
options.jacobian_type = 'analytic';
options.perturbation = 1e-8;
fprintf(1,'Analytic Jacobian ...\n');
Ja = edgecond2vertices_jacobian(H,F,E,find(~isbnde),...
    Dm,V,C,s,cond_type_test,options);

%% Plot singular values
% leg = cell(length(J)+1,1);
% col_str = 'robogokomocor*';
% for i=1:length(J)
%     [QQ,RR] = qr(J{i});
%     [Uv,Sv,Vv] = svd(QQ*RR);
%     Sv = diag(Sv)./Sv(1,1);
%     semilogy(Sv,col_str((i-1)*2+(1:2))); hold on;
%     leg{i} =sprintf('p = %.0e',p(i));
% end
[QQ,RR] = qr(Ja(:,1:3*nvi));
[Uv,Sv,Vv] = svd(QQ*RR);
[QQ,RR] = qr(Ja(:,3*nvi+1:end));
[Us,Ss,Vs] = svd(QQ*RR);
% Sv = diag(Sv)./Sv(1,1);
% figure(1); clf; semilogy(diag(Sv),'r*');%,'LineStyle','-');
% hold on; semilogy(diag(Ss),'bo'); hold off%,'LineStyle','-');

J = [Ja(:,1:3*nvi)/Sv(1),Ja(:,3*nvi+1:end)/Ss(1)];
[QQ,RR] = qr(J);
[UU,SS,VV] = svd(QQ*RR);
```

```
SS = diag(SS)./SS(1,1);
figure(4); semilogy(SS,'o');
ax = axis; ax(3:4) = [1e-18,1e0]; axis(ax);
% leg{length(J)+1} = 'Analytic';
% hold off;
% legend(leg);

figure(2); imagesc(VV(:,1:ne)); axis image; colorbar
figure(3); imagesc(VV(:,ne+1:end)); axis image; colorbar
```