DISSERTATION

A POSTERIORI ERROR ESTIMATES FOR THE POISSON PROBLEM ON CLOSED, TWO-DIMENSIONAL SURFACES

Submitted by

William F. Newton

Department of Mathematics

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Summer 2011

Doctoral Committee:

Advisor: Donald Estep

Michael Holst
Yongcheng Zhou
Jay Breidt
Simon Tavener

ABSTRACT

A POSTERIORI ERROR ESTIMATES FOR THE POISSON PROBLEM ON CLOSED, TWO-DIMENSIONAL SURFACES

The solution of partial differential equations on non-Euclidean Domains is an area of much research in recent years. The Poisson Problem is a partial differential equation that is useful on curved surfaces. On a curved surface, the Poisson Problem features the Laplace-Beltrami Operator, which is a generalization of the Laplacian and specific to the surface where the problem is being solved. A Finite Element Method for solving the Poisson Problem on a closed surface has been described and shown to converge with order h^2. Here, we review this finite element method and the background material necessary for defining it. We then construct an adjoint-based a posteriori error estimate for the problem, discuss some computational issues that arise in solving the problem and show some numerical examples. The major sources of numerical error when solving the Poisson problem are geometric error, discretization error, quadrature error and measurement error. Geometric error occurs when distances, areas and angles are distorted by using a flat domain to parametrize a curved one. Discretization error is a result of using a finite-dimensional space of functions to approximate an infinite-dimensional space. Quadrature error arises when we use numerical quadrature to evaluate integrals necessary for the finite element method. Measurement error arises from error and uncertainty in our knowledge of the surface itself. We are able to estimate the amount of each of these types of error and show when each type of error will be significant.

ACKNOWLEDGMENTS

This dissertation could not have been written without the help and support of many people. With apologies to anyone I may miss, I would like to thank the following people.

My mathematical education as an undergraduate student at the University of Utah prepared me very well for my graduate education. Some professors who were especially valuable in my education there were Dr Robert Bell, Dr Grant Gustafson, Dr Peter Alfeld, and Dr James Keener. I would like to especially thank James Keener for being the math professor who first captured my imagination. I also give special thanks to the late Dr Janet Andersen, who supervised the first research project I worked on and was a valued and trusted mentor for me.

At Colorado State University, I have had wonderful colleagues among my fellow graduate students. They have been major part of my success in the graduate program. Among them have been Troy Butler, Eric Holt, Byungsoo Kim, Keith Mertens, Megan Buzby, Alan Von Herrman and Jutta Bikowski. My experience at Colorado State University would not have been the same without them and their influence and support is greatly appreciated. I would also like to thank Nathanial Burch for his valuable help in formatting my dissertation. Bryan Elder has also been very helpful thoughout my time as a graduate student, and he has my thanks.

I have also had great professors at Colorado State. The late Dr Daniel Rudolph continues to be a major influence in the way I think about math and my love for the subject. I miss him greatly. Dr Barry Noon has taught me much about ecology, research and life in general. I greatly appreciate his influence on my time as a graduate student. Dr Dan Cooley has been a pleasure to work with and a great mentor in research in teaching. I greatly appreciate the time I have had with him. I also would, of course, like to thank the members of my committee: Dr Yongcheng Zhou, Dr Simon Tavener, Dr Michael Holst and Dr Jay Breidt. I appreciate their time and help. Most of all, I would like to thank my advisor, Dr Donald Estep. No one else has been more important to my mathematical development and to my success at Colorado State University.

Finally, I would like to thank my family for their love and support during my education. My

parents, Bill and Bonnie Newton, taught me to be a good student and hard worker who was curious about the world. My sisters, Angela Berry and Ashley Newton, and my brother-in-law, John Berry, have given me valuable support and camaraderie though my time at Colorado State University. Finally, I thank my wife, Heather. Without her love and support, I could not be where I am today. I love all of you very much and I thank you very much.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ALGORITHMS

1. INTRODUCTION

Suppose we have an object with a shape close to the warped torus in Figure 1.

Suppose that it is partially immersed in water with temperature T_1, while the surrounding air is at temperature T_2. Suppose further that we wish to know the difference between the temperature of the surface underwater and the average temperature of the surface above water at equilibrium. To do this, we will need to solve the Poisson equation,

$$-\Delta_\Gamma u = f(x),$$

on the surface. $u(x)$ is the tempertaure at x. Here, Γ is the surface, Δ_Γ is the Laplace-Beltrami operator for the surface and $f(x)$ describes the forcing as a result of the different temperatures. This problem can be solved using a finite element method as in [12]. This finite element method is valid for an arbitrary closed surface with an arbitrary forcing term as long as both are sufficiently regular. Using that method, there will be numerical error from the following sources.

- Geometric error when areas, distances and angles are distorted by using a piecewise flat domain to approximate a curved one.
- Discretization error due to looking for an approximate solution in a finite-dimensional space of functions.
- Quadrature error due to using numerical quadrature to evaluate the integrals necessary for the finite element method.
- Measurement error of the surface itself – the measurements of the object are subject to error.

Our method for representing the surface and computing the solution to the Poisson Problem comes from [12]. The a priori error estimate in this paper is extended to second order quasi-linear systems in [19]. In [10], the a priori convergence result is extended to higher-order finite elments.

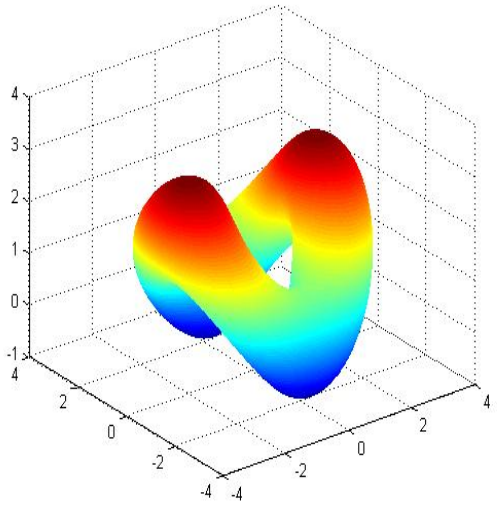

Fig. 1.1: A warped torus

In [20], the work in [12] and [10] is extended to higher dimensions and to general differential forms. In [21], the a priori convergence results are further extended to nonlinear problems. In [11], an a posteriori error estimate for the problem using this method is introduced. A posteriori error estimates are also considered in [19]. The estimate in [11] does not use duality. The adjoint is defined but not used to estimate error in [19]. In this dissertation, we extend the methods in [16] and [17] to the problem and develop an a posteriori error estimate that does use the adjoint. Other work that has been done on partial differential equations on surfaces includes [14], where time-dependent problems involving the Laplace-Beltrami operator are solved on evolving surfaces. It is possible to extend the work in this dissertation to all of this analysis. Nonlinear diffusion problems on surfaces may be relevant to applications in biology such as pattern formation as in [3].

The Poisson Problem can also be solved on surfaces using other approaches. In [1], the Poisson Problem on a sphere is solved using a global parameterization. In [2], spherical finite elements are used to solve the problem on a sphere. In [26], curved elements are used to approximate surfaces. It is also possible to look at the problem without explicitly representing the surface. We instead have a level set of a function defined in a three dimensional set in $\mathbb{R}^3$. This is the approach taken in [4], [5], [6] and [22].

In this paper, we review the background material necessary to understand the finite element method and then develop adjoint-based a posteriori error estimates for each of these types of error.

Estimating each of these error sources is useful when refining the mesh in order to gain a more accurate solution. The error estimates are specific to the problem being solved and the quantity of interest being computed from the solution, so they are useful for measuring how much of what type of error comes from every part of the surface. This is useful for efficient refinement of the mesh, if necessary. We discuss some computational issues that arise when computing the solution and error estimates and how they can be addressed. Among the issues we discuss are the well-posedness of the problem and specifying the forcing term f in a manner that is computationally feasible.

In Chapter 2, we consider some elementary concepts from differential geometry. These concepts are important to understanding the finite element method for the problem.

In Chapter 3, we consider the Laplacian and its various meanings. The Laplacian of a function is related to the difference between the value of the function at the point and the average value of neighboring points. We then construct the Laplace-Beltrami Operator, which has the same properties for a function defined on a surface.

In Chapter 4, we review numerical methods for solving the Poisson Problem on a Euclidean Domain. We consider various methods for dealing with the error in the finite element method for this problem, including an adjoint-based a posteriori error estimate.

In Chapter 5, we review the finite element method for the Poisson Problem on a Closed Surface and introduce an adjoint-based a posteriori error estimate for the problem. We interpret and show how to compute each term in the estimate. An a posteriori error estimate is an estimate that is computed after the approximate solution has been computed. Since it applies only to the solution that has been computed, rather than to a broad class of problems as in the case with an a priori error bound, it is generally much sharper and more useful. We also show that the error estimate goes to zero with order h^2.

In Chapter 6, we show the details of implementing the finite element solution and error estimate.

In Chapter 7, we show numerical examples of the finite element method and error estimate.

In Chapter 8, we discuss dealing with measurement error and uncertainty about the surface and show some numerical examples.

2. BASIC NOTIONS FROM DIFFERENTIAL GEOMETRY

2.1 Basic Definitions

Definition 1 (Manifold)**.** *A subset Γ of $\mathbb{R}^n$ is said to be a k-dimensional manifold if, at every point $x \in M$, there is an open set $W \subset \mathbb{R}^n$ containing x such that $M \cap W$ is homeomorphic to $\mathbb{R}^k$. (Here, k is less than n.)*

A two-dimensional manifold in $\mathbb{R}^3$ is usually called a *surface*. This paper deals with surfaces. In addition, most of the surfaces considered here are closed surfaces.

Definition 2. *If Γ is a $k-$dimensional manifold, x is said to be a boundary point of Γ if there is an open set W of $\mathbb{R}^n$ containing x that is homeomorphic to $\mathbb{R}^k_+$ where x is mapped to the origin.*
A manifold is said to be closed if it contains no boundary points.

A closed surface is simply a closed two-dimensional manifold in $\mathbb{R}^3$. A homeomorphism from a subset of $\mathbb{R}^k$ to a subset of Γ given by $\Gamma \cap W$ as above is called a *surface patch*. A collection of surface patches where every $x \in \Gamma$ is in the image of at least one surface patch is called an *atlas*. This produces a way to describe the whole manifold. In this paper, we perform calculus on surfaces. For this to be possible, a surface needs to have a certain amount of regularity, which can be characterized using surface patches and atlases.

Definition 3. *For $k \in \mathbb{N}$, an atlas is said to be C^k if, for any two surface patches with homeomorphisms f and g, with images $\Im f$ and $\Im g$ respectively, where $\Im f \cap \Im g$ is not empty, $g^{-1}(f(x))$ is C^k wherever it is defined.*

Definition 4. *A surface is C^k if it has an atlas that is C^k.*

Definition 5. *A surface is said to be orientable if it has an atlas where, for any two surface patches with homeomorphisms f and g where $\Im f \cap \Im g$ is not empty, $det(J(g^{-1}(f(x)))) > 0$ wherever it is defined. J is the Jacobian of the transformation $g^{-1}(f(x))$.*

Definition 6. *A surface is said to be regular if it is C^∞ and it has an atlas where every surface patch has a Jacobian that is rank* 2*.*

We carry out computations under the assumption of regular, orientable surface patches. Most examples are also closed and bounded.

2.2 Representing a Surface

The three main methods for representing a surface are as a parameterization, as a graph and as a level set of a function.

Parameterization

When we use a parameterization to represent a surface, we mean that we are using a convenient atlas. We have a set (or a union of sets) in $\mathbb{R}^2$ and a function (or functions) taking the set(s) to Γ.

Example 1. *The Unit Sphere can be parametrized using spherical coordinates:*

$$x = \cos u \cos v$$

$$y = \cos u \sin v$$

$$z = \sin u.$$

Here, $u \in [0, 2\pi]$ and $v \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$.

We assume, unless stated otherwise, that we have an atlas consisting of a single surface patch with a homeomorphism where each component function is C^∞.

Once we have a parameterization, we are free to define functions on the surface by defining them on the domain of the parameterization. The problem then arises of the accurate representation of distances. We are, in effect, mapping a curved surface to a flat surface with the parameterization. A well-known example of this problem arises in Geography: any map of the entire world distorts distances. [23] One consequence is that if we have a function defined on a domain Θ where there is a map $\theta : \Gamma \to \Theta$, naively taking derivatives of the function does not give accurate results – the derivatives will depend on the parameterization and will not be consistent under alternative parameterizations.

Definition 7 (Metric Matrices)**.** *If we have a domain Θ with a parameterization $\sigma : \Theta \to \Gamma$, we denote the Jacobian of f by J. The generalized inverse of J is denoted by L. We can define the matrix $G = J^\top J$ and its inverse $G^{-1} = L^\top L$.*

Remark 1. *We can see, using the singular value decomposition, that* $J^\top L = L^\top J = I_2$ *and* $JL^\top = LJ^\top = I - \boldsymbol{nn}^\top$*, where* $\boldsymbol{n}$ *is the unit normal vector to* Γ.

The matrix G is known as the *First Fundamental Form* of the surface. It is of great importance for performing calculus on the surface because it tells us how to measure lengths. A curve on the surface can be represented by a function

$$\gamma : [0, T] \to \Theta$$

Then $f(\gamma(t))$ is on the surface. Its length is given by

$$\int_0^T \gamma'(t)^\top G \gamma'(t) dt$$

Note that if we let $\xi(t) = f(\gamma(t))$, then $\xi'(t) = J\gamma'(t)$, so this is really just the standard formula for arc length.

If we have a different parameterization for Γ, it is equivalent to having a second set Θ_1 with a one-to-one function $\beta : \Theta_1 \to \Theta$ and then the function taking Θ_1 to Γ is just $f(\beta(u, v))$. If the Jacobian of β is given by T, then the Jacobian of $f(\beta(u, v))$ is given by JT. The first fundamental form of the new parameterization is given by $T^\top GT$. The reparameterization gives the same unit normal vector, since the columns of JT span the same space as the columns of J.

Definition 8. *The second fundamental form of the surface,* B*, is given by*

$$\begin{pmatrix} \boldsymbol{f}_{xx} \cdot \boldsymbol{n} & \boldsymbol{f}_{xy} \cdot \boldsymbol{n} \\ \boldsymbol{f}_{xy} \cdot \boldsymbol{n} & \boldsymbol{f}_{yy} \cdot \boldsymbol{n} \end{pmatrix}$$

Under a change of parameters as before, the new second fundamental form is $T^\top BT$.

Representation as a Graph

When we think of a surface as a graph, it means that the graph is given as

$$\begin{pmatrix} x \\ y \\ u(x, y) \end{pmatrix}$$

for some function $u(x, y)$. We can see that this is just a special case of a parameterization. We have

$$G = I + \nabla u^\top \nabla u$$

$$G^{-1} = I - \frac{1}{1 + \|\nabla u\|^2} \nabla u^\top \nabla u$$

$$B = \frac{1}{\sqrt{1 + \|\nabla u\|^2}} Hu$$

where Hu denotes the Hessian matrix of u.

Level Sets

A surface can also be represented as the level set of a function of three variables $h(x, y, z)$. The unit sphere, for example, is given by $x^2 + y^2 + z^2 = 1$. Locally, we can use the implicit function to think of this as a special case of a graph, with

$$\begin{pmatrix} x \\ y \\ z(x, y) \end{pmatrix}.$$

For a surface patch where this is valid, the formulas from above apply with

$$\nabla u = \begin{pmatrix} -\frac{h_x}{h_z} & -\frac{h_y}{h_z} \end{pmatrix}.$$

2.3 Curvature

Definition 9 (Curvature)**.** *Let $\boldsymbol{x}(t)$ be a C^2 curve where $\|\dot{\boldsymbol{x}}(t)\| > 0$ for all t. If s is the arc length, then we define the curvature to be*

$$\kappa(t) = \|\ddot{\boldsymbol{x}}(s)\|.$$

The following formula is useful for curves in $\mathbb{R}^3$:

Theorem 1. *If $\boldsymbol{x}(t)$ is a curve in $\mathbb{R}^3$, the curvature is given by*

$$\kappa = \frac{\|\ddot{\boldsymbol{x}}(t) \times \dot{\boldsymbol{x}}(t)\|}{\|\dot{\boldsymbol{x}}(t)\|^3}.$$

Two important special cases of this formula are when $\mathbf{x}$ is a plane curve:

$$\kappa = \frac{|x'y'' - y'x''|}{((x'^2) + (y'^2))^{3/2}},$$

and when y is given as a function of x:

$$\kappa = \frac{|y''|}{(1 + y'^2)^{3/2}}.$$

For a plane curve, the curvature has an important geometric interpretation. Let $\mathbf{x}(t)$ be a unit-speed plane curve, so that the tangent vector $\dot{\mathbf{x}}(t)$ is a unit vector. We can define the *signed normal unit vector* $\mathbf{n}_S$ by rotating the tangent vector counterclockwise by $\frac{\pi}{2}$. Since the curve is unit-speed, $\ddot{\mathbf{x}}(t)$ is perpendicular to the tangent vector, and hence we have

$$\ddot{\mathbf{x}} = \kappa_S \mathbf{n}_S,$$

and

$$\kappa = \|\ddot{\mathbf{x}}\| = \|\kappa_s \mathbf{n}_S\| = |\kappa_S| \,.$$

We call κ_S the *signed curvature.*

Now if Δt is small, the distance between $\mathbf{x}(t + \Delta t)$ and the tangent line at t is given by

$$|(\mathbf{x}(t + \Delta t) - \mathbf{x}(t))\mathbf{n}_S| \,.$$

Expanding $\mathbf{x}(t + \Delta t)$, we get

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \dot{\mathbf{x}}(t)\Delta t + \ddot{\mathbf{x}}(t)\frac{(\Delta t)^2}{2} + \ldots.$$

Since $\mathbf{n}_S$ is orthogonal to the tangent vector and $\ddot{\mathbf{x}} = \kappa_S \mathbf{n}_S$, we get

$$|(\mathbf{x}(t + \Delta t) - \mathbf{x}(t))\mathbf{n}_S| \approx \frac{\kappa}{2}\Delta t^2.$$

So the curvature measures how much a curve moves away from the tangent line when t is incremented by a small amount.

We can also define the curvature of a surface. To do so, we need the First and Second Fundamental Forms of the surface. Here, we show alternate notation for them.

Definition 10 (The First Fundamental Form, Alternate Notation)**.** *The First Fundamental Form of a surface given by $\sigma(u, v)$ is gven by*

$$Edu^2 + 2Fdudv + Gdv^2$$

where

$$E = \|\sigma_u\|^2 \,, F = \sigma_u \cdot \sigma_v, G = \|\sigma_v\|^2 \,.$$

Definition 11 (The Second Fundamental Form, Alternate Notation)**.** *If $\sigma(u, v)$ defines a surface with normal unit vector $\boldsymbol{n}$, then the second fundamental form of the surface is given by*

$$Ldu + 2Mdudv + Ndv^2$$

where

$$L = \sigma_{uu} \cdot \boldsymbol{n}, M = \sigma_{uv} \cdot \boldsymbol{n}, N = \sigma_{vv} \cdot \boldsymbol{n}.$$

We can now measure the curvature of a surface by considering the curvatures of unit-speed curves on the surface. We consider such a curve with unit-speed defined by

$$\mathbf{x}(t) = \sigma(u(t), v(t)).$$

First note an important similarity between the second fundamental form of the surface and the curvature of a plane curve. The distance between $\mathbf{x}(t + \Delta t)$ and the tangent plane at t is given by

$$(\mathbf{x}(t + \Delta t) - \mathbf{x}(t)) \cdot \mathbf{n}.$$

Expanding $\mathbf{x}(t + \Delta t)$, we get

$$\mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \dot{\mathbf{x}}(t)\Delta t + \ddot{\mathbf{x}}(t)\frac{\Delta t^2}{2}.$$

Now

$$\ddot{\mathbf{x}}(t) = \sigma_{uu} \|\dot{u}\| + 2\sigma_{uv}\dot{u} \cdot \dot{v} + \sigma_{vv} \|\dot{v}\|^2 .$$

Since $\dot{\mathbf{x}}(t)$ is orthogonal to the normal vector, the distance between $\mathbf{x}(t)$ and the tangent plane is approximately

$$\frac{1}{2}\left(\sigma_{uu} \cdot \mathbf{n} \|\dot{u}\|^2 + 2\sigma_{uv} \cdot \mathbf{n}\dot{u} \cdot \dot{v} + \sigma_{vv} \cdot \mathbf{n} \|\dot{v}\|^2\right).$$

So the second fundamental form approximates the displacement from the tangent plane for a unit-speed curve on the surface defined by σ and is analgous in this way to the curvature of a plane curve. Because of this, we define the normal curvature of a surface curve by

$$\kappa_N = \begin{pmatrix} \dot{u} & \dot{v} \end{pmatrix} \begin{pmatrix} L & M \\ M & N \end{pmatrix} \begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix},$$

where L, M, N are given by the second fundamental form of the surface. It might appear that the extreme values of the normal curvature should be given by the eigenvalues of the second fundamental form. This is not the case, however, since we are only allowing unit-speed curves, and the speed is given by

$$\|\gamma\|^2 = \dot{\gamma}^\top G \dot{\gamma},$$

where G is the matrix given by the first fundamental form. To find the critical points of the normal curvature, we need to calculate the roots of

$$\det(B - \kappa G) = 0.$$

We call the roots of this polynomial the principal curvatures of the surface at the point x. There are n roots (assuming this is an n-dimensional surface), when counted with multiplicities. It is not immediately clear that all the principal curvatures are real. This can be proven, however.

Theorem 2. *The principal curvatures are all real.*

Proof. We let the vectors $\mathbf{t}_1, \ldots, \mathbf{t}_n$ be an orthonormal basis of the tangent space. We have, for every i,

$$\mathbf{t}_i = \sum_{k=1}^{n} c_{ik} \sigma_{u_k},$$

where the vectors σ_{μ_k} are the partial derivatives of the parameterization σ, for some invertible matrix C. We now have

$$\begin{aligned} \delta_{ij} &= \\ &= \mathbf{t}_i \cdot \mathbf{t}_j \\ &= \sum_{k=1}^{n} \sum_{l=1}^{n} c_{ik} c_{jl} \sigma_{\mu_k} \sigma_{\mu_l} \\ &= \mathbf{c}_i^\top G \mathbf{c}_j. \end{aligned}$$

This shows that

$$C^\top G C = I.$$

Note that $C^\top \neq C^{-1}$. We now have

$$C^\top (B - \kappa G) C = A - \kappa I$$

for a matrix A that is symmetric since B is. The principle curvatures are the eigenvalues of this matrix and are therefore all real. □

We note that the matrix A in this proof is not unique since the choice of the orthonormal basis was arbitrary. However, since there is an orthonormal similarity transform between any two orthonormal bases, the matrix A is unique up to a similarity transform, and the set of eigenvalues (and hence principal curvatures) is therefore also unique.

A very important matrix whose eigenvalues also give the principal curvatures is the Weingarten matrix, given by

$$W = G^{-1} B.$$

It is clear from the definition that the eigenvalues of W are the principal curvatures. We can see from this fact that the principal curvatures do not change if the surface is reparametrized, since the new

fundamental forms are given by $M^\top GM$ and $M^\top BM$ for some invertible matrix M, and the new Weingarten matrix is therefore simlar to the old one. The Weingarten matrix has another important interpretation. Consider the standard unit normal vector $\mathbf{n}$. Since it is a unit vector, $\mathbf{n} \cdot \mathbf{n}_{u_k} = 0$ for all k. This means that the vectors $\mathbf{n}_{u_k}$ are all in the tangent space and there is therefore a matrix A such that

$$\mathbf{n}_{u_j} = \sum_{k=1}^{n} a_{jk} \sigma_{u_k}$$

for every j.

Since, for every i, $\mathbf{n} \cdot \sigma_{u_i} = 0$, we have for every j,

$$\begin{aligned} \mathbf{n}_{u_j} \cdot \sigma_{u_i} + \mathbf{n} \cdot \sigma_{u_i u_j} &= 0 \\ \mathbf{n}_{u_j} \cdot \sigma_{u_i} &= -b_{ij} \\ \sum_{k=1}^{n} a_{jk} \sigma_{u_k} \cdot \sigma_{u_i} &= -b_{ij}, \\ \sum_{k=1}^{n} a_{jk} g_{ki} &= -b_{ij} \end{aligned}$$

and we have

$$A^\top = -W.$$

With the principal curvatures defined, we can define two other very important notions of curvature.

Definition 12 (Gaussian Curvature)**.** *The Gaussian Curvature K of a surface is the product of the principal curvatures.*

Definition 13 (Mean Curvature)**.** *The Mean Curvature H of a surface is the sum of the principal curvatures.*

Remark 2. *A common alternate definition for the mean curvature is the sum of the principal curvatures divided by the dimension of the surface.*

2.4 Special Parameterizations

In this section, we consider two special categories of parameterizations: conformal parameterizations and isometric parameterizations.

Conformal Parameterizations

A *conformal parameterization* is a parameterization that preserves angles mapped from $\mathbb{R}^2$ to Γ. Every regular two-dimensional surface has an atlas consisting of conformal surface patches. The first fundamental form for a conformal parameterization takes the form

$$G = \begin{pmatrix} E & 0 \\ 0 & E \end{pmatrix}.$$

Its second fundamental form takes the form

$$B = \begin{pmatrix} L & 0 \\ 0 & N \end{pmatrix}.$$

The Weingarten matrix is then

$$W = \begin{pmatrix} \frac{L}{E} & 0 \\ 0 & \frac{N}{E} \end{pmatrix},$$

and then the Gaussian and Mean Curvatures are

$$K = \frac{LN}{E^2}$$

and

$$H = \frac{L+N}{E}.$$

Based on the properties of the first fundamental form for this parameterization, we have

$$\sigma_u \cdot \sigma_v = 0$$

and

$$\sigma_u \cdot \sigma_u = \sigma_v \cdot \sigma_v.$$

If we differentiate both of these equations with respect to both parameters, we get the conditions

$$\sigma_{uu} \cdot \sigma_v + \sigma_u \cdot \sigma_{uv} = 0$$

$$\sigma_u \cdot \sigma_{vv} + \sigma_{uv} \cdot \sigma_v = 0$$

$$\sigma_u \cdot \sigma_{uu} = \sigma_{uv} \cdot \sigma_v = 0$$

$$\sigma_u \cdot \sigma_{uv} = \sigma_v \cdot \sigma_{vv} = 0.$$

From these conditions we see that

$$(\sigma_{uu} + \sigma_{vv}) \cdot \sigma_u = (\sigma_{uu} + \sigma_{vv}) \cdot \sigma_v = 0.$$

The Laplacian of the parameterization is therefore spanned by the unit normal vector:

$$\Delta\sigma = \|\Delta\sigma\| \, \mathbf{n},$$

and therefore, the norm of the Laplacian is proportional to the mean curvature,

$$H = \frac{L+N}{E} = \frac{\sigma_{uu} \cdot \mathbf{n} + \sigma_{vv} \cdot \mathbf{n}}{E} = \frac{\|\Delta\sigma\|}{E}.$$

Isometric Parameterizations

An isometric parameterization is a parameterization that maps curves in $\mathbb{R}^2$ to curves on M of the same length. Its first fundamental form is simply the identity matrix,

$$G = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

As we will see later, an isometric paramterization leads to a second fundamental form of rank at most one. The Gaussian curvature is therefore zero. Only surfaces with this property can be parametrized isometrically. More generally, any isometric mapping between two surfaces maintains the Gaussian curvature.

3. THE LAPLACIAN IN EUCLIDEAN SPACE

The Laplacian of a function in $\mathbb{R}^n$ is defined as

$$\Delta u = \sum_{j=1}^{n} \frac{\partial^2 u}{\partial x_k^2}$$

and arises in many situations. We have already seen its relationship to the mean curvature of a surface that is conformally parametrized. In this chapter, we discuss several geometric interpretations of the Laplacian and then develop the Laplace-Beltrami operator, a generalization of the Laplacian for surfaces.

3.1 The Energy Functional and the Calculus of Variations

If Ω is an open subset of $\mathbb{R}^n$ and $u \in C^2(\Omega)$, we can define its energy functional to be

$$E(u) = \frac{1}{2} \int_\Omega \sum_{j=1}^{n} \left(\frac{\partial u}{\partial x_j} \right)^2 dV.$$

This is an example of a functional, or a real-valued function with a set of functions as its domain. It arises in many physical applications. The Calculus of Variations is the branch of mathematics concerned with finding functions that minimize functionals. We look for minimizers by first finding the variation of the functional. The variation of a functional is analagous to the derivative of a function. It is a linear functional that approximates the functional for small perturbations. To get the variation of the energy functional, we let $\epsilon(x)$ be a small perturbation of the function $u(x)$ where $\epsilon(x)$ is zero at the boundary of U. Then

$$\begin{aligned} E(u+\epsilon) &= \frac{1}{2} \int_\Omega \sum_{j=1}^{n} \left(\frac{\partial u}{\partial x_j} + \frac{\partial \epsilon}{\partial x_j} \right)^2 dV \\ &= \frac{1}{2} \int_\Omega \sum_{j=1}^{n} \left(\frac{\partial u}{\partial x_j} \right)^2 dV + \int_\Omega \sum_{j=1}^{n} \frac{\partial u}{\partial x_j} \frac{\partial \epsilon}{\partial x_j} dV + O(\epsilon^2). \end{aligned}$$

We neglect the higher-order term, assuming that ϵ is a small perturbation. We have, therefore,

$$E(u+\epsilon) \approx E(u) + \int_\Omega \sum_{j=1}^{n} \frac{\partial u}{\partial x_j}\frac{\partial \epsilon}{\partial x_j} dV.$$

Now, using the fact that ϵ is zero at the boundary of Ω, integration by parts on the second term yields

$$\begin{aligned}\int_\Omega \sum_{j=1}^{n} \frac{\partial u}{\partial x_j}\frac{\partial \epsilon}{\partial x_j} dV &= \int_{\partial\Omega} \epsilon \frac{\partial u}{\partial \mathbf{n}} d\sigma - \int_\Omega \epsilon \sum_{j=1}^{n} \frac{\partial^2 u}{\partial x_j^2} dV \\ &= -\int_\Omega \epsilon \Delta u dV.\end{aligned}$$

This last term is a linear functional of the perturbation $\epsilon(x)$, and so is the variation of the energy functional. The Laplacian of u, $-\Delta u = \sum_{j=1}^{n} \frac{\partial^2 u}{\partial x_j^2}$ is related to the variation of the energy functional and indicates how far u is from being a minimizer of the energy functional.

3.2 The Divergence Theorem

Theorem 3 (The Divergence Theorem)**.**

$$\int_{\partial\Omega} \frac{\partial u}{\partial \nu} d\sigma = -\int_\Omega \Delta u dV.$$

If u is the concentration of a quantity such as heat or a chemical on the domain Ω, then the integral on the left hand side gives the flux of u through the boundary of Ω. We can see, using the mean value theorem for integrals, that

$$\Delta u = \lim_{r\to 0} \frac{1}{\pi r^2} \text{flux}(u)$$

through a circle of radius r containing the point x.

Another important theorem regarding the Laplacian of a function is the following:

Theorem 4. *If we take the closed path integral of a function u around the circle of radius r around a point, we get*

$$\lim_{r\to 0} \frac{1}{\pi r^2} \left(u(x) - \frac{1}{2\pi r} \oint u d\sigma \right) = -\Delta u.$$

So the Laplacian of a function indicates the difference between the value of u at x and the values of u at nearby points.

3.3 Random Walks

The Laplacian also emerges from a random walk. As an example, consider a biological population $u(t)$. A very general way to model the population might be

$$u(t) = \int_{t_0}^{t} k(s, u(s))ds$$

where the kernel function k depends both on the population in the past and external conditions that change with time. In field studies, it is common to estimate the population at intervals and measure the proportional change:

$$\frac{u(t+\Delta t)}{u(t)} = \int_{t}^{t+\Delta t} k(s, u(s))ds = c_t.$$

We now let $U(t)$ be the natural logarithm of $u(t)$, and we consider the probability density function $P(x,t)$ for $U(t)$. The probability that U is in some interval of length Δx at a future time $t + \Delta t$ is given by

$$\int_{x}^{x+\Delta x} P(\xi, t+\Delta t)d\xi = \int_{x}^{x+\Delta x} \int P(\xi, t+\Delta t|y,t)P(y,t)dyd\xi.$$

Here,

$$P(\xi, t+\Delta t|y,t)$$

is the conditional probability density function for ξ at $t+\Delta t$ given y at t. A very simple assumption for $P(x, t+\Delta t|y,t)$ might be

$$P(x, t+\Delta t|y,t) = .25\delta(y-(x-\Delta x)) + .5\delta(y-x) + .25\delta(y-(x+\Delta x)),$$

where δ is the Dirac delta function. This kernel means that if U is x at time t, then at time $t+\Delta t$ it will be $x+\Delta x$ with probability .25, $x-\Delta x$ with probability .25, or x with probability .5. More generally, it means that we expect small changes in the amount of time Δt but that increases are as likely as decreases. The exact kernel is less important than how it integrates. In this case, we let

$$P_k^t = \int_{x+(k-1)\Delta x}^{x+k\Delta x} P(y,t)dy.$$

We get

$$P_k^{t+\Delta t} = .25P_{k-1}^t + .5P_k^t + .25P_{k+1}^t$$

Then,

$$P_k^{t+\Delta t} = P_k^t + \Delta t \frac{\partial}{\partial t} P_k^t + O(\Delta t^2).$$

and

$$.25P_{k-1}^t + .5P_k^t + .25P_{k+1}^t = P_k^t + (\Delta x^2)\Delta_x P_k^t +) + O(\Delta x^4),$$

where Δ_x is the Laplacian in the x variable.

If we neglect higher order terms, this yields

$$\frac{\partial}{\partial t} P_k^t = \frac{\Delta x^2}{\Delta t} \Delta_x P_k^t.$$

Now, if P is a C^2 function, we can take the limit as Δx and Δt go to zero along a path where the fraction on the right side of the equation above is constant and get

$$\frac{\partial}{\partial t} P_k^t = \Delta_x P_k^t$$

So the population distribution evolves according to the diffusion equation. We have used a biological example here, but random walks are ubiquitous in science and engineering. We could have random walks with multiple spatial dimensions as well with a similar derivation. If some regions of the domain tend to attract or repel particles, we might get a Poisson problem in the limit.

3.4 The Laplace-Beltrami Operator

The Laplace-Beltrami operator has analogous properties to the Laplacian. We develop the Laplace-Beltrami operator in n dimensions, even though all computational examples in the paper are in two dimensions. We avoid the use of tensors in this derivation and instead focus on using the language of linear algebra.

The Derivative of a Matrix

We need to consider matrices whose elements are functions of several variables. This brings up the issue of how to represent the derivatives of these matrices. In the case of one independent variable, we can take the derivative of a matrix $M(t)$ by taking the derivative of each component of M and arranging these in a matrix where component k, j is the derivative of component k, j of $M(t)$. This is easy to visualize and write down.

Much more complicated is the case where M depends on several variables – we denote the number of variables by p. The total derivative of an $m \times n$ matrix depending on p variables, which we denote by $\mathcal{D}M$, (where M is the matrix), must be an element of $\mathcal{L}(\mathbb{R}^p, \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m))$. $\mathcal{L}(A, B)$ is the set of linear transformations taking linear space A to linear space B. We visualize this as a sort of vector of matrices, where each matrix is formed by taking the partial derivative of the original matrix with respect to one variable. We index the elements by $(\mathcal{D}M)_{ijk}$, where the ijk element is the ij element in the k matrix, or $\dfrac{\partial m_{ij}}{\partial \theta_k}$.

This operator takes elements of $\mathbb{R}^p$ to $\mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$ by multiplying each element of the vector in $\mathbb{R}^p$ by the corresponding matrix as though it is a scalar and adding the resulting matrices together to get a single $n \times m$ matrix. We denote this operation by

$$[w, \mathcal{D}M]\,.$$

This operation gives an $n \times m$ matrix where the i, j element is given by

$$\sum_{k=1}^{p} w_k (\mathcal{D}M)_{ijk}.$$

We think of this operation as multiplying the vector from above. Matrices and vectors with the correct dimensions can also be multiplied from the left or the right. We do this by multiplying the vector or matrix independently by each matrix in $\mathcal{D}M$ and keeping the new matrices in the vector. For example, if we multiply $\mathcal{D}M$ by the $m \times m$ matrix A, we get a new three-dimensional matrix N whose i, j, k component is given by

$$\sum_{l=1}^{m} (\mathcal{D}M)_{ilk} A_{lj}.$$

We denote this type of multiplication like normal matrix multiplication, $\mathcal{D}MA$ in this example.

We can commute this type of multiplication with the multiplication from above:

$$A\,[w, \mathcal{D}M]\,B = [w, A\mathcal{D}M]\,B = A\,[w, \mathcal{D}MB] = [w, A\mathcal{D}MB]\,.$$

We can readily differentiate the the inverse of a matrix:

$$\mathcal{D}(M^{-1}) = -M^{-1}\,(\mathcal{D}M)\,M^{-1}.$$

This can easily be seen from differentiating the identity

$$MM^{-1} = I$$

and solving for $\mathcal{D}(M^{-1})$.

The n-Dimensional Cross Product

Suppose we fix n vectors in $\mathbb{R}^{n+1}$. If we add any other element of $\mathbb{R}^{n+1}$ to this collection, we can use them to form an $(n+1) \times (n+1)$ matrix. The absolute value of the determinant of this matrix gives the volume spanned by the $n+1$ vectors. The determinant by itself could be thought of as the signed volume. The signed volume is a linear functional of the last vector we have added to the collection. By the Reisz representation theorem, there is a vector $\mathbf{A} \in \mathbb{R}^{n+1}$ such that $\mathbf{A} \cdot \mathbf{v}$ is the signed volume from adding $\mathbf{v}$ to the collection for all $\mathbf{v} \in \mathbb{R}^{n+1}$. Using the properties of determinants, we can easily see that $\mathbf{A}$ is zero unless our original n vectors are linearly independent and that $\mathbf{A}$ is orthogonal to each of the original n vectors. Finally, if $\mathbf{A}$ is nonzero, we see that the signed volume that results from adding $\frac{\mathbf{A}}{\|\mathbf{A}\|}$ to the collection is

$$\mathbf{A} \cdot \frac{\mathbf{A}}{\|\mathbf{A}\|} = \|\mathbf{A}\| .$$

Here, we have taken the volume spanned by a set of n vectors and a unit vector orthogonal to the first n vectors. This is also the area spanned by the first n vectors. We denote

$$\mathbf{A} = v_1 \times \ldots \times v_n.$$

Then $\|\mathbf{A}\|$ is the area spanned by $v_1, \ldots, v_n$. The n-dimensional cross product is a multilinear function.

A Closer Look at The First Fundamental Form

We now look at an arbitrary smooth, closed $n-$dimensional surface in $\mathbb{R}^{n+1}$. We assume that it is parametrized with a single smooth function $X : \Theta \to \Omega$, where $\Theta \in \mathbb{R}^n$ and Ω is the surface. X is a bijection from Θ onto Ω, and thus the variables in Θ give a coordinate system for Ω. At each point in Θ, we can take the Jacobian of X, J. If Ω is a non-degenerate surface, which we assume it is, then J is of rank n, and its columns are linearly independent. The columns of J give a basis for the tangent space of the surface at $X(\theta)$. We denote them by $\{x_1, \ldots, x_n\}$. Their cross product, which we denote by $\mathbf{N}$, is orthogonal to the surface at $X(\theta)$.

Now consider the singular value decomposition of J,

$$J = U \Sigma V^\top.$$

The diagonal elements of Σ are nonzero since J is of rank n. It should be noted from here that the columns of J are all linear combinations of the first n columns of U. The $(n+1)$ column of U is therefore a unit normal vector to the surface at $X(\theta)$, and we denote it by $\mathbf{n}$. We can form the matrix

$$L = U\Sigma^{+}V^{\top}$$

by inverting the nonzero elements of Σ. It is then easy to see that

$$L^{\top}J = J^{\top}L = I_n.$$

The columns of L are also linear combinations of the first n columns of U, and thus form the dual basis to the tangent space to the columns of J. We denote the columns of L by $\left\{x^1, \ldots, x^n\right\}$.

If we perform the multiplication $LJ^{\top}$, we get $I_{n+1} - \mathbf{n}\mathbf{n}^{\top}$.

We can now form the matrix

$$G = J^{\top}J.$$

The elements of G are the coefficients of the first fundamental form. The elements of G are denoted by G_{jk}.

As we have seen, G is important in measuring lengths and areas. For example, the square root of the determinant of G gives the area spanned by $\{x_1, \ldots, x_n\}$,

$$\sqrt{|G|} = \|\mathbf{N}\|.$$

We can see this by taking the determinant of the the matrix

$$\left(\begin{array}{c} J^{\top} \\ \hline N^{\top} \end{array}\right)\left(J \;\middle|\; N\right) = \left(\begin{array}{c|c} J^{\top}J & \mathbf{0} \\ \hline \mathbf{0}^{\top} & \|\mathbf{A}\|^2 \end{array}\right).$$

Using the properties of the cross product and of determinants, we see that the determinant of this matrix is $\|\mathbf{N}\|^4$. It is also $\det(J^{\top}J)\|\mathbf{N}\|^2$. Therefore, $\sqrt{\det G} = \|\mathbf{N}\|$.

Similarly, we can find

$$G^{-1} = L^{\top}L.$$

This matrix is also important in measuring lengths. We denote its elements by G^{jk}.

The Christoffel symbols and the Second Fundamental Form

We now consider the second derivatives of the parameterization X. We denote by $\mathbf{x}_{jk}$ the vector that results when we take the partial derivatives of the elements of X with respect to θ_j and then θ_k. Since the vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ form a basis for the tangent space of Γ, we can find constants such that

$$\mathbf{x}_{jk} = \sum_{l=1}^{n} \Gamma_{jk}^{l} \mathbf{x}_l + b_{jk}\mathbf{n}.$$

We can also find constants such that

$$\mathbf{x}_{jk} = \sum_{l=1}^{n} \Gamma_{jkl} \mathbf{x}^l + b_{jk}\mathbf{n}.$$

The matrix B formed from the coefficients b_{jk} is the second fundamental form. It gives information about the shape of the surface. We are not concerned with the second fundamental form here, but the other coefficients, called the Christoffel symbols, are of great importance.

The coefficients Γ_{ijk} can be calculated by

$$\Gamma_{ijk} = \mathbf{x}_{ij} \cdot \mathbf{x}_k.$$

They are called the Christoffel symbols of the first kind.

The coefficients Γ_{ij}^{k} can be calculated by

$$\Gamma_{ij}^{k} = \mathbf{x}_{ij} \cdot \mathbf{x}^k.$$

They are called the Christoffel symbols of the second kind.

Differentiating a function on the surface

We now consider a function f defined on the surface Γ. We also consider a curve on the surface, parametrized by $\theta(t)$. Here, $\theta(t)$ is a smooth function from a closed interval on the real line to Θ. Then $X(\theta(t))$ is a curve that is actually on the surface Γ in $\mathbb{R}^{n+1}$. We denote the function $\gamma(t) = X(\theta(t))$.

Then $F(t) = f(X(\theta(t))) = f(\gamma(t))$ is a real-valued function of one variable, and can be readily differentated. We also define the function $g : \Theta \to \mathbb{R}$ by $g(\theta) = f(X(\theta))$. Then, we also have $F(t) = g(\theta(t))$. If we could define a gradient for the function f, we would have

$$F'(t) = \nabla f \dot{\gamma}(t) = \nabla g \dot{\theta}(t).$$

It is clear that $\dot{\gamma}(t) = J\dot{\theta}(t)$, so the equation is satisfied if

$$\nabla f J = \nabla g.$$

We now require that ∇f be orthogonal to $\mathbf{n}$. This is reasonable since f is defined only on Γ. Then, we can multiply both sides of the equation by $L^\top$ to get

$$\nabla f = \nabla g L^\top.$$

This definition of ∇f lets the chain rule be used consistently and this does not change if we reparametrize the surface. The next step is to find rules for second derivatives of f. This leads to the Laplace-Beltrami operator. Heuristically, if we take total derivatives of the equation

$$\nabla f \dot{\gamma} = \nabla g \dot{\theta},$$

we get

$$\dot{\gamma}^\top H f \dot{\gamma} + \nabla f \ddot{\gamma} = \dot{\theta}^\top H g \dot{\theta} + \nabla g \ddot{\theta}$$

where H denotes the Hessian, or second derivative matrix. Here, we note that

$$\ddot{\gamma} = \left[\dot{\theta}, \mathcal{D}J\right] \dot{\theta} + J\ddot{\theta}.$$

We can now cancel the terms $\nabla f J\ddot{\theta} = \nabla g \ddot{\theta}$, since they tell nothing about the second derivatives of f, to get

$$\dot{\gamma}^\top H f \dot{\gamma} + \nabla f \left[\dot{\theta}, \mathcal{D}J\right] \dot{\theta} = \dot{\theta}^\top H g \dot{\theta}.$$

All of these terms except Hf are already well-defined, so we define Hf such that the equation holds. First, however, we consider the second term on the left hand side. We can write it as

$$\nabla f \left[\dot{\theta}, \mathcal{D}J\right] \dot{\theta} = \nabla g L^\top \left[\dot{\theta}, \mathcal{D}J\right] \dot{\theta} = \nabla g \left[\dot{\theta}, L^\top \mathcal{D}J\right] \dot{\theta}.$$

The multiplication $L^\top \mathcal{D}J$ gives Γ_2, a matrix of Christoffel symbols of the second kind. The i, j, k element of is $\mathbf{x}^i \cdot \mathbf{x}_{jk} = \Gamma^i_{jk}$. This gives

$$\nabla g \left[\dot{\theta}, \Gamma_2\right] \dot{\theta}.$$

We rewrite this quantity as

$$\dot{\theta} \left[\nabla g, \Gamma_2^*\right] \dot{\theta}$$

where the $*$ indicates that we have interchanged the first and third indices. (The more familiar transpose, by contrast, interchanges the first and second indices.)

If we bring this quantity to the other side and use the fact that $\dot{\gamma} = J\dot{\theta}$, we see that

$$\dot{\theta}^\top J^\top H f J \dot{\theta} = \dot{\theta}^\top H g \dot{\theta} - \dot{\theta}^\top \left[\nabla g, \Gamma_2^*\right] \dot{\theta}.$$

This equation must be satisfied for all choices of $\theta(t)$, so we drop the $\dot{\theta}$:

$$J^\top H f J = Hg - \left[\nabla g, \Gamma_2^*\right].$$

We note here that $[\nabla g, \Gamma_2^*] = \left[\nabla g, (G^{-1}\Gamma_1)^*\right] = \left[\nabla g G^{-1}, \Gamma_1^*\right]$,

so we have

$$J^\top H f J = Hg - \left[\nabla g G^{-1}, \Gamma_1^*\right].$$

We now add and subtract the quantity $\left[\nabla g G^{-1}, \Gamma_1^\top\right]$ on the right hand side:

$$J^\top H f J = Hg - \left[\nabla g G^{-1}, \Gamma_1^* + \Gamma_1^\top\right] + \left[\nabla g G^{-1}, \Gamma_1^\top\right].$$

The i, j, k element of $\Gamma_1^* + \Gamma_1^\top$ is given by

$$\left(\Gamma_1^* + \Gamma_1^\top\right)_{i,j,k} = \Gamma_{jik} + \Gamma_{ikj} = \mathbf{x}_{ji} \cdot \mathbf{x}_k + \mathbf{x}_j \cdot \mathbf{x}_{ki} = \frac{\partial G_{jk}}{\partial \theta_i},$$

so we have

$$J^\top H f J = Hg - \left[\nabla g G^{-1}, (\mathcal{D}G)^*\right] + \left[\nabla g G^{-1}, \Gamma_1^\top\right].$$

We can now make the adjustment

$$J^\top H f J = Hg - \left[\nabla g, (G^{-1}\mathcal{D}G)^*\right] + \left[\nabla g G^{-1}, \Gamma_1^\top\right].$$

Now, multiply both sides by G^{-1} from the right. Note that $JG^{-1} = JL^\top L = (I_{n+1} - \mathbf{n}\mathbf{n}^\top)L = L$ since the columns of L are orthogonal to $\mathbf{n}$, so

$$J^\top H f L = Hg G^{-1} \left[\nabla g, (G^{-1}\mathcal{D}G)^*\right] + \left[\nabla g G^{-1}, \Gamma_1^\top\right].$$

When a matrix is multiplied from the right, we can take it inside the *. Also, multiplying G^{-1} from the right of $\Gamma_1^\top$ gives $\Gamma_2^\top$. So we now have

$$J^\top H f L = Hg G^{-1} + \left[\nabla g, -(G^{-1}\mathcal{D}G G^{-1})^*\right] + \left[\nabla g G^{-1}, \Gamma_2^\top\right].$$

We now note that the three-dimensional matrix in the second term is the total derivative of G^{-1}, or $\mathcal{D}G^{-1}$. This means that the first two terms combine to give the total derivative of $\nabla g G^{-1}$,

$$J^\top H f L = \mathcal{D}(\nabla g G^{-1}) + \left[\nabla g G^{-1}, \Gamma_2^\top\right].$$

Therefore, we define

$$Hf = L\left(\mathcal{D}(\nabla g G^{-1}) + \left[\nabla g G^{-1}, \Gamma_2^\top\right]\right) J^\top$$

and the required conditions are satisfied. We return now to

$$J^\top H f L = \mathcal{D}(\nabla g G^{-1}) + \left[\nabla g G^{-1}, \Gamma_2^\top\right].$$

We consider the eigenvalues of the matrix. It is clear from the way we have defined Hf that it has a zero eigenvalue corresponding to **n**. If we augment **n** onto the ends of L and J, forming L_1 and J_1, then

$$L_1^\top H f J_1$$

is similar to Hf, and is in fact the matrix from the right hand side of the equation in the first n rows and columns and zero everywhere else. This shows that the matrix on the right hand side has the same eigenvalues except the zero eigenvalue corresponding to **n**, and therefore the same sum of eigenvalues, which is its trace. The Laplace-Beltrami operator is the sum of the eigenvalues:

$$\sum_{i=1}^{n}\left(\frac{\partial}{\partial\theta_i}\left(\sum_{j=1}^{n}\frac{\partial g}{\partial\theta_j}G^{ji}\right) + \sum_{j=1}^{n}\sum_{k=1}^{n}\frac{\partial g}{\partial\theta_j}G^{jk}\Gamma_{ik}^{i}\right).$$

We can simplify this expression by considering

$$\frac{\partial}{\partial\theta_i}|G|.$$

We have already shown

$$|G| = \mathbf{N}\cdot\mathbf{N}.$$

If we take the derivative with respect to θ_i, we get

$$2\mathbf{N}\cdot\frac{\partial}{\partial\theta_i}\mathbf{N}.$$

Remember that $\mathbf{N}$ is the n-dimensional cross product of the vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$, so we use the multilinearity of the cross product to get

$$\frac{\partial}{\partial \theta_i}\mathbf{N} = \left(\frac{\partial \mathbf{x}_1}{\partial \theta_i} \times \ldots \times \mathbf{x}_n\right) + \ldots + \left(\mathbf{x}_1 \times \ldots \times \frac{\partial \mathbf{x}_n}{\partial \theta_i}\right).$$

We can expand these derivatives in terms of the Christoffel symbols of the second kind and the second fundamental form:

$$\frac{\partial \mathbf{x}_k}{\partial \theta_i} = \sum_{j=1}^{n} \Gamma_{ki}^{j} \mathbf{x}_j + B_{ki}\mathbf{n}.$$

Now, for example, we have

$$\left(\frac{\partial \mathbf{x}_1}{\partial \theta_i} \times \ldots \times \mathbf{x}_n\right) = \sum_{j=1}^{n} \Gamma_{1i}^{j} \left(\mathbf{x}_j \times \ldots \times \mathbf{x}_n\right) + b_{1i}\left(\mathbf{n} \times \ldots \times \mathbf{x}_n\right) = \Gamma_{1i}^{1}\mathbf{N} + b_{1i}\left(\mathbf{n} \times \ldots \times \mathbf{x}_n\right)$$

since the cross product gives zero when the vectors are not linearly independent. The cross product involving $\mathbf{n}$ is orthogonal to $\mathbf{N}$, so it will also vanish when we take the dot product with $2\mathbf{N}$. The same thing happens for every other term in the original expansion of $2\mathbf{N} \cdot \frac{\partial}{\partial \theta_i}\mathbf{N}$, so we are left with

$$\frac{\partial}{\partial \theta_k}|G| = 2|G| \sum_{i=1}^{n} \Gamma_{ik}^{i}$$

$$\frac{\partial}{\partial \theta_k} \log \sqrt{|G|} = \sum_{i=1}^{n} \Gamma_{ik}^{i}.$$

This simplifies the Laplace-Beltrami operator to

$$\sum_{i=1}^{n} \frac{\partial}{\partial \theta_i}\left(\sum_{j=1}^{n} G^{ji}\frac{\partial g}{\partial \theta_j}\right) + \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial g}{\partial \theta_j} G^{ij} \frac{\partial}{\partial \theta_i} \log(\sqrt{|G|})$$

$$= \sum_{i=1}^{n}\left(\frac{\partial}{\partial \theta_i}\left(\sum_{j=1}^{n} G^{ji}\frac{\partial g}{\partial \theta_j}\right) + \frac{1}{\sqrt{|G|}}\sum_{j=1}^{n} \frac{\partial g}{\partial \theta_j} G^{ij} \frac{\partial \sqrt{|G|}}{\partial \theta_i}\right)$$

which is equal to

$$\frac{1}{\sqrt{|G|}}\left(\sum_{i=1}^{n} \sqrt{|G|}\left(\frac{\partial}{\partial \theta_i}\left(\sum_{j=1}^{n} G^{ji}\frac{\partial g}{\partial \theta_j}\right) + \frac{\partial |G|}{\partial \theta_i}\sum_{j=1}^{n} \frac{\partial g}{\partial \theta_j} G^{ij}\right)\right)$$

$$= \frac{1}{\sqrt{|G|}}\sum_{i=1}^{n} \frac{\partial}{\partial \theta_i}\left(\sum_{j=1}^{n} \sqrt{|G|} G^{ij}\frac{\partial g}{\partial \theta_i}\right). \quad (3.1)$$

This is the classic formula for the Lapace-Beltrami operator.

Inner Products of Gradients and the Divergence Theorem

We saw in the previous derivation that if u is a differentiable function defined on a surface Γ and parameterized with $\theta : \Theta \to \Gamma$, then $\nabla_\Gamma u = \nabla u L^\top$ is a gradient in $\mathbb{R}^{n+1}$ that is does not depend on the parameterization. If v is another differentiable function on Γ, we have, since $L^\top L = G^{-1}$,

$$\int_\Gamma \nabla_\Gamma u \cdot \nabla_\Gamma v d\sigma = \int_\Theta \nabla u G^{-1} \nabla v^\top \sqrt{|G|} d\theta.$$

We can see from this that the Laplace-Beltrami operator behaves like the Laplacian with respect to the divergence theorem and integration by parts when we integrate $-\Delta uv$ over Γ:

$$\begin{aligned}
\int_\Gamma -\Delta_\Gamma uv d\sigma &= -\int_\Theta \frac{1}{\sqrt{|G|}} \sum_{i=1}^n \frac{\partial}{\partial \theta_i} \left(\sum_{j=1}^n \sqrt{|G|} G^{ij} \frac{\partial u}{\partial \theta_i} \right) v \sqrt{|G|} d\theta \\
&= \int_\Theta \nabla u G^{-1} \nabla v^\top \sqrt{|G|} d\theta \\
&= -\int_\Theta \frac{1}{\sqrt{|G|}} \sum_{i=1}^n \frac{\partial}{\partial \theta_i} \left(\sum_{j=1}^n \sqrt{|G|} G^{ij} \frac{\partial v}{\partial \theta_i} \right) u \sqrt{|G|} d\theta \\
&= \int_\Gamma -\Delta_\Gamma vu d\sigma.
\end{aligned}$$

3.5 Special Cases

We now look at the Laplace-Beltrami operator in a few special cases.

Conformal Parameterization

As we have seen, the first fundamental form of a conformally parametrized surface in two dimensions is

$$\begin{pmatrix} E & 0 \\ 0 & E \end{pmatrix}.$$

In this case, the Lapace-Beltrami operator is

$$\Delta_\Gamma u = \frac{1}{E} \left(\frac{\partial^2 u}{\partial u^2} + \frac{\partial^2 u}{\partial v^2} \right).$$

The Laplace-Beltrami operator here is therefore proportional to the Laplacian.

Isometric Parameterization

We have seen that, under an isometric parameterization, the first fundamental form is simply the identity matrix. The Laplace-Beltrami operator is simply the Laplacian:

$$\Delta_\Gamma u = \frac{\partial^2 u}{\partial u^2} + \frac{\partial^2 u}{\partial v^2}.$$

We mentioned in the previous chapter that the Gaussian Curvature is zero under an isometric parameterization. This is a result of Gauss' Theorema Egregium, which says that the Gaussian Curvature is given by

$$K = \left(\frac{\partial}{\partial x}\Gamma_{yyx} - \frac{\partial}{\partial y}\Gamma_{yxx} + \Gamma^x_{xy}\Gamma_{xyx} + \Gamma^y_{xy}\Gamma_{xyy} - \Gamma^x_{yy}\Gamma_{xxx} - \Gamma^y_{yy}\Gamma_{xxy} \right) / \sqrt{|G|}$$

Since the first fundamental form is the identity matrix, it can easily be shown that all of the Christoffel Symbols are zero. The Gaussian curvature, therefore, is always zero.

A Surface Defined as a Level Set

When a surface is defined as a level set of some function $f(x, y, z)$, we can get a parameterization locally as described in the previous chapter. We have seen in this chapter, however, that on any surface, we have

$$JL^\top = I_{n+1,n+1} - \mathbf{n}^\top \mathbf{n}$$

and that our process of differentiating a function essentially entails defining it as a function of $n+1$ variables (3 in the case of a surface) and projecting out the unit normal vector from the resulting gradient. If our function f that defines the surface is a distance function, then ∇f is the unit normal vector. Thus, when we take the Laplace-Beltrami operator for a function u, we can extend u to a neighborhood of the surface. (Any extension works: the simplest method is to make it constant along the unit normal vector as in [12].) Then, we take the gradient and project out the unit normal vector. Differentiate the gradient and again project out the unit normal vector from each column. This gives the correct Hessian matrix and the trace of this matrix is the Laplace-Beltrami Operator.

3.6 Discussion

The Laplace-Beltrami operator is the correct generalization of the Laplacian to surfaces. It is the trace of the Hessian matrix when the Hessian Matrix is taken correctly to get distances and angles

correct. It is also the variation of the energy functional for the surface and satisfies the divergence theorem and Poisson Integral Formula for the surface.

4. NUMERICAL METHODS FOR THE POISSON PROBLEM ON A EUCLIDEAN DOMAIN

So far, we have reviewed some concepts from differential geometry necessary for extending the notion of the Laplacian to a closed surface. Our goal is to solve the Poisson Problem involving the Laplace-Beltrami operator on closed surfaces numerically. In order to prepare the background necessary for this, we consider how the Poisson Problem in Euclidean space is solved numerically.

4.1 Finite Element Methods

To solve the Poisson problem in Euclidean space, we use the finite element method. This arises from considering the variational form of the differential equation.

Problem 1 (Poisson Problem with Zero Dirichlet Boundary Conditions, Classical Form)**.** *Find* $u \in C^2(\bar{\Omega})$ *such that*

$$\begin{cases} -\Delta u = f(x), & x \in \Omega \\ u(x) = 0, & x \in \partial\Omega, \end{cases}$$

where Ω *is an open, bounded subset of* $\mathbb{R}^n$ *and* $f(x) \in C^0(\Omega)$*, either* $\partial\Omega$ *is smooth or* Ω *is convex and* $\partial\Omega$ *is piecewise polygonal, and* $C^k(\Omega)$ *denotes the set of all functions defined on* Ω *for which all derivatives up to order* k *exist and are continuous.*

The variational form is motivated by integration by parts. If u is a classical solution to the problem, we choose $v \in C_0^1(\Omega)$ ($C_0^1(\Omega)$ is the set of functions in $C^1(\Omega)$ with compact support.) Then,

$$\begin{aligned} \int_\Omega -\Delta u v dx = \int_{\partial\Omega} = \frac{\partial u}{\partial \nu} v d\sigma + \int_\Omega \nabla \cdot u \nabla v dx \\ = \int_\Omega \nabla u \cdot \nabla v dx \end{aligned}$$

The boundary term disappears because of the assumption we have made on v. Therefore, u satisfies

$$\int_\Omega \nabla u \cdot \nabla v dx = \int_\Omega f v dx, \forall v \in C_0^1(\Omega).$$

This is the main form of the variational formulation of the problem. We need a few more definitions before we can rigorously formulate the problem. We first define the space in which our analysis takes place.

Definition 14 ($L^2(\Omega)$)**.** *For an open, bounded, connected set $\Omega \subset \mathbb{R}^n$ where either $\partial\Omega$ is smooth or Ω is convex and $\partial\Omega$ is piecewise polygonal, $L^2(\Omega)$ is the set of measurable functions f defined on Ω such that the L^2 norm,*

$$\|f\|_{L^2(\Omega)} = \sqrt{\int_\Omega f(x)^2 dx},$$

is finite. We take this integral in the Lebesgue sense.

The space $L^2(\Omega)$ has some important properties for our purposes.

Lemma 1 (Important properties of $L^2(\Omega)$.)**.** $L^2(\Omega$ *has the following important properties:*

1. $L^2(\Omega)$ *is a linear space – if f, g are in $L^2(\Omega)$ and $a, b \in \mathbb{R}$, then $af(x) + bg(x) \in L^2(\Omega)$ and $0 \in L^2(\Omega)$.*

2. $L^2(\Omega)$ *has an inner product $(f, g)_{L^2(\Omega)} = \int_\Omega f(x)g(x)dx$.*

3. $L^2(\Omega)$ *is complete with respect to its norm – if $\{f_n\}_{n=1}^\infty$ is a Cauchy sequence in the L^2 norm of functions in $L^2(\Omega)$, then it converges to a function $f \in L^2(\Omega)$.*

These properties mean that $L^2(\Omega)$ is a Hilbert Space. One other important property of $L^2(\Omega)$ is the following.

Lemma 2. $C_0^\infty(\Omega)$ *is dense in $L^2(\Omega)$ – for every $f \in L^2(\Omega)$ and ϵ, there exists a $v \in C_0^\infty(\Omega)$ such that*

$$\|f - v\|_{L^2(\Omega)} < \epsilon.$$

We now consider the notion of a weak derivative. A function $g(x)$ is said to be a *weak derivative* of $f(x)$ on Ω if it satisfies

$$\int_\Omega f(x)v'(x)dx = -\int_\Omega g(x)v(x)dx$$

for all $v(x) \in C_0^\infty(\Omega)$. We use $\Omega \subset \mathbb{R}$ for the definition but we can readily expand the definition to any dimension using partial derivatives. It is clear that a classical derivative is also a weak derivative using integration by parts. We can define the set of functions $f \in L^2(\Omega)$ that have weak derivatives up to order k that are themselves in $L^2(\Omega)$.

Definition 15 ($H^k(\Omega)$)**.** *For any natural number k,*

$$H^k(\Omega) \subset L^2(\Omega)$$

is the set of all functions in $L^2(\Omega)$ for which all weak derivatives up to order k exist and are in $L^2(\Omega)$. This means that the $H^k(\Omega)$ norm,

$$\|f\|_{H^k(\Omega)} = \left(\int_\Omega \sum_{|\alpha| \leq k} (D^\alpha f)^2 \, dx \right)^{\frac{1}{2}},$$

is defined and finite. Here, α is a multi-index – $\alpha \in \mathbb{N}^n$, $D^\alpha f = \frac{\partial}{\partial x_1}^{\alpha_1} \cdots \frac{\partial}{\partial x_n}^{\alpha_n} f$ and $|\alpha| = \sum_{j=1}^n \alpha_j$.

We look for solutions to the variational problem in the Sobolev Spaces. In order to deal with boundary conditions, we define the trace of a function in $H^1(\Omega)$ as follows.

Theorem 5 (Trace Theorem)**.** *There exists a continuous, bounded, linear transformation $T : H^1(\Omega) \to L^2(\partial\Omega)$ such that*

$$Tf = f|_{\partial\Omega}$$

if $f \in C^1(\Omega)$. We call Tf the trace of f on $\partial\Omega$.

Using the trace, we can define the important space of functions $H_0^1(\Omega)$.

Definition 16 ($H_0^1(\Omega)$)**.** *$H_0^1(\Omega)$ is the subspace of $H^1(\Omega)$ with a trace of zero on $\partial\Omega$.*

We now have all the terminology we need to define the variational form of the Poisson problem.

Problem 2 (Poisson Problem with zero Dirichlet Boundary Condtions, Variational Form)**.** *Find $u \in H_0^1(\Omega)$ such that*

$$\left\{ \int_\Omega \nabla u \cdot \nabla v dx = \int_\Omega f v dx \quad \forall v \in H_0^1(\Omega). \right.$$

Remark 3. *A solution $u \in H_0^1(\Omega)$ is called a weak solution. A solution $u \in H^2(\Omega)$ is called a strong solution. We only need to require $f \in L^2(\Omega)$, rather than $C^0(\Omega)$, to have a unique weak solution. If the $\partial\Omega$ is smooth or if Ω is convex, then the weak solution is also a strong solution.*

$H_0^1(\Omega)$ is an infinite-dimensional vector space. The finite element method is a special case of the Galerkin method, which essentially projects the problem onto a finite-dimensional subspace of H_0^1.

Problem 3 (Finite-Dimensional Weak Problem)**.** *Find $U \in V$ such that*

$$a(U, v) = (f, v)_{L^2(\Omega)}$$

where

$$a(U, v) = \int_\Omega \nabla U \cdot \nabla v dx.$$

for all $v \in V$, where V is some finite-dimensional subspace of $H_0^1(\Omega)$.

Now, if $\{v_j\}_{j=1}^N$ is a basis for V, then we have $U = \sum_{j=1}^n U_j v_j$. We also have, for every k,

$$a(U, v_k) = \sum_{j=1}^n U_j a(v_j, v_k) = (f, v_k)_{L^2(\Omega)}.$$

We can now retrieve the coefficients U_j of U by solving the linear system

$$AU = B,$$

where A is the $n \times n$ stiffness matrix with $A_{jk} = a(v_j, v_k)$ and B is the $n \times 1$ vector where $b_j = (v_j, f)_{L^2(\Omega)}$.

A Finite Element Method is a special kind of Galerkin method where we use basis functions that are equal to zero except on a finite subset of the domain and consist of piecewise polynomial functions. Finite Element Methods have sparse, structured matrices and coefficients that are generally easy to recover.

Example 2. *Choose for V the span of tent functions. We triangulate the domain – choose points to be vertices and have edges connecting them. There is a basis function for each vertex. It is equal to 1 at that vertex and decreases linearly to zero at each neighboring vertex and is zero everywhere else. Since these functions have compact support, it leads to a sparse elasticity matrix. This is called the CG1 method. (CG stands for Continuous Galerkin.)*

4.2 Overview of Error

When we use a finite element method for a differential equation, we are approximating the solution rather than calculating it directly. Any time we do this, it is important to know whether we have a good approximation and to have an idea of how far our approximation may be from the true solution. There are two major approaches to the study of error: a priori error analysis and a posteriori error

analysis. A priori error analysis takes place before any approximations are actually computed and covers a large class of problems. It provides information on the rate of convergence as the mesh is refined. A posterori error analysis, however, takes place after an approximation has been computed and is specific to the problem in which the approximation was computed. A priori error analysis gives us general error bounds that tell that a general method is sound and yields a more accurate solution if more work is done. A posteriori error analysis gives an estimate of how far a particular calculation is from the true value.

Example 3 (A priori error bound)**.** *For the Poisson problem*

$$\begin{cases} -\Delta u = f(x), & x \in [0,1] \times [0,1] \\ u(x) = 0, & x \in \partial\,[0,1] \times [0,1]\,. \end{cases}$$

If U is a finite element solution to the problem and $f \in L^2$, then there exists a C such that

$$\|u - U\|_{L^2(\Omega)} \leq C \|f\|_{L^2(\Omega)} h^2$$

where h is the diameter of the triangulation. This means that as we refine the triangulation, we get a better approximation with the convergence occurring at a quadratic rate.

4.3 Adjoint-Based A Posterori Error Estimate

When we are solving a differential equation like the Poisson problem, we are often not interested in the entire solution, but only in some aspect of the solution. We call this aspect of the solution the quantity of interest and when we are interested in one, a posterori error analysis should focus on error in this quantity. This is the approach taken in [16] and [17]. This allows us to have more efficient estimates and avoid unnecessary effort on carefully resolving parts of the solution that may not be important to the application for which the problem is being solved.

A key definition for this analysis is the definition of the dual space.

Definition 17 (Dual Space)**.** *If V is a linear space, its Dual Space V^* is the set of all continuous linear functionals (maps to $\mathbb{R}$) defined on V.*

The dual space is itself a vector space and every linear space has a dual space.

When the quantity of interest is a linear functional of the solution, the adjoint gives a convenient method for doing a posteriori error analysis. In general, we assume we are solving

$$Au = f$$

for some linear operator A, where u and f are elements of linear spaces V_1 and V_2 respectively with $A : V_1 \to V_2$. In the case of the Variational Poisson Problem with zero Dirichlet Boundary Conditions, $A = -\Delta$, $V_1 = H_0^1(\Omega)$ and $V_2 = L^2(\Omega)$. We also assume that we want to know a certain quantity of interest denoted by q such that

$$q = (u, \psi),$$

where ψ is an element of the dual space V_1^* of V_1. The notation (u, ψ) means that we are evaluating the linear functional ψ at u. In the case of the Poisson Problem with zero Dirichlet Boundary Conditions, we assume $\psi \in L^2(\Omega)$ and $(,)$ is the $L^2(\Omega)$ inner product.

Using the notion of a dual space, we define the adjoint operator.

Definition 18 (Adjoint Operator)**.** *If A is a linear transformation from the vector space V_1 to the vector space V_2, take $\psi \in V_2^*$. Then,*

$$L(u) = (Au, \psi), u \in V_1$$

is a linear functional on V_1, which we denote by $A^*\psi$. We call A^* the adjoint operator to A. It is a linear transformation from V_2^* to V_1^*.

Taking the adjoint of a linear operator is commutative with taking the inverse.

Lemma 3.

$$\left(A^{-1}\right)^* = (A^*)^{-1}$$

Proof. If $A : V_1 \to V_2$ is invertible, consider $u \in V_1$, $\phi \in V_1^*$. Then

$$\begin{aligned}\left(u, \left(A^{-1}\right)^* A^*\phi\right) &= \left(A^{-1}u, A^*\phi\right) \\ &= \left(AA^{-1}u, \phi\right) \\ &= (u, \phi).\end{aligned}$$

The claim holds since the choice of u and ϕ was arbitrary. □

Returning to our analysis, we can solve an adjoint problem

$$A^*\phi = \psi$$

where A^* is the adjoint to the linear operator A. Then, by definition,

$$q = (\psi, u) = \left(\psi, A^{-1} f\right) = \left((A^{-1})^* \psi, f\right) = (\phi, f)\,.$$

A useful fact about the adjoint solution is that it does not depend on the data f. If f is perturbed by a term ϵ, we can see that ϕ indicates the sensitivity to the perturbation.

$$q(f + \epsilon) - q(f) = (\phi, f + \epsilon) - (\phi, f) = (\phi, \epsilon)\,.$$

In the Poisson problem, the linear operator A is the Laplacian, which is a self-adjoint operator so $A^* = A$. We can see this using integration by parts. If $u, v \in H^2(\Omega) \cap H_0^1(\Omega)$, then

$$(-\Delta u, v)_{L^2(\Omega)} = a\,(u, v) = (u, \Delta v)_{L^2(\Omega)}\,.$$

Now let U be a finite element solution, i.e. we have

$$a\,(U, v) = (f, v)_{L^2(\Omega)}$$

for any function v in the span of the tent functions we have described. In particular, if $\pi_h \phi_h$ is a projection of ϕ into the space, such as the nodal interpolant, we have

$$(\nabla U, \nabla \pi_h \phi_h) = (f, \pi_h \phi_h)\,.$$

This is known as Galerkin orthogonality.

The true value of the quantity of interest is given by

$$q = (\psi, u)_{L^2(\Omega)}\,.$$

We are approximating the quantity of interest by using U instead of u. We get

$$\hat{q} = (\psi, U)_{L^2(\Omega)}\,.$$

The error in our approximation of the quantity of interest is given by

$$(\psi, u - U)_{L^2(\Omega)}\,.$$

Using the properties of u, U, ψ and ϕ we can expand this to get

$$(\psi, u - U)_{L^2(\Omega)} = (\phi, f)_{L^2(\Omega)} - a\,(\phi, U) = ((\phi - \pi_h \phi)\,, f)_{L^2(\Omega)} - a\,((\phi - \pi_h \phi)\,, U)\,.$$

This gives a computable estimate for the error in the quantity of interest, provided we compute or approximate the adjoint solution ϕ. In practice, we use a different mesh or a different basis, such as CG2, to calculate the adjoint solution ϕ in order to evaluate $\phi - \pi_h\phi$ in the Galerkin Orthogonality. Once we have ϕ, every term in the estimate above is computable.

Accurate a posteriori error estimates are useful for adaptive mesh refinement. We can calculate the above formula separately over each triangle in the mesh. On each triangle S_k, we get

$$e_{S_k} = \int_{S_k} (\phi - \pi_h\phi) f dx - \int_{S_k} \nabla (\phi - \pi_h\phi) \cdot \nabla U dx.$$

The sum of all of the integrals gives the total error. When we look at the integrals for the individual triangles, however, we can see what parts of the mesh contribute most to the error and therefore need to be resolved more carefully. In adaptive mesh refinement, we use this information to refine the mesh until the error is within an acceptable range.

5. SOLVING THE POISSON PROBLEM ON A SURFACE

So far we have discussed some important notions from differential geometry, culminating in the extension of the Laplacian from Euclidean space to the Laplace-Beltrami Operator on a surface. We have also discussed the numerical solution of the Possion problem on a Euclidean space and a practical a posteriori method for estimating error in the solution. In this chapter, we show how to extend the finite element method to the Poisson problem on a two dimensional surface and extend the error estimate to this problem. To define the finite element problem, we follow the analysis in [12] and [11].

5.1 Statement of Problem

Problem 4 (Diffusion on a Surface)**.** *On the closed, bounded, smooth surface* Γ*, find* $u(x,t)$ *with continuous derivative with respect to time and continuous second derivatives for the spatial variables such that*

$$\begin{cases} u_t - \Delta_\Gamma u = f(x,t) & x \in \Gamma, t \in [0,T] \\ u(x,0) = g(x). \end{cases}$$

The Laplace-Beltrami operator on the surface is analagous to the Laplacian in Euclidean space. If f does not depend on t then we have a steady-state problem.

Problem 5 (Steady-State of Diffusion Problem)**.** *If the problem above reaches a steady state, it satisfies*

$$-\Delta_\Gamma u = f(x).$$

Once a finite element method and error estimate have been established for the steady-state problem, it can be readily extended to the diffusion problem following the same approach as [16]. We therefore focus on the steady-state problem here. Unfortunately, the Steady-State Diffusion Problem as stated above is ill-posed on a closed surface.

Lemma 4. *The Steady-State Diffusion Problem has no solution unless*

$$\int_\Gamma f d\sigma = 0.$$

Proof. If u is a weak solution of the problem, we can take $v = 1$ as a test function in $H^2(\Gamma)$ to get

$$\int_\Gamma f d\sigma = (f, v)_{L^2(\Gamma)} = (\nabla_\Gamma u, \nabla_\Gamma v)_{L^2(\Gamma)} = 0.$$

□

This is to be expected because of the physical meaning of the problem. We have diffusion of heat or of a chemical under an outside forcing term. The requirement that $\int_\Gamma f d\sigma = 0$ means that heat, for example, is being introduced to the surface at the same rate it is being taken away from the surface. If this is not the case, then it would be impossible for heat on the surface to reach a steady state. Besides the uniqueness issue, we also have

Lemma 5. *A solution to the Steady-State Diffusion Problem is not unique.*

This is evident from the fact that the Laplacian of any constant function is zero.

To create a problem that has a unique solution, we choose the function $f(x)$ carefully and take the solution $u(x)$ that satisfies $\int_\Gamma u(x) d\sigma = 0$. However, a much simpler expediency is to instead consider an alternate problem.

Problem 6 (Alternate Steady-State Problem)**.** *Find u such that*

$$-\Delta_\Gamma u + u = f(x).$$

The modified problem has a unique solution for any $f \in L^2(\Gamma)$. This follows from a standard variational proof. The alternate problem is thus free of the complications of the original problem and does not introduce any new computational or theoretical issues. This problem is also important on its own. As in the Euclidean case, we proceed first by formulating the weak problems.

Problem 7 (Steady-State Problem, Variational Formulation)**.** *Find $u \in H^2(\Gamma)$ satisfying*

$$\left\{ a_\Gamma(u, v) = (f, v)_{L^2(\Gamma)}, \quad \forall v \in H^2(\Gamma) \right.$$

Note that in the case of a closed surface, we have no boundary. This means there is no boundary term in the integration by parts,

$$\int_\Gamma -\Delta_\Gamma u v d\sigma = \int_\Gamma \nabla_\Gamma u \cdot \nabla_\Gamma v d\sigma.$$

We also look for solutions in $H^2(\Omega)$ immediately rather than solutions in $H_0^1(\Omega)$. This is because the proof of improved regularity in the Euclidean case relies on the regularity of the boundary only for points near the boundary. In the case of a closed surface, there is no boundary and all points are interior points.

We define the bilinear forms on Γ

$$a_\Gamma(f,g) = \int_\Gamma \nabla_\Gamma f \cdot \nabla_\Gamma g d\sigma$$

and

$$(f,g)_L^2(\Gamma) = \int_\Gamma f g d\sigma.$$

The gradients used above are the gradients in the three-dimensional sense as described in chapter three.

The alternate problem is defined the same way.

Problem 8 (Alternate Steady-State Problem, Weak Formulation)**.** *Find* $u \in H^2(\Gamma)$ *such that*

$$\left\{ a_\Gamma(u,v) + (u,v)_{L^2(\Gamma)} = (f,v)_{L^2(\Gamma)}, \quad \forall v \in H^2(\Gamma). \right.$$

As before, we have $L^2(\Gamma)$ as the set of all square-integrable functions defined on Γ. We can define weak derivatives on Γ in an analogous fashion to their definition in Euclidean space, $g(x)$ is the weak partial derivative with respect to x_k of $f(x)$ on Γ if

$$(g,v)_{L^2(\Gamma)} = -\left(f, \frac{\partial v}{\partial x_k}\right)_{L^2(\Gamma)}$$

for every $v \in C_0^\infty(\Gamma)$. $H^2(\Gamma)$ is the set of L^2 functions on Γ with weak first- and second-order derivatives that are in $L^2(\Gamma)$. The partial derivatives of functions v and the inner product must be computed as in chapter 3.

We want to be able to deal with cases where the surface is very complicated and derivatives are expensive to compute. In order to do this efficiently, we need a simpler surface. We therefore build a finite element method using an approximate surface.

Definition 19 (Approximate Surface)**.** *If* Γ *is a smooth, closed, two-dimensional surface, its approximating surface* Γ_h *is a piecewise linear surface defined by nodes on* Γ*. We call* Γ_h *a triangulation of* Γ*.*

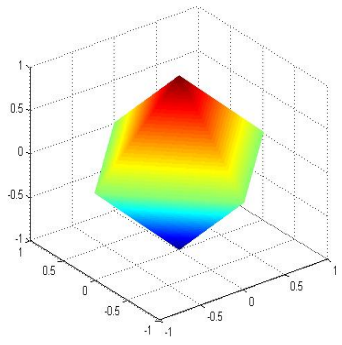

Fig. 5.1: Crude mesh for a sphere

Example 4. *For the unit sphere, the triangulation pictured in figure 5.1 with nodes at* $(1,0,0)$*,* $(-1,0,0)$*,* $(0,1,0)$*,* $(0,-1,0)$*,* $(0,0,1)$ $(0,0,-1)$ *is a very simple approximating surface.*

Once we have an approximate surface, we need a map between the true surface and the approximate surface.

Definition 20 (Lift and Projection)**.** *If we have* Γ *and a sufficiently fine* Γ_h *(as explained in [12]), then we define the projection*

$$P : \Gamma \to \Gamma_h$$

for every $x \in \Gamma$ *as follows. If* W *is a sufficiently small neighborhood in* $\mathbb{R}^3$ *of* x*, then there is a unique* $y \in \Gamma_h \cap U$ *lying on the line defined by* x *and* $\boldsymbol{n}$*, the unit normal vector to* Γ *at* x*.*
The lift

$$l : \Gamma_h \to \Gamma$$

is the inverse of the projection.

Remark 4. *In [11], it is shown that in order to ensure the uniqueness of the lift and projection, the distance* d *between* x *and* $P(x)$ *must satisfy*

$$d \leq \left(\min_{i=1,2} \|\kappa_i\|_{L^\infty(\Gamma)} \right)^{-1},$$

where κ_1, κ_2 *are the principle curvatures of* Γ*.*

Once we have this map, we can use functions defined on Γ to define functions on Γ_h and use functions defined on Γ_h to define functions on Γ.

Definition 21 (Lifted and Projected Functions)**.** *If u is a function defined on Γ_h, we define its projection $\hat{u}(x)$ on Γ by*

$$\hat{u}(x) = u(P(x)).$$

If u is a function defined on Γ, we define its lift $u^l(x)$ on Γ_h by

$$u^l(x) = u(l(x)).$$

When we move between the surfaces, we need to take into account the difference in area between them.

Definition 22 (Surface Measure)**.** *The surface measure μ_h is a function defined on Γ_h by*

$$\mu_h = \sqrt{\frac{\det(G(l(x)))}{\det(G_h(x))}}$$

where G_h is the First Fundamental Form of Γ_h at x.

Using the surface measure, we can see that u^l converges to u in something resembling an L^2 sense:

$$\begin{aligned}
\|u\|^2_{L^2(\Gamma)} - \left\|u^l\right\|^2_{L^2(\Gamma)} &= \int_\Gamma u^2 d\sigma - \int (u^l)^2 d\sigma_h \\
&= \int_{\Gamma_h} \left(\left(u^l\right)^2 \mu_h - \left(u^l\right)^2 \right) d\sigma_h \\
&\leq \left\|u^l\right\|^2_{L^2(\Gamma_h)} \int_{\Gamma_h} (\mu_h - 1)\, d\sigma_h.
\end{aligned}$$

The surface measure μ_h converges to 1 with order h^2 because of the approximate surface that we have chosen. The integral of μ_h over Γ is the ratio of surface areas between the surface Γ and its piecewise affine interpolant. The convergence is a property of piecewise linear interpolants as in [12].

Our plan is to use a finite element method using basis functions defined on Γ_h. We can then define U on Γ_h as well. By lifting U to Γ from Γ_h, we achieve an approximate solution in $H^2(\Gamma)$. To do this, we first need to choose a function on Γ_h to serve as the right hand side.

Definition 23 (f_h)**.** *The function f_h is defined on Γ_h and serves as the right-hand side for the differential equation on Γ_h. One possible choice is*

$$f_h(x) = f^l(x).$$

This satisfies the critical requirement that f_h converge to f with order h^2 as Γ_h converges to Γ.

We next need a space of functions defined on Γ_h. For a CG1 method, we use piecewise affine functions dependent on the nodes of the triangulation Γ_h, i.e. for each node x_j of Γ_h, we let $v_j(x)$ be the piecewise affine function such that $v_j(x_j) = 1$ and $v_j(x_k) = 0$ for all other nodes x_k. From this, we have the finite element problem.

Problem 9 (Finite Element Method)**.** *Find $U = \sum_{j=1}^{N} U_j v_j$ on Γ_h such that*

$$a_{\Gamma_h}(U, v_j) = (v_j, f_h)_{L^2(\Gamma_h)}$$

for every j in the case of the Poisson problem and

$$a_{\Gamma_h}(U, v_j)_{\Gamma_h} + (U, v_j)_{L^2(\Gamma_h)} = (v_j, f_h)_{L^2(\Gamma_h)}$$

for every j in the case of the alternate problem.

In [12], the method is analyzed in depth. It is shown to converge with order 2,

$$\left\|u - U^l\right\|_{L^2(\Gamma)} + h\left|u - U^l\right|_{H^1(\Gamma)} \leq Ch^2,$$

where

$$\left|u - U^l\right|_{H^1(\Gamma)} = \sqrt{a_\Gamma(u - U^l)}.$$

The proof depends first on the triangulation Γ_h. It converges to Γ with order h^2 in the L^∞ norm. As it converges, the surface measure μ_h converges to one and the lift and projection operators converge to isometric parmaterizations. We need to choose f_h so that $\left\|f - \hat{f}_h\right\|_{L^2(\Gamma)} \leq Ch^2 \|f\|_{L^2(\Gamma)}$.

Using the Adjoint

As we did in the Euclidean case, we want to use the adjoint to analyze the error in the solution. The Laplace-Beltrami operator, like the Laplacian, is self adjoint since for any $f, g \in H^2(\Omega)$,

$$(f, -\Delta_\Gamma g)_{L^2(\Gamma)} = -\int_\Gamma f \Delta_\Gamma g d\sigma = \int_\Gamma \nabla_\Gamma f \cdot \nabla_\Gamma g d\sigma = -\int_\Gamma \Delta_{\Gamma_h} f g d\sigma = (-\Delta_\Gamma f, g)_{L^2(\Gamma)}.$$

The Laplace-Beltrami operator on the approximating surface Γ_h is also self adjoint using the same argument. The non-Euclidean case of the Poisson Problem is more difficult to compute the error estimate for, however. Before, we had a single transformation that took the input f to the solution U.

$$f \xrightarrow{-\Delta^{-1}} U \xrightarrow{q} \mathbb{R}.$$

Now, however, we have another transformation:

$$f \xrightarrow{l} f_h \xrightarrow{-\Delta_h^{-1}} U \xrightarrow{q} \mathbb{R}.$$

Both transformations are subject to error and both require an adjoint solution.

Suppose we have our solution $u(x)$ defined on Γ. We again assume that we are interested in a particular quantity of interest q and that it is a linear functional of the solution. We write

$$q = \int_\Gamma u(x)\psi(x)d\sigma$$

where $\psi(x)$ is the Reisz representor for the quantity of interest q.

Definition 24 ($\psi_h(x)$)**.** *If we have $\psi(x)$, then $\psi_h(x)$ is defined on Γ_h and analogous to $f_h(x)$. It serves as the right-hand side for a Poisson problem formulated on Γ_h and should converge to ψ in the same sense that f_h converges to f.*

We can think of $\psi_h(x)$ as a representor on Γ_h of $\psi(x)$. Using $\psi(x)$ and the lift operation l, we have a linear functional L on $L^2(\Omega)$ defined by

$$L(g) = \int_\Gamma g^l \psi(x)d\sigma.$$

Using the standard Reisz argument, there must be a function $\psi_h(x) \in L^2(\Gamma_h)$ such that

$$L(g) = (\psi_h(x), g(x))_{\Gamma_h}.$$

We use $\psi_h(x)$ for the first adjoint problem that we consider. This problem is analogous to the adjoint problem for the Euclidean case.

Problem 10 (First Adjoint Problem)**.** *The first adjoint problem is on Γ_h,*

$$-\Delta_{\Gamma_h}\phi_h + \phi_h = \psi_h.$$

The second adjoint problem is, as discussed, needed because of the transformation from $H^2(\Gamma)$ to $H^2(\Gamma_h)$ that we make to compute U.

Problem 11 (Second Adjoint Problem)**.** *The second problem is on Γ,*

$$-\Delta_\Gamma\phi + \phi = \psi.$$

With the two adjoint solutions defined, we can begin to build an error estimate.

Definition 25 (Error in Quantity of Interest)**.** *The error in the quantity of interest is given by*

$$e = (\psi, u)_{L^2(\Gamma)} - (\psi_h, U)_{L^2(\Gamma_h)}$$

where we use the standard L^2 inner products:

$$(\psi, u)_{L^2(\Gamma)} = \int_\Gamma \psi u d\sigma$$

and

$$(\psi_h, U)_{L^2(\Gamma_h)} = \int_{\Gamma_h} \psi_h U d\sigma_h.$$

An important difference between this estimate and the estimate for the Euclidean case should be stressed. Before, u and U were both in $H^2(\Omega)$. Here, they are not in the same space or defined on the same domain. The lift and projection operators are crucial to understanding how the exact and approximate solutions relate to each other.

Using integration by parts and the definitions of the forward and adjoint solutions, we can get immediately

$$\begin{aligned}(\psi, u)_{L^2(\Gamma)} &= (-\Delta_\Gamma \phi + \phi, u)_{L^2(\Gamma)} \\ &= (\nabla_\Gamma \phi, \nabla_\Gamma u)_{L^2(\Gamma)} + (\phi, u)_{L^2(\Gamma)} \\ &= (f, \phi)_{L^2(\Gamma)}.\end{aligned}$$

The second term in the error formula gives, by definition and integration by parts,

$$\begin{aligned}(\psi_h, U)_{L^2(\Gamma_h)} &= (-\Delta_{\Gamma_h} \phi_h + \phi_h, U)_{L^2(\Gamma_h)} \\ &= (\nabla_{\Gamma_h} \phi_h, \nabla_{\Gamma_h} U)_{L^2(\Gamma_h)} + (\phi_h, U)_{L^2(\Gamma_h)}.\end{aligned}$$

Now, after some careful addition and subtraction, we can rewrite the error as

$$e = \left(f, \phi - \phi_h^l\right)_{L^2(\Gamma)} + \left(\hat{f}\mu_h - f_h, \phi_h\right)_{L^2(\Gamma_h)} + (f_h, \phi_h)_{L^2(\Gamma_h)} - (\nabla_{\Gamma_h} \phi_h, \nabla_{\Gamma_h} U)_{L^2(\Gamma_h)} - (\phi_h, U)_{L^2(\Gamma_h)}.$$

Remember that U satisfies Galerkin Orthogonality – i.e. for any function $v = \sum_{j=1} c_j v_j(x)$ where the functions $v_j(x)$ are the basis functions described above,

$$a_{\Gamma_h}(U, v) + (U, v)_{L^2(\Gamma_h)} - (f, v)_{L^2(\Gamma_h)} = 0.$$

For this reason, we consider the nodal interpolant of ϕ_h.

Definition 26 (Interpolant)**.** *If $c(x)$ is any function defined on Γ_h, then its interpolant*

$$\pi_h c(x)$$

is the piecewise affine function that takes the same values as $c(x)$ at the nodes of Γ_h.

We use Galerkin Orthogonality to write

Lemma 6.

$$(\nabla_{\Gamma_h} U, \nabla_{\Gamma_h} \pi_h \phi_h)_{L^2(\Gamma_h)} + (\pi_h \phi_h, U)_{L^2(\Gamma_h)} = (f_h, \phi_h)_{L^2(\Gamma_h)} .$$

The Galerkin Orthogonality represents the cancellation of errors that gives the accuracy in a Galerkin approximation. It is useful in the a posteriori error estimate for better "localizing" element contributions to the error. From this we get our formula for the a posterori error estimate.

Theorem 6 (A Posteriori Error)**.** *We can estimate the a posteriori error in the computation of the quantity of interest ψ from the finite element solution U by*

$$\begin{aligned} e = &\left(f, \phi - \phi_h^l\right)_\Gamma + \left(\hat{f}\mu_h - f_h, \phi_h\right)_{L^2(\Gamma_h)} \cdots \\ &+ (f_h, \phi_h - \pi_h \phi_h)_{L^2(\Gamma_h)} - \left(\nabla_{\Gamma_h} \left(\phi_h - \pi_h \phi_h\right), \nabla_{\Gamma_h} U\right)_{L^2(\Gamma_h)} - (\phi_h - \pi_h \phi_h, U)_{L^2(\Gamma_h)} . \end{aligned}$$

The first term is the difference between the second adjoint solution and the lift of the first one, integrated against the data f on Γ. The second term is the difference between f_h and the projection of f multiplied by the surface measure μ_h integrated against the first adjoint solution on Γ_h. Together, these terms give the geometric error – error resulting from the distortion of lengths, areas and angles when we approximate the surface Γ with Γ_h as well as error in the evaluation of f and ψ that arises because of this approximation. The remaining terms give the discretization error – error that results from solving the problem

$$-\Delta_{\Gamma_h} U + U = f_h$$

with a finite-dimensional function space. It uses the Galerkin Orthogonality property of U and is analogous to the error estimate that we had in the Euclidean case. We look at the terms in more depth.

Definition 27 (Geometric Error)**.** *Geometric error is error in the solution that arises due to approximating the surface. In the estimate above, the terms corresponding to geometric error are*

$$e_g = \left(f, \phi - \phi_h^l\right)_\Gamma + \left(\hat{f}\mu_h - f_h, \phi_h\right)_{L^2(\Gamma_h)} .$$

The first of these terms is the difference between the adjoint solutions. Remember that the quantity of interest is given by

$$(u, \psi)_\Gamma = (\nabla_\Gamma u, \nabla_\Gamma \phi)_{L^2(\Gamma)} + (u, \phi)_{L^2(\Gamma)} = (f, \phi)_{L^2(\Gamma)} .$$

This means that we can think of this term as measuring the difference in computing the quantity of interest with ϕ and with ϕ_h. This indicates how well Γ has been approximated and how close the data ψ_h for ϕ_h is to being the representor of ψ on Γ_h.

The second term is the difference between the data f and its projection $\hat{f}$ on Γ_h. The functions f and f_h are different from ψ and ψ_h because in practice, f is something that is observed and measured while ψ and ψ_h are defined and not necessarily observed from an outside source. This second term, therefore, tells us how our measurements of f are being influenced by approximating the surface. The difference is integrated against the first adjoint solution ϕ_h, so the term only says how much the variation of f affects the quantity of interest.

We can also note from this estimate that an interesting choice for f_h is

$$f_h = f^l \mu_h$$

This appears to eliminate the second term in the estimate. Evaluating f^l and computing μ_h on a complicated surface, however, may be difficult to do accurately.

Definition 28 (Discretization Error)**.** *Discretization error is error that arises from discretizing the Laplace-Beltrami operator. The terms corresponding to discretization error are*

$$e_d = (f_h, \phi_h - \pi_h \phi_h)_{L^2(\Gamma_h)} - (\nabla_{\Gamma_h} (\phi_h - \pi_h \phi_h), \nabla_{\Gamma_h} U)_{L^2(\Gamma_h)} - (\phi_h - \pi_h \phi_h, U)_{L^2(\Gamma_h)} .$$

Quadrature Error

All of the integrals for the finite element method are evaluated using numerical quadrature. Our implementations use Gaussian Quadrature, but other methods may also be used. Regardless of what numerical quadrature is used, the effect is to give solutions U, ϕ and ϕ_1 that satisfy approximate inner products, rather than the true H_1 inner products. We denote the approximate inner products by

$$(g_1, g_2)_{\Gamma_h, q} = \sum_{k=1}^{L} \sum_{j=1}^{n} w_j g_1(x_j) g_2(x_j) A(S_k),$$

where we are summing over all simplices of Γ_h, $A(S_k)$ is the area of the simplex S_k, the points x_j are the quadrature points and the numbers w_j are the quadrature weights. In our solutions, we use third-order Gaussian quadrature on each simplex, so

$$(g_1, g_2)_{\Gamma_h,q} = \sum_{k=1}^{L} \sum_{j=1}^{3} \left(\frac{1}{3}\right) g_1(e_{j,m}) g_2(e_{j,m}) A(S_k),$$

where $e_{j,m}$ indicates the midpoint of edge j. This means that rather than U satisfying

$$(U, \phi)_{L^2(\Gamma_h)} + (\nabla_h U_h, \nabla_h \phi_h)_{L^2(\Gamma_h)} = (f_h, \phi_h)_{L^2(\Gamma_h)},$$

it satisfies

$$(U_h, \phi_h)_{\Gamma_h,q} + (\nabla_{\Gamma_h} U_h, \nabla_{\Gamma_h} \phi_h)_{\Gamma_h,q} = (f_h, \phi_h)_{\Gamma_h,q}.$$

This means that we do not have true Galerkin Orthogonality. In order to use Galerkin Orthogonality in the error estimate, we must add and subtract carefully. We begin our previous estimate before we introduced Galerkin Orthogonality:

$$e = \left(f, \phi - \phi_h^l\right)_{L^2(\Gamma)} + \left(\hat{f}\mu_h - f_h, \phi_h\right)_{L^2(\Gamma_h)} + (f_h, \phi_h)_{L^2(\Gamma_h)} - (\nabla_{\Gamma_h}\phi_h, \nabla_{\Gamma_h} U)_{L^2(\Gamma_h)} - (\phi_h, U)_{L^2(\Gamma_h)}.$$

We would like to subtract $\pi_h\phi_h$ from the appropriate terms as before, but we do not have Galerkin orthogonality so we must subtract and add:

$$\begin{aligned} e = &\left(f, \phi - \phi_h^l\right)_{L^2(\Gamma)} + \left(\hat{f}\mu_h - f_h, \phi_h\right)_{L^2(\Gamma_h)} + (f_h, \phi_h - \pi_h\phi_h)_{L^2(\Gamma_h)} - \left(\nabla_{\Gamma_h}\left(\phi_h - \pi_h\phi_h, U\right)\right)_{L^2(\Gamma_h)} \\ &\ldots - (\phi_h - \pi_h\phi_h, u)_{L^2(\Gamma_h)} + (f_h, \pi_h\phi_h)_{L^2(\Gamma_h)} - (\nabla_{\Gamma_h}\pi_h\phi_h, U)_{L^2(\Gamma_h)} - (\pi_h\phi_h, U)_{L^2(\Gamma_h)}. \end{aligned}$$

We can now add in the approximate inner products using quadrature to get the following error estimate.

Theorem 7 (Error Estimate with Quadrature)**.** *The error in the quantity of interest when U is computed using numerical quadrature is*

$$\begin{aligned} e = &\left(f, \phi - \phi_h^l\right)_{\Gamma} + \left(\hat{f}\mu_h - f_h, \phi_h\right)_{L^2(\Gamma_h)} + \ldots \\ &(f_h, \phi_h - \pi_h\phi_h)_{L^2(\Gamma_h)} - \left(\nabla_{\Gamma_h}\left(\phi_h - \pi_h\phi_h\right), \nabla_{\Gamma_h} U\right)_{L^2(\Gamma_h)} - (\phi_h - \pi_h\phi_h, U)_{L^2(\Gamma_h)} \\ &\ldots + (\nabla_h\pi_h\phi_h, U_h)_{\Gamma_h,q} - (\nabla_h\pi_h\phi_h, U_h)_{L^2(\Gamma_h)} + (\pi_h\phi_h, U_h)_{\Gamma_h,q} - (\pi_h\phi_h, U_h)_{L^2(\Gamma_h)} \\ &\ldots + (f, \pi_h\phi_h)_{L^2(\Gamma_h)} - (f, \pi_h\phi_h)_{\Gamma_h,q}. \end{aligned}$$

With the third-order quadrature that we are using, we have second-order convergence provided the integrands are sufficiently smooth. For a general integrand $I(x)$, third-order quadrature on the simplex S_k gives

$$\frac{1}{3}\sum_{j=1}^{3} I(e_{j,m})A(\S_k) = \int_{S_k} l_h I(x)$$

where $l_h I(x)$ is the affine function on S_k with the same values at the midpoints of the edges as $I(x)$. This is bounded by

$$\max_{S_k} \Delta I(x) A(S_k).$$

Thus, the integral over all of Γ_h converges with order h^2. In the estimate above, the three integrals we approximate with Gaussian quadrature are

$$(\pi_h \phi_h, f_h)_{L^2(\Gamma_h)},$$

$$(\pi_h \phi_h, U_h)_{L^2(\Gamma_h)},$$

and

$$(\nabla_{\Gamma_h} \pi_h \phi_h, \nabla_{\Gamma_h} U_h)_{L^2(\Gamma_h)},$$

where U_h is smooth on each simplex. The regularity of ϕ_h depends on that of ψ_h. Therefore, we have second order convergence provided that f_h and ψ_h are sufficiently smooth.

Theorem 8 (Error Representation Formula)**.** *The error in the quantity of interest e satisfies*

$$e = \left(f, \phi - \phi_h^l\right)_{L^2(\Gamma)} + \left(\hat{f}\mu_h - f_h, \phi_h\right)_{L^2(\Gamma_h)} + (R(U), \phi_h - \pi_h\phi_h)_{L^2(\Gamma_h)} + \sum_k \int_{\partial S_k} \left(\frac{\partial U}{\partial \nu}, \phi_h - \pi_h \phi_h\right) + e_q.$$

Here, each set S_k is a simplex and the sum is over all simplices and $R(U)$ represents the residual,

$$R(U) = f_h + \Delta_{\Gamma_h} U - U.$$

The Laplace-Beltrami operator here is the weak Laplace-Beltrami operator. From the formula above, we get

Theorem 9 (Convergence)**.** *For any quantity of interest ψ, the error formula converges to* 0 *with order h^2, provided f_h and ψ_h are suitably chosen.*

Proof. The first term converges due to the a priori convergence of the method. Remember that ϕ is actually a finite element solution on a very fine approximation of Γ. We have chosen ψ_h in order to give second-order convergence. The second term converges to zero quadratically if f_h is chosen appropriately. For example, $f_h = f^l$ and $f_h = f^l \mu_h$ are both choices that give second-order convergence. The third and fourth terms represent discretization error and include the term $\phi_h - \pi_h \phi_h$, which converges to 0 with order h^2. The quadrature error also converges to zero for sufficiently precise quadrature rules as shown above. □

5.2 Summary

In this chapter, we have shown how to use the finite element method to solve a Poisson problem on a C^2 surface and where error occurs in the process. We have developed an adjoint-based A Posterori Error Estimate for the error that estimates discretization error, geometric error and quadrature error. We have shown how to approximate the terms in the estimates in order to make the estimates computable. In the next chapter, we look at some examples of the method.

6. IMPLEMENTING THE METHOD AND ESTIMATE

In this chapter, we revisit the method and error estimate from the previous chapter and show how all of the terms can actually be computed. We also consider some practical issues in computing the solution and how they might be addressed.

6.1 Generating Surfaces

The first practical question is how much we know of our surface and how we obtain this information. The three general ways we might know the surface are as a parameterization, as a level set of a function or as a list of measurements for the positions of the nodes. In any case, we need to generate an approximating surface Γ_h. The error estimate we have derived also requires an adjoint solution ϕ computed on the true surface. Since the finite element method we are considering does not allow us to solve on the true surface, we approximate ϕ by using the same finite element method to find Φ, an approximate adjoint solution on a finer triangulation of the surface than Γ_h. This means we need two approximating surfaces for Γ. With these rough and fine approximations of the surface, we can approximate μ_h by using the first fundamental form of the fine surface in place of the first fundamental form of the true surface. The surface measure μ_h will be piecewise constant in this case. We can show that the error introduced in this way is asymptotically finer than the error of the coarsest surface, hence asymptotically does not affect the accuracy of the estimate.

Parametrization

A parameterization is, in many ways, the ideal case for computational purposes. This means that we know essentially everything there is to know about the surface. We can compute the first fundamental form G as described earlier or any other information about the surface that we may need. Generating triangulations for Γ is a matter of generating triangulations for the Euclidean parameter space and using the parameterization to place the nodes in three-dimensional space.

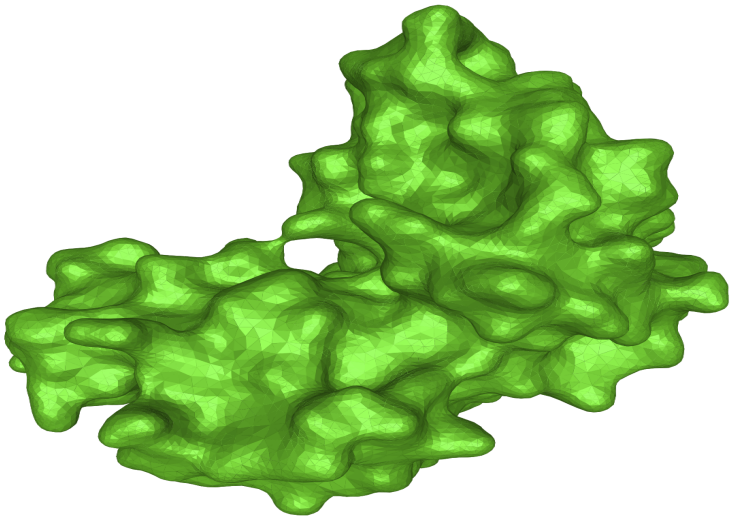

Fig. 6.1: A model of a complex biological molecule.

Unfortunately, this is the least realistic case for actual computations. Most surfaces that might be encountered in real life will be too complicated to parametrize effectively. Consider, for example, the surface shown in Figure 6.1. Most of the time, we will not have a parameterization for such a surface.

Level Set

A surface may be defined as the level set of a function of three variables. This is a less ideal case, since geometric information about the surface such as the first fundamental form is more difficult to compute accurately. Generating a mesh is still quite feasible, though. One way to do it is to start with a mesh of a simpler surface of the same topological type, (sphere, torus, etc...), and then to move the nodes from the reference surface to the desired surface using a diffeomorphism. The mesh can then be refined as desired using the following algorithm.

Algorithm 1: Refining the Mesh for a Level Set

while *Not Satisfied* **do**
  Find the midpoint of each edge in the triangulation;
  Move each midpoint onto the surface using the equation that describes the surface;
  On each simplex, connect the midpoints to form four new simplices;
  Map the rough mesh to the new fine one simplex-by-simplex using affine transformations;
  If the fine mesh is not sufficiently fine, it becomes the rough mesh and we use it to find a new rough mesh;
end

This algorithm globally refines the surface. If we only want local refinement, we can still follow the general strategy of choosing a point on an edge that we want to break up and using the defining function and its gradient to move the point to the surface and make it a node of the new triangulation.

Measurements

The most realistic and also most computationally difficult scenario is one where all we know of the surface is a list of measurements. It is not obvious how to compute geometric information about the surface in this case or how to refine the mesh if desired. Furthermore, there is no way to determine mathematically whether we have enough measurements to adequately describe the surface. As a one-dimensional example, suppose we are studying a curve and that we have measured the points in the figure on the left in Figure 6.2. We will compute the same solution whether the true curve is the one pictured in the middle or the one on the right.

Obviously, we need to know that we have enough measurements before we try to compute anything.

The error estimate we have will not tell us whether we have enough measurements or not. Finally, all measurements are subject to uncertainty, which will introduce another source of error to the computation. We deal with this error in the last chapter.

When what we know about the surface is a list of measurements, we can still generate a mesh of the surface by making the measurement points nodes and connecting them. It may be that we can then get more measurements in order to refine the mesh. We could then do something similar to the previous algorithm by trying to get measurements near the midpoints of the edges in our triangulation. If more measurements are not available, we might use only a subset of our measurements to

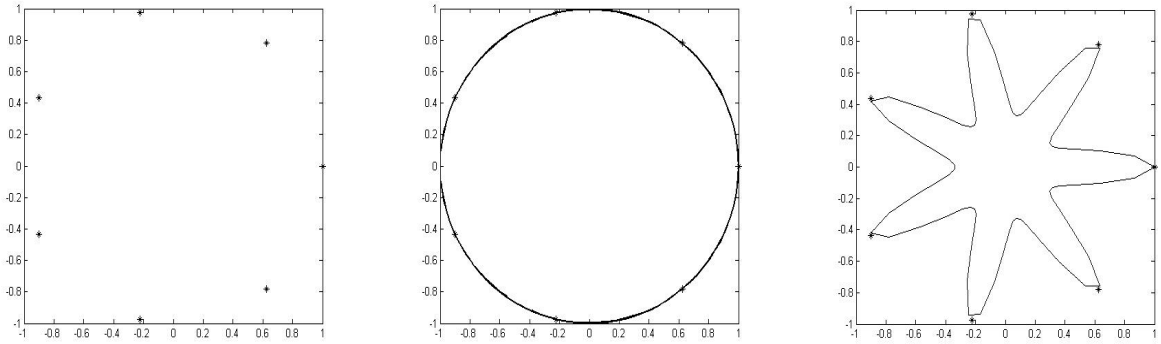

Fig. 6.2: The problem of measurements.

form Γ_h and use all measurements to get the fine mesh. Alternatively, we could attempt to interpolate the the measurements with a parameterization or the level set of a function and then proceed as in the previous sections.

6.2 Forcing Terms

If we have uncertainty and incomplete knowledge about the surface Γ, we will also have uncertainty and incomplete knowledge about the functions $f(x)$ and $\psi(x)$ that are defined on Γ. These functions could be defined in a number of ways including via parameterization, as a property of the surface, in terms of coordinates on three-dimensional space, or as a list of measurements.

Parameterization

Again, a parameterization is an ideal case. If we have an accurate parameterization of the surface, then $f(x)$ and $\psi(x)$ can be defined as functions of the parameters. We can readily measure them at any point. Also just like before, this is the least realistic case. If a parameterization is available, it is most likely subject to some error for a realistic or interesting surface.

Property of the Surface

The forcing terms could be properties of the surface, such as the Gaussian Curvature or the distance on the surface from one point to another. The latter case would apply if, for example, we have a point source located on the surface with heat diffusing along the surface. In this case, we can use Γ_h or the fine approximation to approximate f and ψ. If we need the distance between two points

for example, we can use the algorithms 2 and 3.

Algorithm 2: Calculating the distance to each node from a point

```
At the source node, define dist = 0;
while dist is not defined for at least one node do
    List all nodes where dist is not defined but that are connected to at least one node where
    dist is defined;
    for Each node N_0 on the list do
        For each connected node N where dist is defined, calculate dist(N) + dist(N, N_0);
        Define dist(N_0) as the minimum number calculated above;
    end
end
```

This algorithm approximates the distance of every node in the triangulation from the location of the source. It is a reasonably good approximation for well-behaved surfaces but has the drawback of being expensive to compute. At each quadrature point, we can then approximate the distance with algorithm 3 on the next page.

Algorithm 3: Calculating the distance from each quadrature point to a point

```
H for Each Quadrature Point q. do
    q is on Simplex S with nodes N_1, N_2, N_3;
    for Each Node do
        Calculate dist(N_j) + dist(N_j, q);
        Define dist(q) to be the minimum number calculated above;
    end
end
```

We should perform these computations using the same surface, (either Γ_h or the finer approximation) and then lift the values to the other surface rather than computing them independently. Otherwise we may not get quadratic convergence since if f is a function of (x, y, z) for example,

evaluating f independently on Γ_h and the fine surface would give

$$\begin{aligned}\hat{f}\mu_h - f_h &= f(x+h_1, y+h_2, z+h_3)\mu_h - f(x,y,z) \\ &= f(x,y,z)(\mu_h - 1) + \mathbf{h}\cdot\nabla f\mu_h + O(h^2).\end{aligned}$$

Forcing Term as a Property of the Surrounding Space

The functions f and ψ may be functions of three variables defined on the space in which the surface is located. This might be the case, for example, if a surface is being heated from a source that is not located on the surface or if part of the surface is exposed to different conditions than the rest of the surface – for example if it is partially immersed in a fluid.

This is actually an easier case computationally than the previous one. At each quadrature point, we just need to evaluate the function of three variables. As before, we should do all of the computations on the same surface and lift them to the other surface rather than doing them independently on each surface.

A List of Measurements

It may be that we only know f and ψ from a list of measurements. If this is the case, we need a quadrature rule that uses the measurements we have. If we have the measurements at the nodes, for example, then we can use the values at the nodes for quadrature rather than the values at the midpoints. If we are able to make more measurements of f and ψ, we can use other quadrature methods as well.

6.3 Solutions

Once we have measurements of the surface and forcing terms, we need to evaluate the solutions. We consider the steps here.

Choosing f_h and ψ_h.

We need to choose f_h and ϕ_h to solve the finite element problems. For f_h, two possible choices are

$$f_h = f^l$$

and

$$f_h = f^l\mu_h$$

In both cases, we use the fine approximation to stand in for the true surface Γ. Both of these choices give second-order convergence. The second choice eliminates a term in the error estimate. This may not be desirable, since we are approximating f^l and μ_h. There is therefore error in both of these factors and thus the kind of error that the corresponding term in the estimate is supposed to measure. Choosing f^l will not eliminate this term and will give a better idea of how much error is actually being made. Which choice is the best depends on the problem and the purpose for which it is being solved. This list of choices is not meant to be exhaustive. It is important, however, that the choice of f_h converge to f quadratically. The choice of ψ_h has similar issues, but may be easier to compute since it is defined by what is desired from the computation rather than from external data.

Computing the Adjoint Solutions

As previously stated, we use finite element methods to compute both adjoint solutions, which means that they themselves are approximations, which we now denote Φ and Φ_h. We compute Φ with a CG1 method on the fine approximation of the surface. It converges with order h^2 to ϕ. The first adjoint solution Φ_h is defined and computed on Γ_h. We need it to not lie in the same space of piecewise affine functions as U, otherwise the discretization terms in the error estimate will go to zero by Galerkin Orthogonality. We also need it to be smooth on each of the simplexes for our quadrature. We therefore use a CG2 method to solve for Φ_h. We place piecewise quadratic basis functions at all of the nodes in Γ_h and all of the midpoints. In the Euclidean case, a CG2 solution has a higher convergence rate provided the right hand side has enough regularity. This is not the case, however, for a non-Euclidean problem. Since Γ_h still only converges to Γ at a rate of h^2, the convergence remains quadratic. We use the CG2 method to ensure that ϕ_h is in a different space of functions than U. We will want ϕ_h to be different than its piecewise linear interoplant later.

Assembling the Matrix-Vector Problems

We adapt code written by Donald Estep and Michael Holst. When we compute a solution, we need to compute the matrix and the vector. We compute the entries simplex by simplex. On each simplex, there are six combinations of two basis functions for CG1 and and fifteen combinations for CG2. We use Gaussian Quadrature to compute each integral of the form

$$\int_{S_k} \nabla_h v_j \cdot \nabla_h v_k + v_j v_k d\sigma_h,$$

as well as each integral of the form

$$\int_{S_k} f\phi_j d\sigma_h.$$

For the dual problems, we use a seven-point Gaussian Quadrature rule. For the forward problem U, we use three-point quadrature as described above. The three-point quadrature rule is the least fine rule that gives second-order convergence. The seven-point rule for the dual problems ensures that all integrals are computed with a high degree of accuracy so that we only need to estimate the quadrature error for the forward solution. We get the contribution for each entry from each simplex in this manner and add it to the corresponding entry of the matrix and vector. Remember, we form three matrices for three problems this way:

- The forward problem on the rough mesh, using CG1.
- The adjoint problem on the rough mesh, using CG2.
- The adjoint problem on the fine mesh, using CG1.

6.4 Terms in the Error Estimate

Our general plan for computing each integral will be the following:

Algorithm 4: General Plan for Computing a Term in the Error Estimate

for *Each Simplex in Rough Mesh* **do**
  Get the values of f_h, U_h, ϕ_h at all vertices;
  Use interpolation and Gaussian Quadrature to evaluate the integral;
  The integral is stored in an array with one entry for each simplex;
end

Geometric Error and Forcing Terms

The terms in the error estimate corresponding to geometric error are

$$e_g = \left(f, \phi - \phi_h^l\right)_{L^2(\Gamma)} + \left(\hat{f}\mu_h - f_h, \phi_h\right)_{L^2(\Gamma_h)}.$$

Remember that we now have

$$e_g \approx \left(f, \Phi - \Phi_h^l\right)_{\Gamma_f} + \left(\hat{f}\mu_h - f_h, \Phi_h\right)_{\Gamma_h}.$$

The first term in the estimate,

$$\int_{\Gamma_h} f^l \left(\Phi^l - \Phi_h\right) \mu_h d\sigma_h,$$

is computed as follows:

Algorithm 5: Computing the First Error Term.

Computation actually performed on the fine mesh.;
for *Each Simplex* **do**
  Use the list of which new simplices come from which old ones to get the appropriate values of Φ_h at the nodes;
  μ_h does not need to be used since we are computing on the fine mesh.;
  Use interpolation and Gaussian quadrature to evaluate the integral.;
  Produces two arrays. One has an entry for each simplex on the fine mesh. The other has an entry for each simplex on the coarse mesh.;
end

If we do not use $f_h = f^l\mu_h$, we also need to compute this term:

$$\int_{\Gamma_h} \Phi_h \left(f^l\mu_h - f_h\right) d\sigma_h.$$

Algorithm 6: Calculating the Second Error Term

for *Each Simplex in Rough Mesh* **do**
  If we are using $f_h = f^l$, just need this at each Quadrature Point along with ϕ_h and μ_h;
  Divide each simplex into fourths since μ_h is a piecewise constant.;
  Use Gaussian Quadrature again to evaluate each integral as before, keeping the integral calculated for each simplex separate. ;
end

One feature of the error estimate we have given is that the geometric error estimate can be computed without computing the solution U. We can use this fact to efficiently refine the triangulation until geometric error is satisfactory. The following algorithm can be used:

Algorithm 7: Obtaining a Mesh With Acceptable Geometric Error

while *Not Satisfied* **do**
  For the current triangulation, get ψ, ψ_h, f and f_h;
  Solve both adjoint problems to get Φ and Φ_h;
  Compute $e_g \approx \left(f, \Phi - \Phi_h^l\right)_{L^2(\Gamma)} + \left(\hat{f}\mu_h - f_h, \Phi_h\right)_{L^2(\Gamma_h)}$;
  Refine mesh if desired;
  Continue until e_g is satisfactory;
end

Discretization Error

The term in the estimate corresponding to the discretization error is approximated by

$$e_d \approx \int_{\Gamma_h} f_h \left(\Phi_h - \pi_h \Phi_h\right) d\sigma_h - \int_{\Gamma_h} \nabla_h U_h \cdot \nabla_h \left(\Phi_h - \pi_h \Phi_h\right) d\sigma_h - \ldots$$
$$\int_{\Gamma_h} U_h \left(\Phi_h - \pi_h \Phi_h\right) d\sigma_h$$

To calculate this, we again go simplex by simplex.

Algorithm 8: Calculating the Discretization Error Term

for *Each Simplex in Rough Mesh* **do**
  Get the values of f_h, U_h, Φ_h at all vertices;
  Use interpolation and Gaussian Quadrature to evaluate integral;
  The integral is stored in an array with one entry for each simplex;
end

Estimating Qudarature Error

The final error terms are

$$e_q \approx \left(\nabla_h \pi_h \Phi_h, U_h\right)_{L^2(\Gamma_h)} - \left(\nabla_h \pi_h \Phi_h, U_h\right)_{h,q} + \ldots$$
$$+ \left(\pi_h \Phi_h, U_h\right)_{L^2(\Gamma_h)} - \left(\pi_h \Phi_h, U_h\right)_{h,q} + \left(f, \pi_h \Phi_h\right)_{L^2(\Gamma_h)} - \left(f, \pi_h \Phi_h\right)_{h,q}$$

We again proceed simplex by simplex. We estimate the true integrals using a seven-point Gaussian Quadrature method. This allows us to accurately estimate the error without computing the true integral. The estimate using this approach converges to zero with order h^2.

Putting Everything Together and Convergence

After these computations, we now have four arrays with an entry for each simplex. When they are added and summed, we get the error estimate for the quantity of interest. Without summing, we

have the contribution of each simplex to the error in the quantity of interest. We have an idea of how much of each type of error we get from each simplex and we may use this for adaptivity to refine the mesh if needed or use higher-order quadrature wherever it is needed. Our estimate of the quantity of interest still converges to the true quantity of interest.

Theorem 10. *The computed quantity of interest*

$$(\psi_h, U)_{L^2(\Gamma_h)}$$

converges to the true value

$$(\psi, u)_{L^2(\Gamma)}$$

with order h^2.

Proof.

$$(\psi, u)_{L^2(\Gamma)} - (\psi_h, U)_{L^2(\Gamma_h)} = (\psi, u)_{L^2(\Gamma)} - (f, \Phi)_{L^2(\Gamma)} + (f, \Phi)_{L^2(\Gamma)} - (\psi_h, U)_{L^2(\Gamma_h)}$$

The first difference converges with order h^2 due to the convergence of the method. The second term converges because of the following. □

Theorem 11. *The estimate using* Φ *and* Φ_h *converges to zero with order* h^2.

Proof. The estimate is given by

$$\begin{aligned} e \approx & \left(f, \Phi - \Phi_h^l\right)_{L^2(\Gamma)} + \left(\hat{f}\mu_h - f_h, \Phi_h\right)_{L^2(\Gamma_h)} \cdots \\ & + (f_h, \Phi_h - \pi_h\Phi_h)_{L^2(\Gamma_h)} - a_{\Gamma_h}\left((\Phi_h - \pi_h\Phi_h), U\right) - (\Phi_h - \pi_h\Phi_h, U)_{L^2(\Gamma_h)} \\ & \ldots + a_{\Gamma_{h,q}}(\pi_h\Phi_h, U_h) - a_{\Gamma_h}(\pi_h\Phi_h, U_h) + (\pi_h\Phi_h, U_h)_{h,q} - (\pi_h\phi_h, U_h)_{L^2(\Gamma_h)} \\ & \ldots + (f, \pi_h\Phi_h)_{L^2(\Gamma_h)} - (f, \pi_h\Phi_h)_{h,q}. \end{aligned}$$

The first term converges by the convergence of the method. If ϕ is the true solution, then

$$\left(f, \Phi - \Phi_h^l\right)_{L^2(\Gamma)} = (f, \Phi - \phi)_{L^2(\Gamma)} + \left(\phi - \Phi_h^l\right)_{L^2(\Gamma)} \leq C \left\|f\right\|_{L^2(\Gamma)} h^2.$$

As stated before, it is important to choose f_h so that

$$f_h^\mu - f_h$$

converges to zero. Using $f_h = f^l$ or $f_h = f^l \mu_h$ will accomplish this. We can use integration by parts on the terms involving $\Phi_h - \pi_h \Phi_h$ to get these terms equal to

$$(R(U), \Phi_h - \pi_h \Phi_h)_{L^2(\Gamma_h)} + \sum_k \int_{\partial S_k} \frac{\partial U}{\partial \nu} \cdot (\phi_h - \pi_h \phi_h) .$$

This converges to zero quadratically since $\Phi_h - \pi_h \Phi_h$ converges to zero quadratically. The quadrature error estimates also converge to zero. Remember that we are using higher-order quadrature to estimate the true integrals in this estimate. For each term, let Q be the true value of the integral, Q_h be the high-order estimate and Q_l be the lower estimate. We have

$$|Q_h - Q_l| \leq |Q_h - Q| + |Q_l - Q| \leq Ch^2$$

The last step follows as before, provided f is sufficiently smooth by the quadrature that we are using.

□

6.5 Adaptivity

As previously stated, evaluating the error estimate gives us four arrays, each having one entry for each simplex. If we sum all of them, we get our error estimate. We can also see how much of each type of error comes from each simplex. The error that we estimate is the error in the computation of the quantity of interest rather than the total error in the entire solution. One useful feature of this estimate is that the contribution is zero from any region of the surface where either f or ψ are zero is zero. If a region is not relevant to the right hand side of the equation or to the quantity of interest, then the geometric error from approximating that portion of the surface does not contribute to error in the quantity of interest and it is not important to approximate precisely there. This helps us refine with maximum efficiency. We can see where we need to refine to improve error in the quantity of interest and what regions do not contribute to the quantity of interest. This will also be a sharper estimate than one achieved by approximating the norm of the Laplace-Beltrami operator of the surface as in [11]. This is because such a bound needs to be true for all possible quantities of interest while our estimate is only valid for one quantity. If a small number of quantities of interest are needed, then this adjoint-based estimate is preferable to other estimates that do not use the adjoint.

7. NUMERICAL EXAMPLES

In this chapter, we look at some interesting examples of the finite element method and the numerical error. We look at the convergence of the method in general and then look at a case where quadrature error dominates, a case where geometric error dominates and a case where discretization error dominates.

7.1 Overview

For our first examples, we use the domain pictured in Figure 7.1.

The domain is derived from a torus parametrized by

$$(X) = \begin{pmatrix} 2\cos\theta + \cos\theta\cos\phi \\ 2\sin\theta + \sin\theta\cos\phi \\ \sin\phi \end{pmatrix}.$$

We move the torus upward by adding 1.1 to the z-coordinate of every point on the mesh. We then add $\cos(2\theta)$ to the z-coordinate of every point in order to obtain the warped shape.

We assume that the surface is partially immersed in a liquid solution. The temperature of the liquid is higher than that of the surrounding air. To find the equilibrium temperature distribution on the surface, we solve the problem

$$-\Delta_\Gamma u + u = f$$

where

$$f(x, y, z) = \chi_{z\leq 0}$$

is equal to 1 when $z \leq 0$ and 0 otherwise. It is defined in the space in which Γ is located. Our quantity of interest is the difference between the average temperature below the surface of the liquid and the average temperature above the surface.

$$\psi(x, y, z) = \frac{1}{A_1}\chi_{z\leq 0} - \frac{1}{A_2}\chi_{z>0},$$

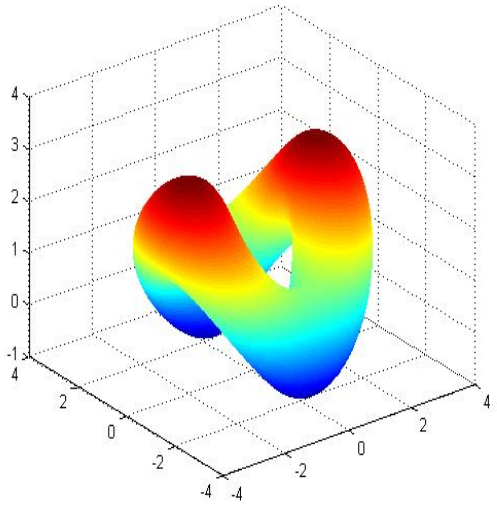

Fig. 7.1: Warped torus domain.

where A_1 is the submerged area and A_2 is the area above the surface.

We solve this problem and, using the algorithms of the previous chapter, estimate the error for a variety of mesh sizes with the following results.

h	Geometric Error, First Term	Geometric Error, Second Term	Total Geometric Error
2.9180	1.38×10^{-2}	.1452	.1590
1.6955	-2×10^{-3}	4.71×10^{-2}	4.5×10^{-2}
0.9146	1.3×10^{-3}	1.02×10^{-2}	1.15×10^{-2}
0.4976	2.0×10^{-4}	2.2×10^{-3}	2.4×10^{-3}
0.2612	1.0×10^{-4}	5×10^{-4}	6×10^{-4}
h	Discretization Error	Quadrature Error	Total Error Error
2.9180	.1298	-7.4×10^{-3}	.2814
1.6955	6.04×10^{-2}	2.63×10^{-2}	.1318
0.9146	2.31×10^{-2}	-8.2×10^{-3}	2.63×10^{-2}
0.4976	7.8×10^{-3}	2×10^{-4}	1.04×10^{-2}
0.2612	2.4×10^{-3}	-1.4×10^{-3}	1.5×10^{-3}

We can see from the results that all forms of error except quadrature error seem to be converging to zero. We will look at the problem with quadrature error more closely in the next section. It should also be noted that all of the error estimates are signed. When they are added, errors partially cancel. In the following figures, we show plots of the solutions for each mesh.

In the Figure 7.3, we show the log-log plots for all of the error terms against h. We can see that

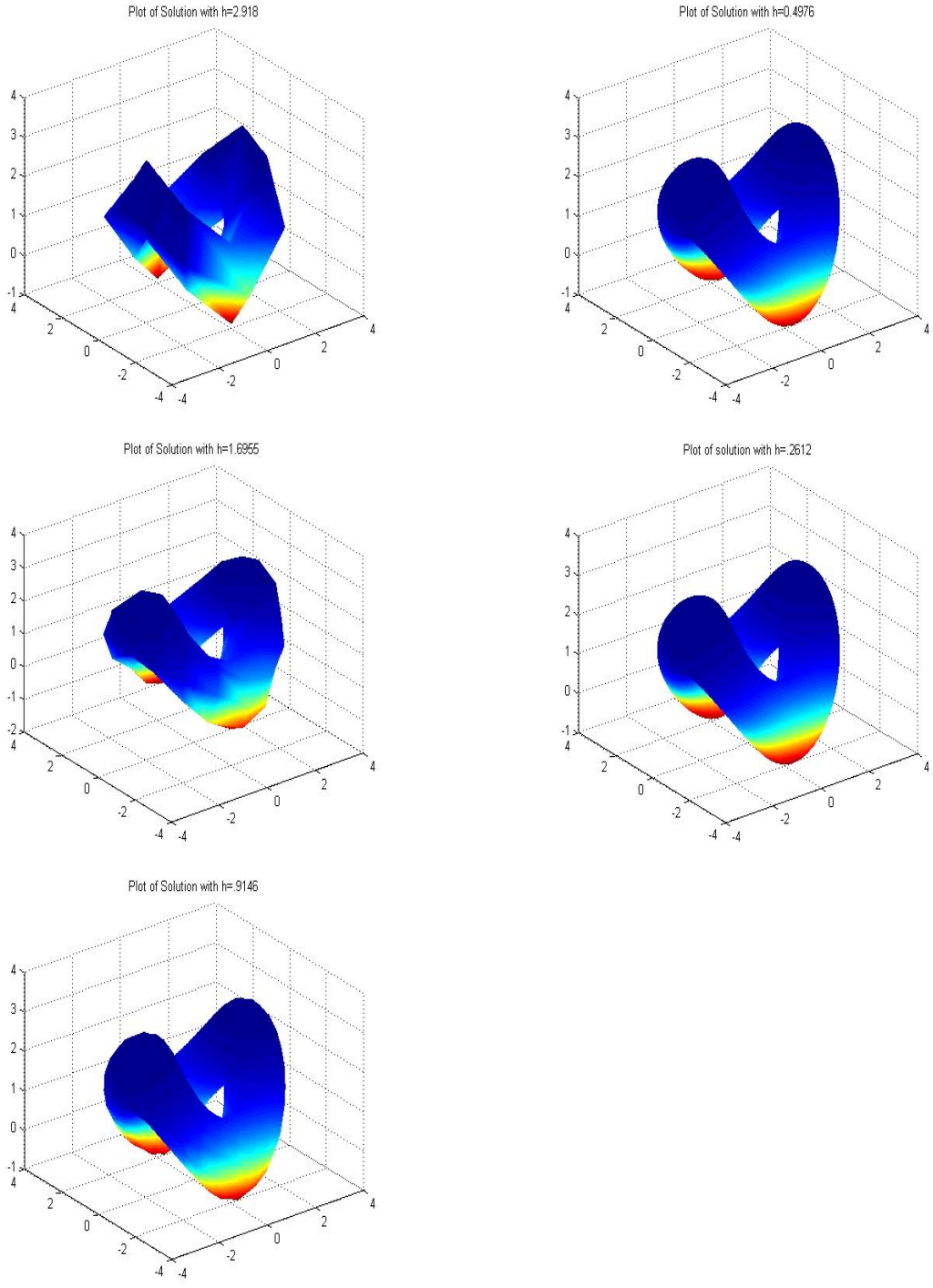

Fig. 7.2: Plots of the Forward Solutions for $h = 2.918, 1.6955, .9146, .4976$ and $.2612$.

they are converging at an approximately quadratic rate except for the quadrature error. This will be explored further in the following section.

7.2 An Example with Signficant Quadrature Error

In the previous example, we had quadratic convergence in every error term except for the quadratic error. This is due to the discontinuous nature of the forcing functions f and ψ. In order to see this, we look at a graph of the quadrature error by simplex in Figure 7.4 for the case of $h = .2612$.

All of the quadrature error is located at the surface of the liquid at $z = 0$. At this resolution, we solve the problem on the same mesh using four-point Gaussian Quadrature rather than three-point Gaussian Quadrature. This reduces the quadrature error to -2.4847×10^{-4}.

In general, quadrature error is large relative to the other error terms when the forcing terms have high variation or discontinuity. When this is the case, using more precise quadrature is an efficient way to improve the accuracy of the result.

7.3 An Example with No Discretization Error

Returning to the warped torus domain, we now consider the same forcing function $f(x)$ but use $\psi(x) = 1$. This means the quantity of interest is the integral of U over the domain:

$$q = \int_\Gamma u d\sigma.$$

When we use this quantity of interest, we have no discretization error. This occurs because the solution for the problem

$$-\Delta_\Gamma u + u = 1$$

is $u = 1$ for any surface. The solution is in the space of trial functions for our finite element method. In particular, since $\phi_h = 1$ is the adjoint solution, $\phi_h = \pi_h \phi_h$. The error formula from the previous two chapters reduces to

$$e = \left(\phi_h, \hat{f}\mu_h - f_h\right)_{L^2(\Gamma_h)} = \int_{\Gamma_h} \hat{f}\mu_h - f_h d\sigma_h.$$

On the mesh with $h = .2612$, the estimate for the geometric error is 8.7×10^{-3}.

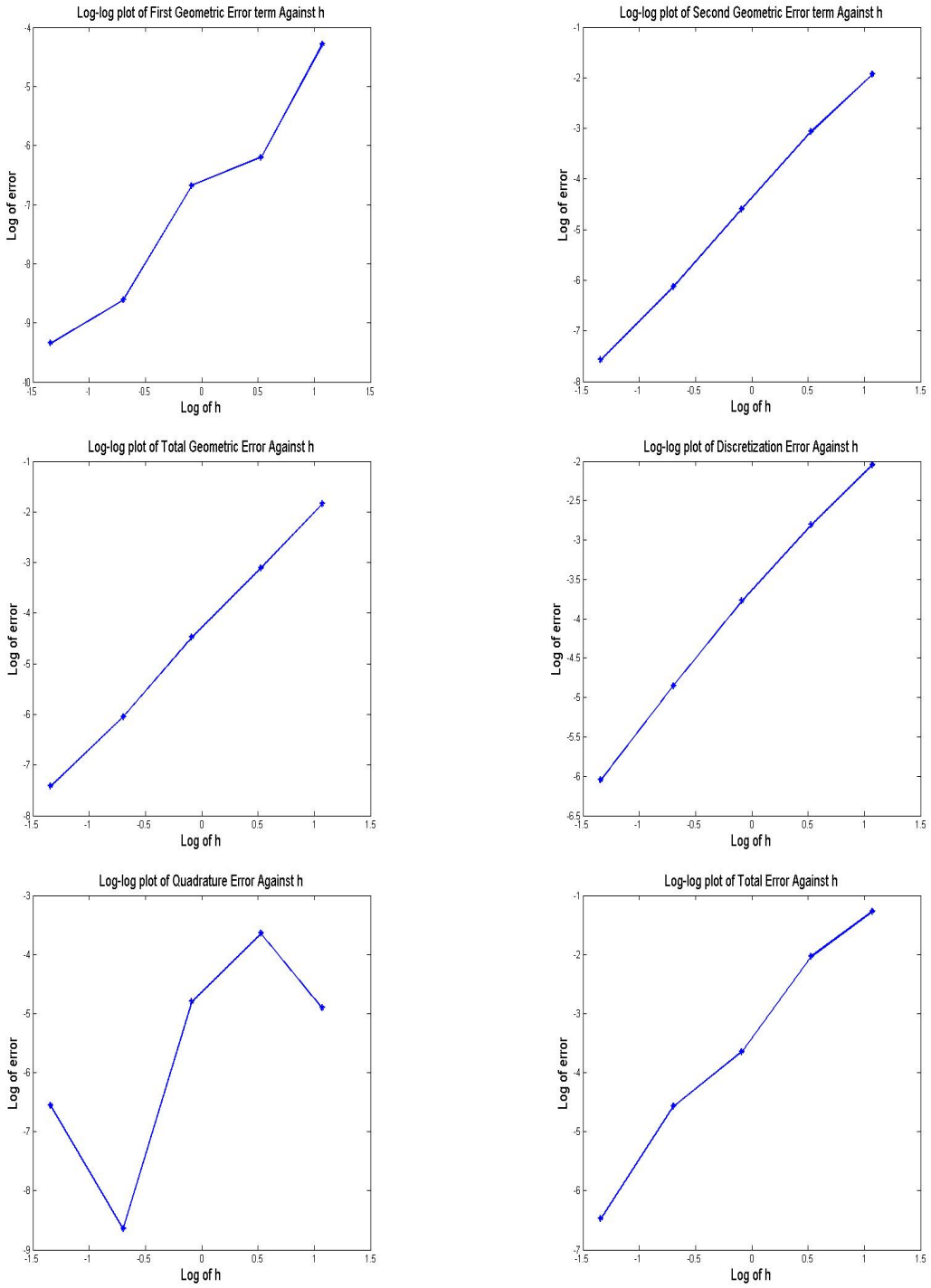

Fig. 7.3: Log-log plots of the error against h for $h = 2.918, 1.6955, .9146, .4976$ and $.2612$. We can see quadratic convergence for all terms except quadrature error.

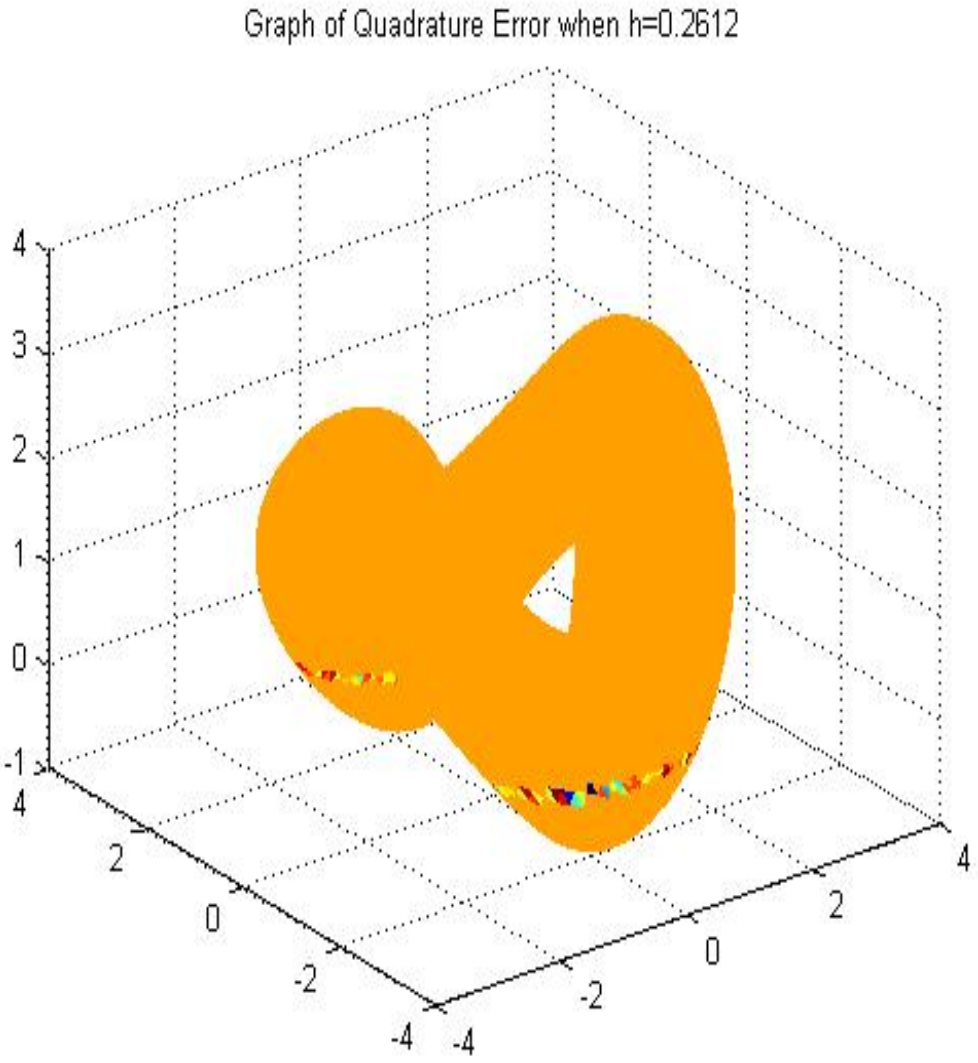

Fig. 7.4: Graph of Quadrature Error by simplex for $h = .2612$.

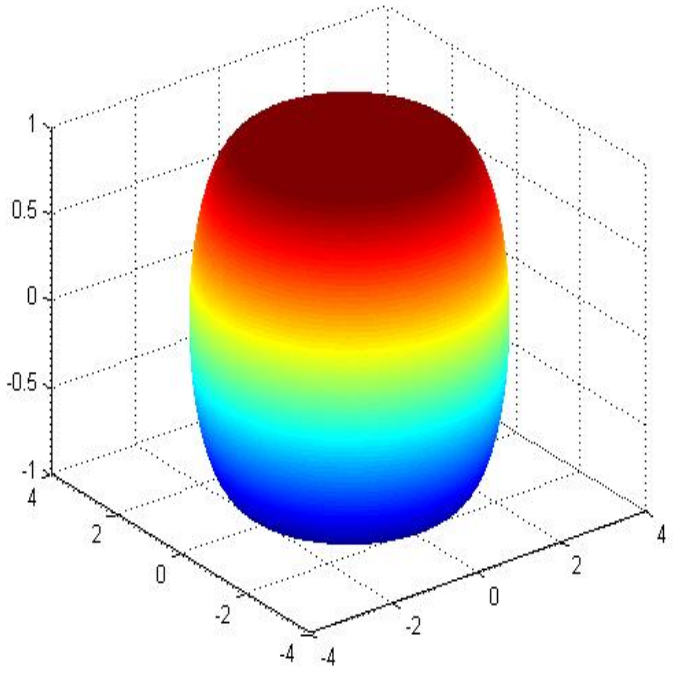

Fig. 7.5: Surface with flat regions.

7.4 An Example with Low Geometric Error and Higher Discretization Error

In this example, we solve the modified steady-state problem on the surface in Figure 7.5.

This surface is obtained from a torus by covering the top and bottom with a flat circle, eliminating the "hole." We use

$$f = \frac{1}{1+r^2}$$

and

$$\psi = 100 * e^{-100(x^2+y^2)}.$$

With this quantity of interest, we are focusing on the region of the surface that is flat and largely ignoring the rounded region of the surface that is less well approximated. When we solve the equation for a variety of mesh sizes, we get the following results.

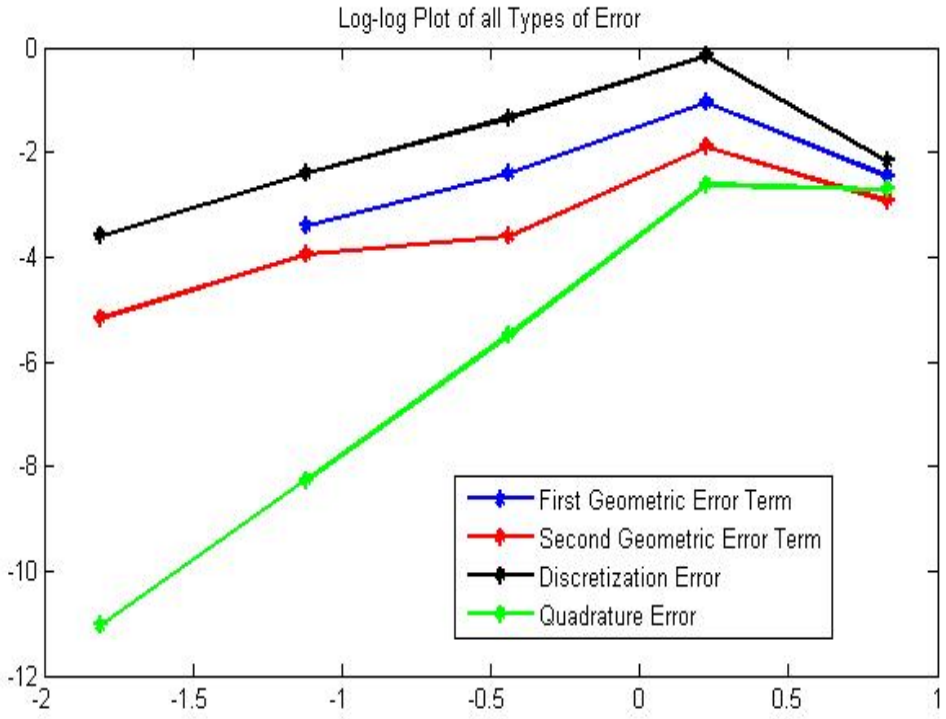

Fig. 7.6: Log-log plot for all types of error.

h	Geometric Error, First Term	Geometric Error, Second Term	Total Geometric Error
2.2961	-8.6×10^{-2}	5.44×10^{-2}	-3.17×10^{-2}
1.2502	$-.3525$	0.1526	$-.1999$
0.6460	-9.08×10^{-2}	2.75×10^{-2}	-6.3×10^{-2}
0.3259	-3.34×10^{-2}	1.95×10^{-2}	-1.39×10^{-2}
h	Discretization Error	Quadrature Error	Total Error
2.2961	0.1155	6.76×10^{-2}	0.1513
1.2502	0.8605	7.4×10^{-2}	0.7346
0.6460	.2652	4.2×10^{-3}	0.2062
0.3259	9.11×10^{-2}	3×10^{-4}	7.74×10^{-2}

The mesh with $h = 2.2961$ is very crude and yields an artificially low error estimate. After that, however, we see convergence of all forms of error and that discretization error is the largest form of error. In Figure 7.6, we show the log-log plot of all types of error against h.

On the same surface, if we use

$$\psi = x^2 + y^2$$

we get a very different result. This quantity of interest places much more emphasis on the sides of

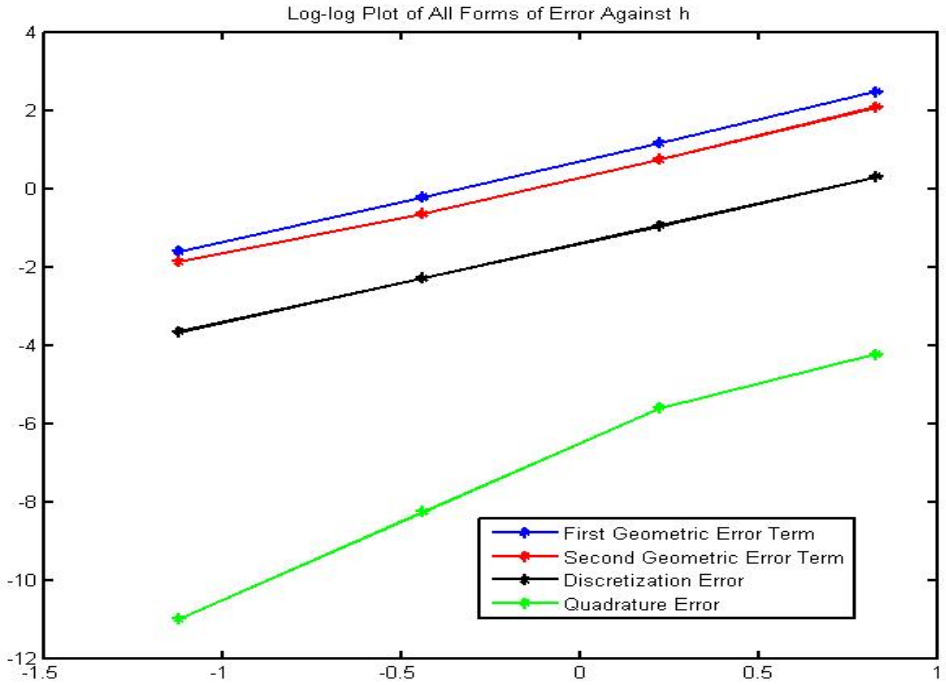

Fig. 7.7: Log-log plot for all forms of error.

the surface where there is geometric error. Here are the results with this quantity of interest.

h	Geometric Error, First Term	Geometric Error, Second Term	Total Geometric Error
2.2961	-11.9832	8.0943	-3.9339
1.2502	-3.1952	2.1088	-1.0864
0.6460	-0.8065	0.5290	$-.2775$
0.3259	-0.2014	0.1545	-4.69×10^{-2}
h	Discretization Error	Quadrature Error	Total Error Error
2.2961	-1.3573	1.46×10^{-2}	-5.2766
1.2502	-0.3869	3.7×10^{-3}	-1.4696
0.6460	-0.1019	3×10^{-4}	-0.3791
0.3259	-2.58×10^{-2}	1.67×10^{-5}	-7.27×10^{-2}

In Figure 7.7, we show the log-log plot of all forms of error against h.

This again highlights the importance of the quantity of interest in determining how much error is made and how much error comes from what source.

7.5 Summary

We have looked at different situations where different aspects of the error are important. In general, geometric error is low when the surface is close to being flat. This is because we are approximating the surface with a piecewise flat surface. Since geometric error arises from the distortion of angles, distances and areas, a surface that is almost flat has low geometric error.

Discretization error is low when the forward solution or adjoint solution is close to being in the space of test functions. We have seen an example where the adjoint solution for our quantity of interest is in the space of test functions. This eliminates discretization error. Conversely, a solution that is highly non-affine would lead to high discretization error.

Quadrature error is high when f or ψ has large derivatives or is discontinuous. This is because the function in these cases varies a lot over a small region, making quadrature less accurate. When quadrature error is high, the accuracy of the solution of the estimate on the quantity of interest can be improved cheaply by using higher-order quadrature.

All forms of error are low in regions of the domain that do not contribute to the quantity of interest.

Error is a result of all of these factors, and understanding how much error comes from what source plays an important role in solving differential equations.

8. ESTIMATING MEASUREMENT ERROR

In most realistic situations, our knowledge of the surface on which we are doing computations come from measurements. Any time we are working with measurements, measurement error plays a role in the computations. Measurement error arises due to errors when actually measuring the surface. This is different from geometric error. Geometric error arises when we use a simpler surface to approximate a known surface. Measurement error is error due to incomplete knowledge of the surface to be approximated. Measurement error is a more stochastic quantity than the other sources of error we have considered since it inherently deals with incomplete knowledge of the surface where the computation is being made. Much work needs to be done on dealing with measurement error. Here, we examine measurement error with Monte Carlo sampling by repeatedly solving the problem with perturbed nodes.

So far, our error estimate has been derived from the following definition of error.

$$e = (u, \psi)_{L^2(\Gamma)} - (U, \psi_h)_{L^2(\Gamma_h)} .$$

This definition assumes that our formulation of the surface Γ is fully accurate and not subject to measurement error. In order to include measurement error, we expand our definition of error to

$$e = \left(\tilde{u}, \tilde{\psi}\right)_{L^2(\tilde{\Gamma})} - (U, \psi_h)_{L^2(\Gamma_h)} = \left(\tilde{u}, \tilde{\psi}\right)_{L^2(\tilde{\Gamma})} - (u, \psi)_{L^2(\Gamma)} + (u, \psi)_{L^2(\Gamma)} - (U, \psi_h)_{L^2(\Gamma_h)} .$$

Here, we have introduced the term $\left(\tilde{u}, \tilde{\psi}\right)_{L^2(\tilde{\Gamma})}$. We denote by $\tilde{\Gamma}$ the true surface. This is distinct from Γ, our best approximation of the surface based on the measurements we have. Similarly, $\tilde{\psi}$ is the Riesz Representor for the true quantity of interest defined on the surface $\tilde{\Gamma}$ and $\tilde{u}$ is the true solution defined on the true surface. The second difference in the above expression,

$$(u, \psi)_{L^2(\Gamma)} - (U, \psi_h)_{L^2(\Gamma_h)} ,$$

corresponds to the error estimate we have already considered in this paper. The first difference,

$$\left(\tilde{u}, \tilde{\psi}\right)_{L^2(\tilde{\Gamma})} - (u, \psi)_{L^2(\Gamma)} ,$$

corresponds to measurement error. To estimate this, we solve the problem

$$-\Delta_{\tilde{\Gamma}} u + u = \tilde{f}$$

on the surface $\tilde{\Gamma}$, where $\tilde{\Gamma}$ for the computation is a randomly generated surface using our best approximation of Γ as the mean. We think of the true surface and the true forcing function $\tilde{f}$ as random to represent our uncertainty and incomplete knowledge about them.

There are a number of problems that might be considered on a randomly perturbed manifold. Here, we proceed by randomly perturbing each of the vertices in our triangulation. This is a reasonable simulation of a situation where our approximate surface is based on measurements that are subject to error. We proceed by Monte Carlo sampling. For each iteration, we draw perturbations for each node from a specified probability distribution. This gives us $\tilde{\Gamma}_n$, a realization of the random surface $\tilde{\Gamma}$. We generate $\tilde{f}_n$ and $\tilde{\psi}_n$ using $\tilde{\Gamma}_n$. We then solve the forward problem and compute the error estimates on $\tilde{\Gamma}_n$ and use this to compute $\tilde{q}_n$.

When $\tilde{q}_n$ is computed, it is still subject to geometric, discretization and quadrature error. We can use the error estimates that we compute to improve the results. Since the error estimate is signed, we can add it back to $\tilde{q}_n$ to obtain a more accurate value for $\tilde{q}_n$. Alternatively, we could reject samples where the absolute value of the estimate is too high. Here, we use the former method.

As we repeat the process many times, we generate many samples of the random variable $\tilde{q}$. Monte Carlo sampling is often concerned with the convergence of the distribution as the number of samples goes to infinity. We are more concerned, however, with the behavior the variance of the empirical distribution when the uncertainty is decreased. We compute the variance of the empirical distribution and use that as a statistic to estimate the measurement error. The method is summarized below.

8.1 Implementation on a Sphere

In this section, we implement the method we have described using a mesh for a sphere. This does not mean that the true surface is a sphere. It means that we are approximating it with a sphere. We use six levels of refinement of the mesh. At each level of refinement, we introduce to each vertex a random perturbation to each coordinate. We draw the random perturbation from a uniform distribution on the interval $(-\max(P), \max(P))$. For each mesh, we use five different values for e.

Algorithm 9: Using Monte Carlo sampling to estimate measurement error

H For a surface, get Γ and Γ_h, the fine and rough meshes corresponding to our best measurements of the surface;
for $n = 1 : N$ **do**
    Generate $\tilde{\Gamma}_n$ and $\tilde{\Gamma}_{h,n}$;
    for *Each node in* Γ **do**
        Draw perturbations in the x, y and z directions from a specified distribution;
        Add perturbations to coordinates of node;
    end
    Get $\tilde{f}_n$, $\tilde{f}_{h,n}$, $\tilde{\psi}_n$ and $\tilde{\psi}_{h,n}$ for $\tilde{\Gamma}_n$ and $\tilde{\Gamma}_{h,n}$;
    Use data to compute $\tilde{q}_n$ and error estimate;
    Update $\tilde{q}_n$ by adding error estimate;
end
Estimate distrbution of $\tilde{q}$ with list $\{\tilde{q}_n\}_{n=1}^N$;
Using estimated distribution, show empirical CDF and calculate variance;
The variance computed is the estimate for measurement error;

We use two different functions for f in each case, $f = z^2$ and $f = \sin(10x)\cos(5y)\sin(10z)$. In every case, we see that the variance of the distribution $\tilde{q}$ goes to zero quadratically as e goes to zero. We also show the cumulative distribution function for each case. On each cumulative distribution function, we also include the minimum and maximum values calculated for q from the case of our largest e for that level of refinement in order to show the convergence of cumulative distribution function more clearly. We also show the log-log plots of the variance against the maximum perturbation. We have one log-log plot for each combination of a mesh size and a forcing term f. Each data point then is the variance when 100 samples are drawn using the same maximum perturbation.

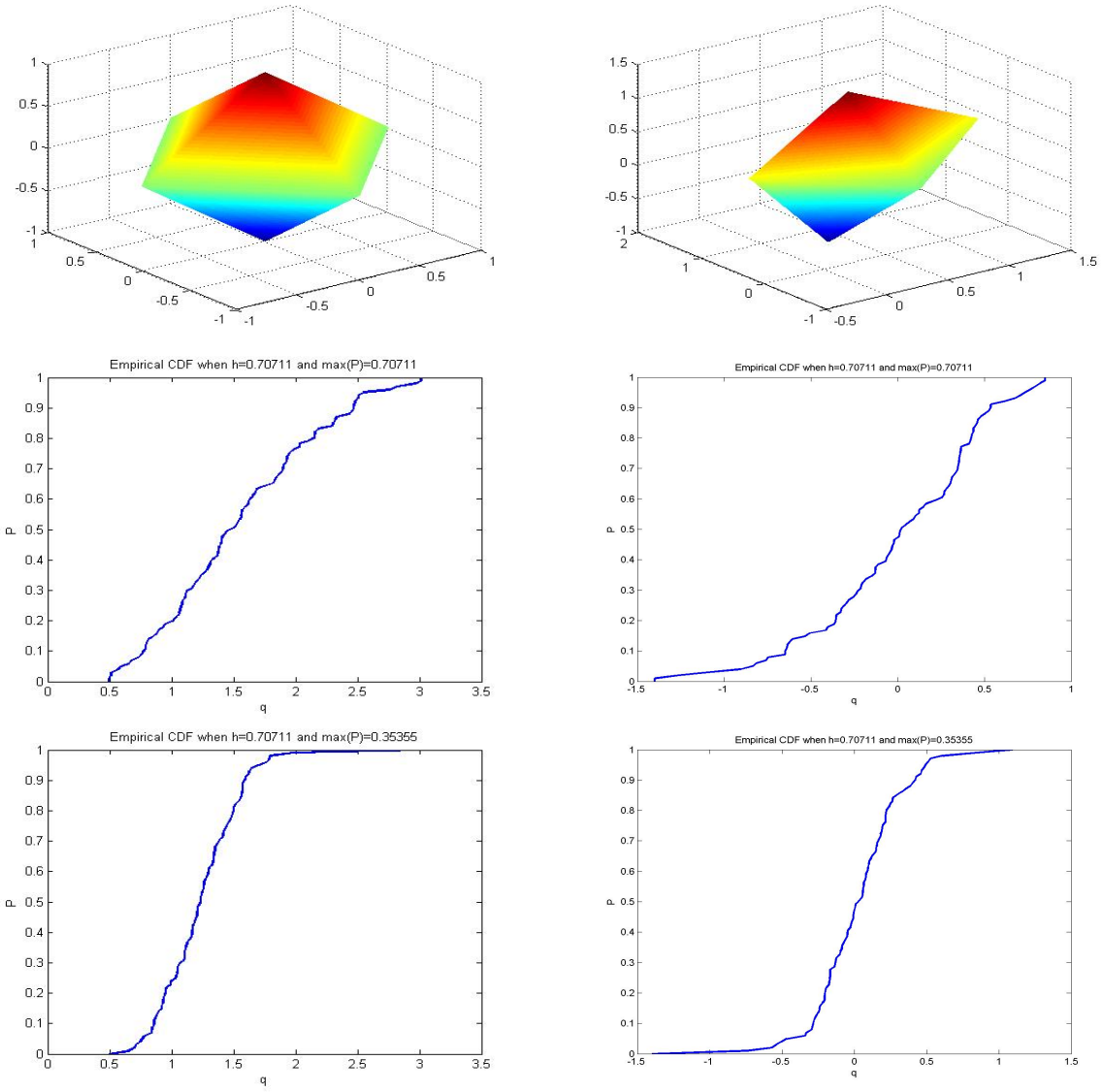

Fig. 8.1: Results on a very rough mesh. In the top row, we have a diagram of the mesh and an example of a perturbed mesh. On the second and third rows, we have the Cumulative Density Function for q with $f = z^2$ on the left and $f = \sin(10x)\cos(5y)\sin(10z)$ on the right.

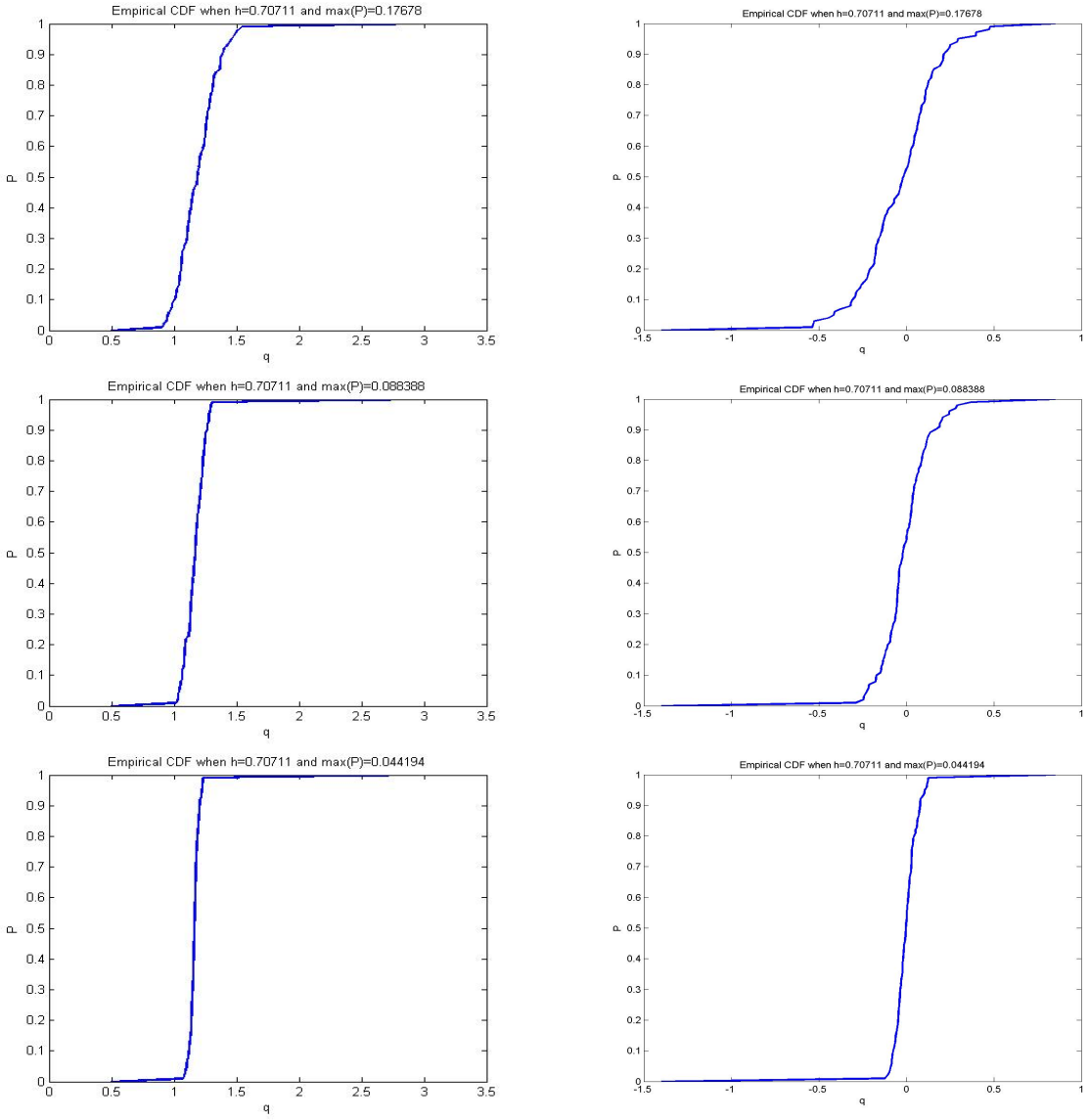

Fig. 8.2: More results for the rough mesh with $f = z^2$ on the left column and $f = \sin(10x)\cos(5y)\sin(10z)$ on the right column. We can see the Cumulative Density Function becoming sharper and sharper as the maximum perturbation decreases.

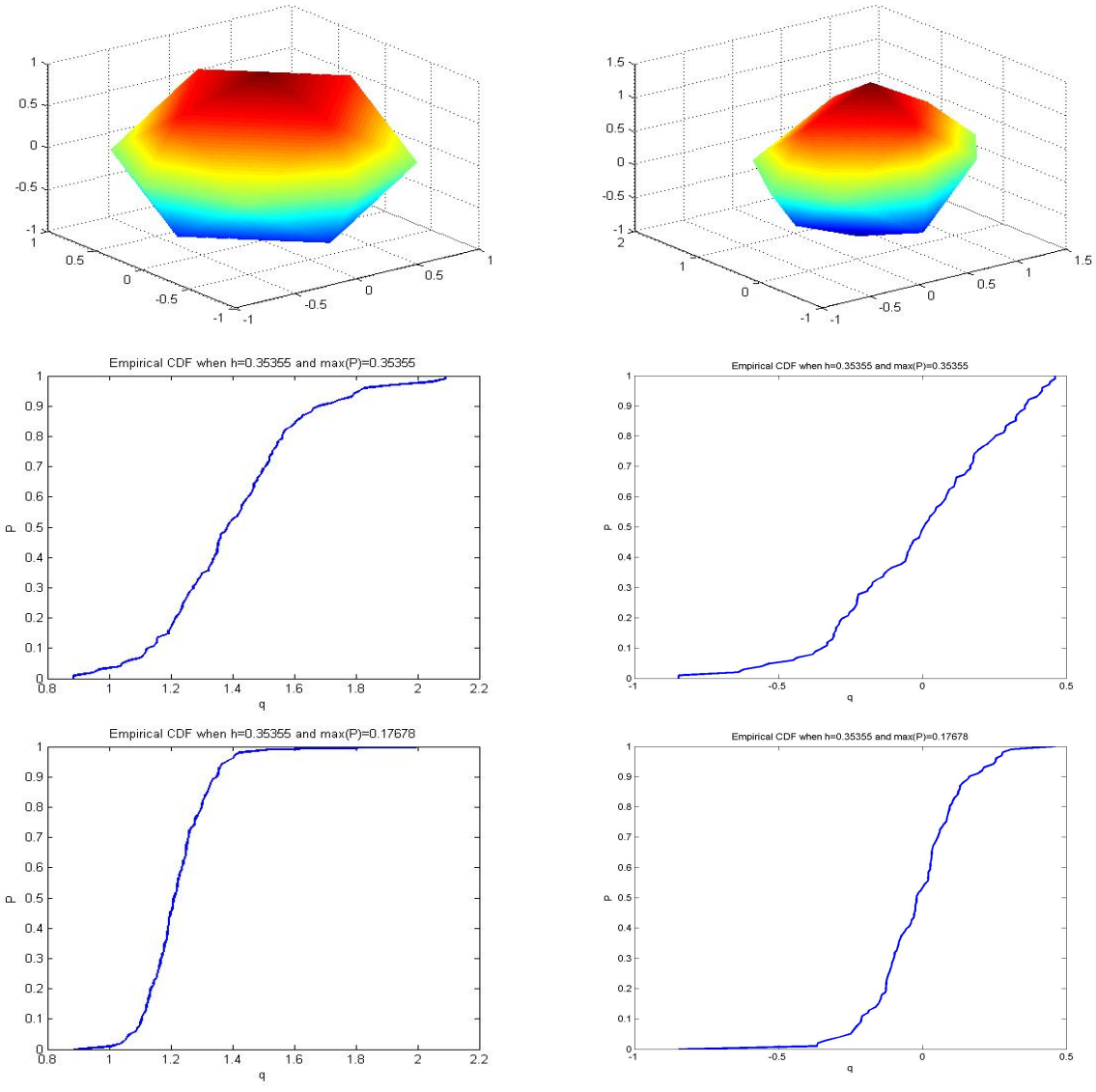

Fig. 8.3: Results after one refinement. In the top row, we have a diagram of the mesh and an example of a perturbed mesh. On the second and third rows, we have the Cumulative Density Function for q with $f = z^2$ on the left and $f = \sin(10x)\cos(5y)\sin(10z)$ on the right.

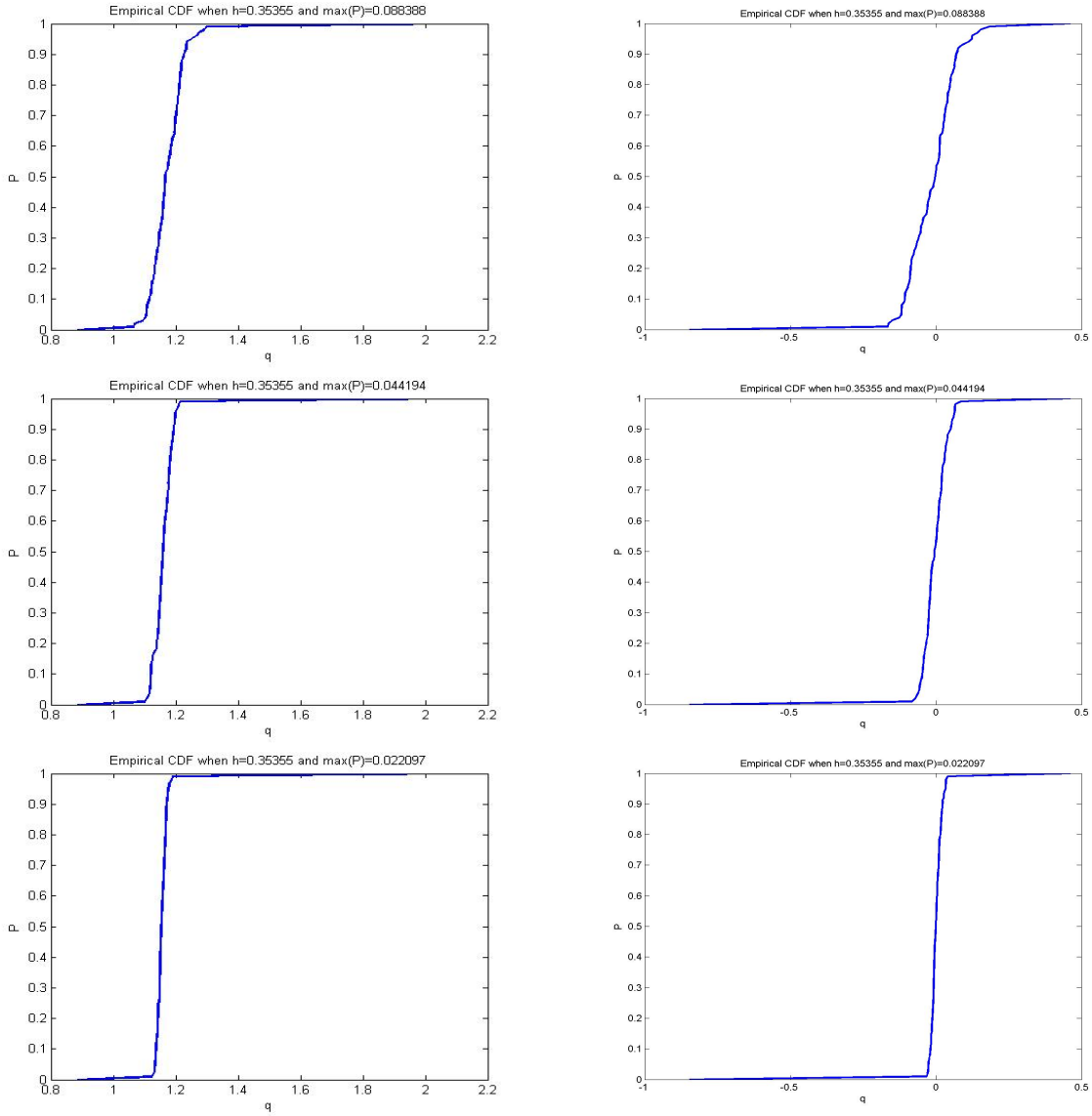

Fig. 8.4: More results for the mesh after one refinement with $f = z^2$ on the left column and $f = \sin(10x)\cos(5y)\sin(10z)$ on the right column. We can see the Cumulative Density Function becoming sharper and sharper as the maximum perturbation decreases.

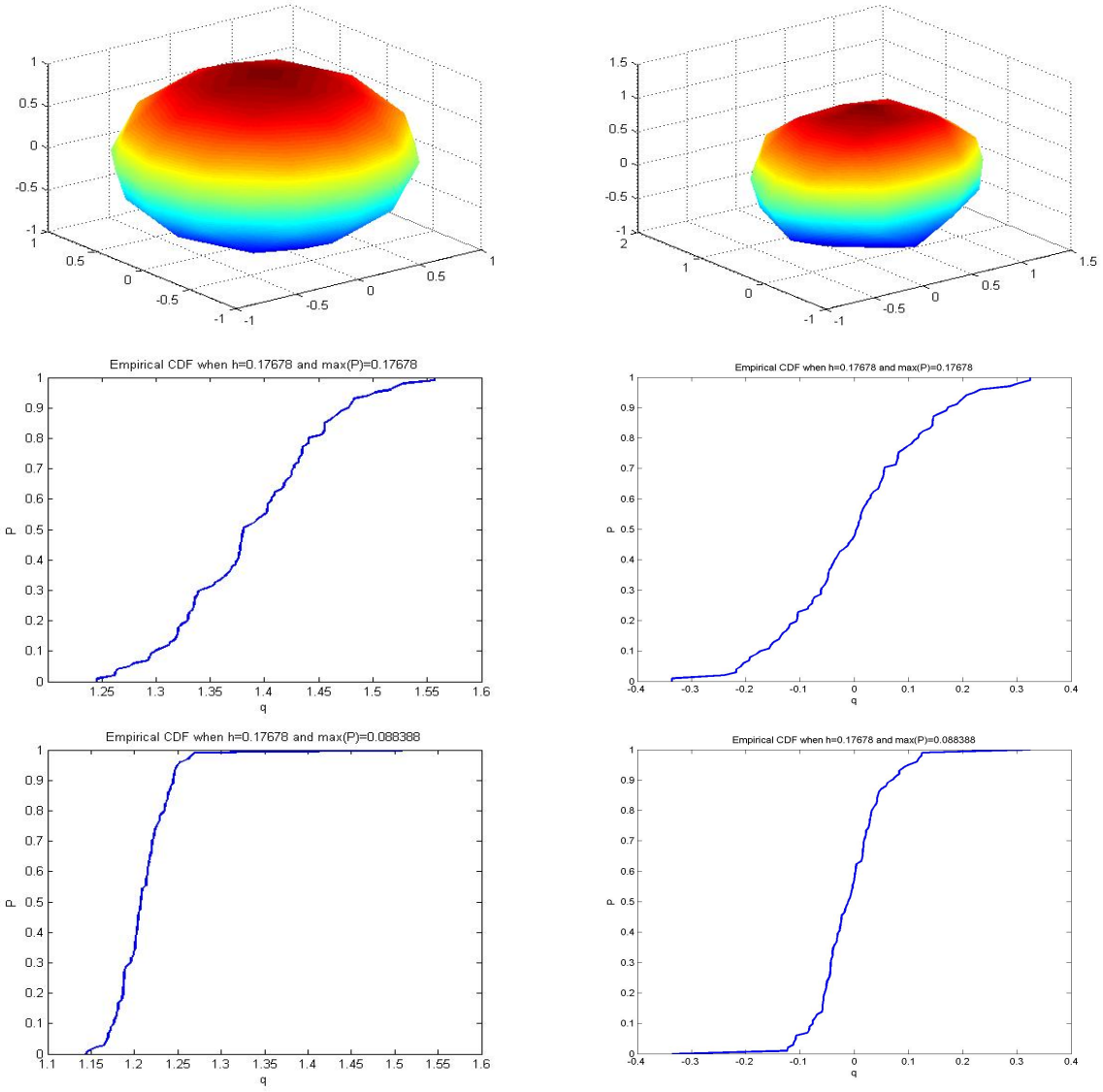

Fig. 8.5: Results on the mesh after two refinements. In the top row, we have a diagram of the mesh and an example of a perturbed mesh. On the second and third rows, we have the Cumulative Density Function for q with $f = z^2$ on the left and $f = \sin(10x)\cos(5y)\sin(10z)$ on the right.

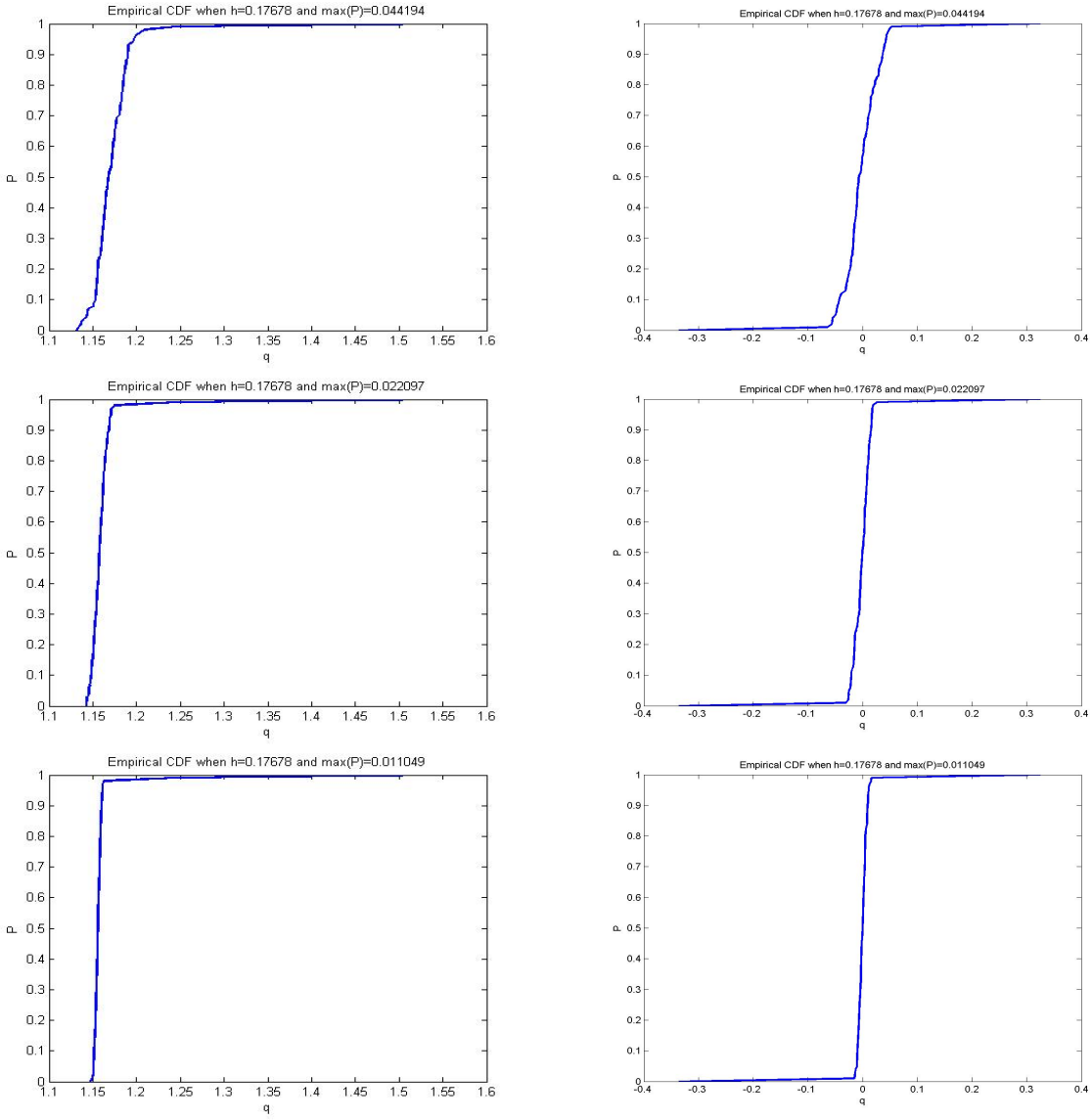

Fig. 8.6: More results for the mesh after two refinements with $f = z^2$ on the left column and $f = \sin(10x)\cos(5y)\sin(10z)$ on the right column. We can see the Cumulative Density Function becoming sharper and sharper as the maximum perturbation decreases.

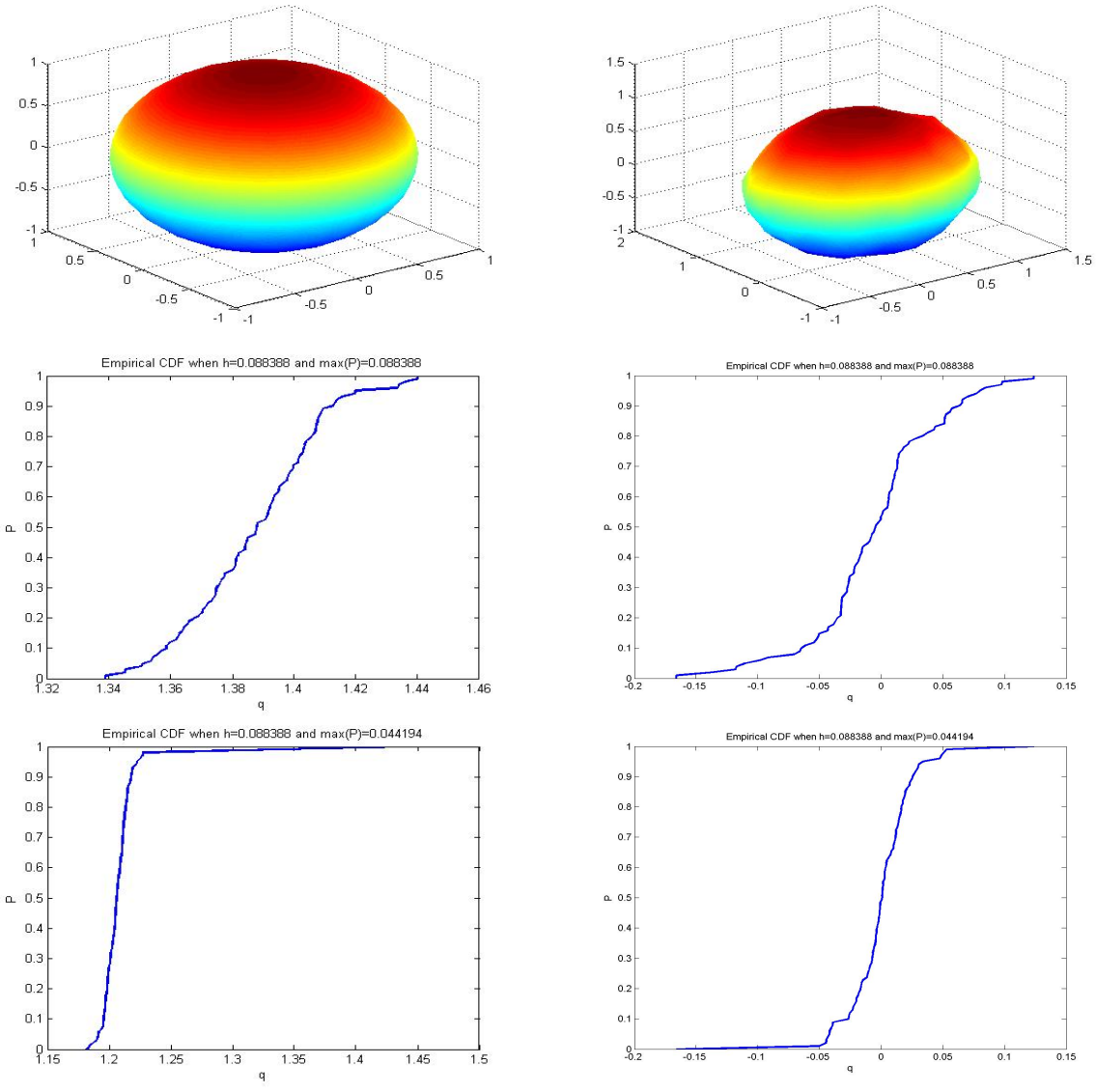

Fig. 8.7: Results on the mesh after three refinements. In the top row, we have a diagram of the mesh and an example of a perturbed mesh. On the second and third rows, we have the Cumulative Density Function for q with $f = z^2$ on the left and $f = \sin(10x)\cos(5y)\sin(10z)$ on the right.

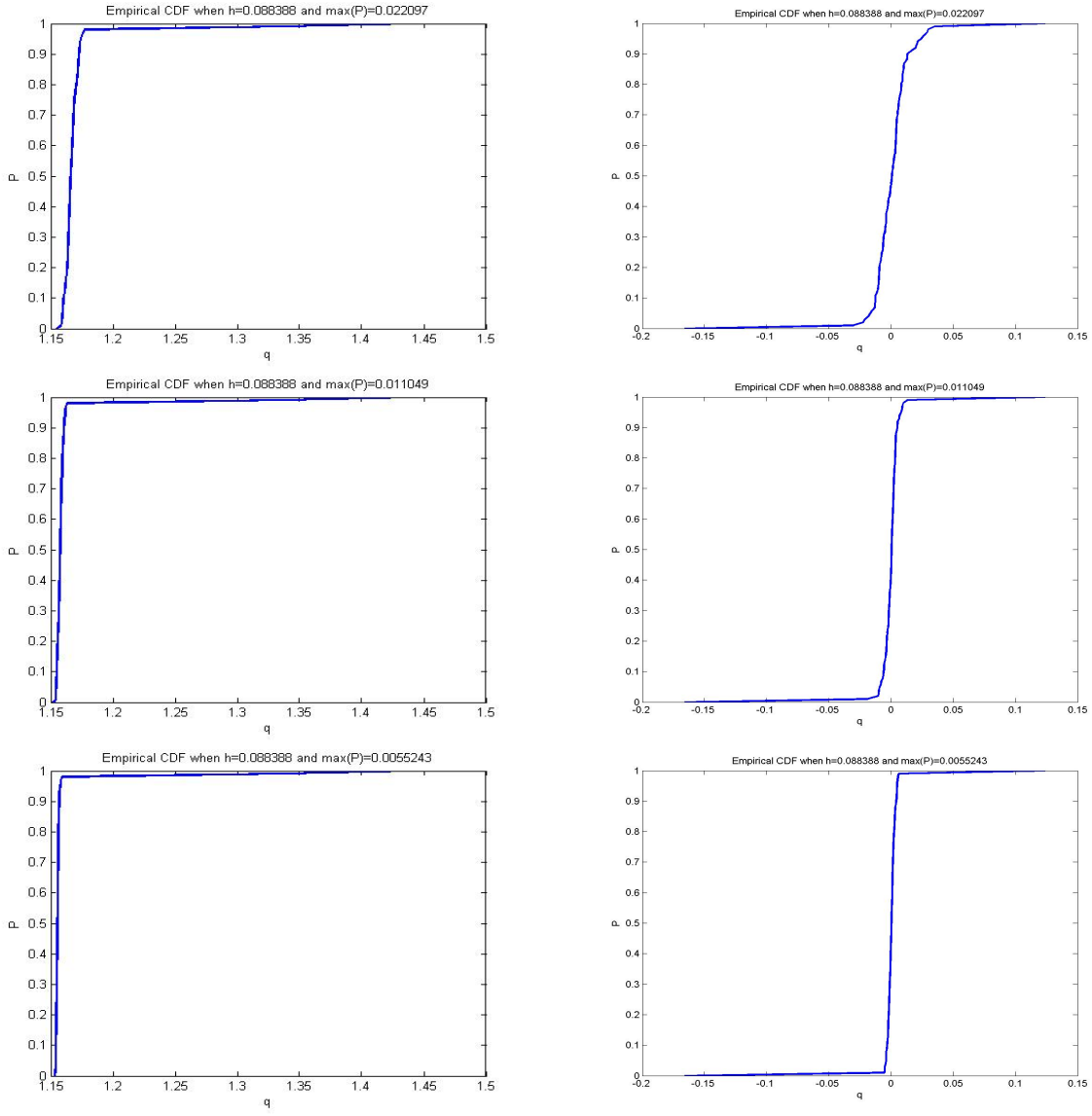

Fig. 8.8: More results for the mesh after three refinements with $f = z^2$ on the left column and $f = \sin(10x)\cos(5y)\sin(10z)$ on the right column. We can see the Cumulative Density Function becoming sharper and sharper as the maximum perturbation decreases.

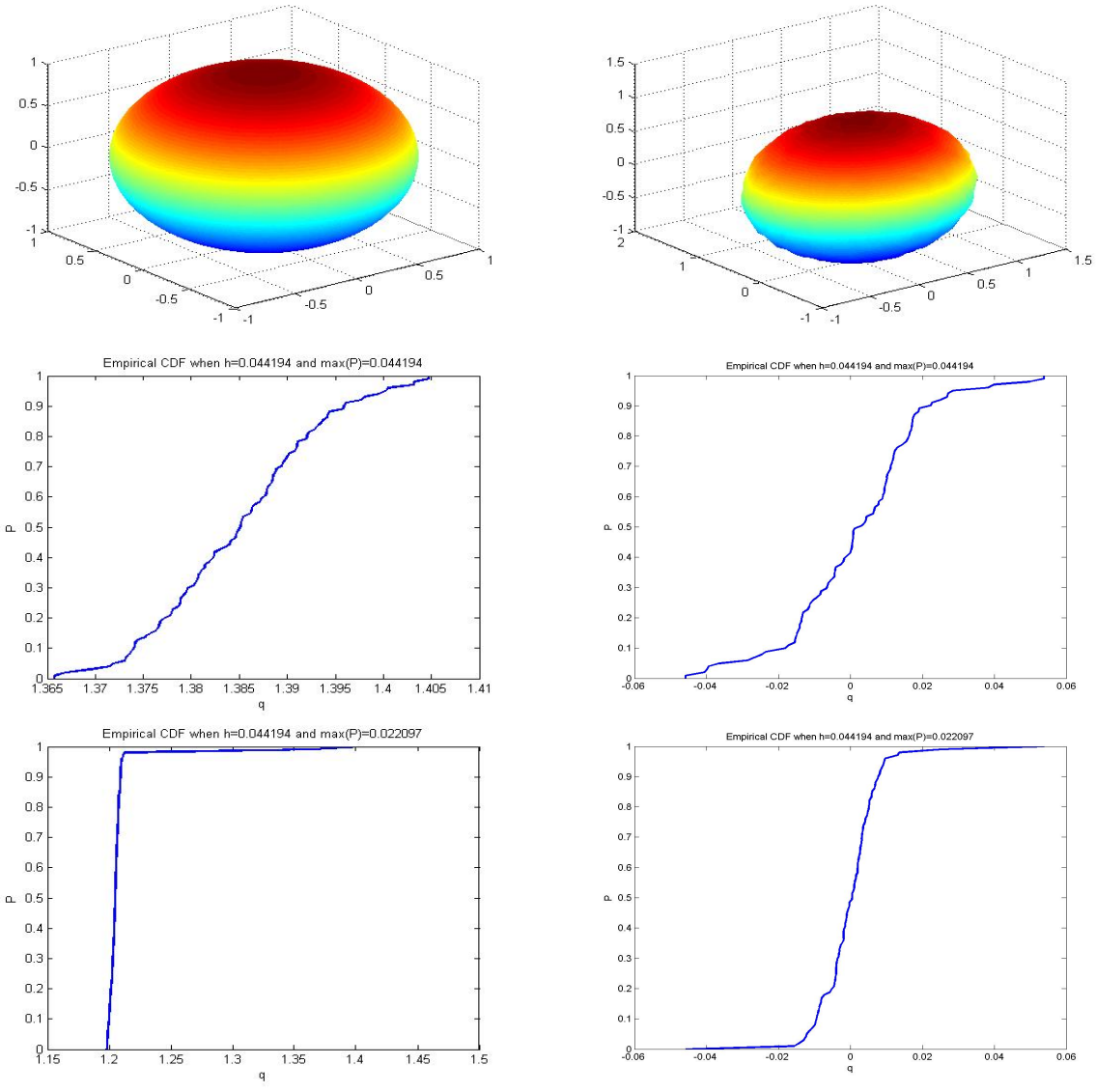

Fig. 8.9: Results on the mesh after four refinements. In the top row, we have a diagram of the mesh and an example of a perturbed mesh. On the second and third rows, we have the Cumulative Density Function for q with $f = z^2$ on the left and $f = \sin(10x)\cos(5y)\sin(10z)$ on the right.

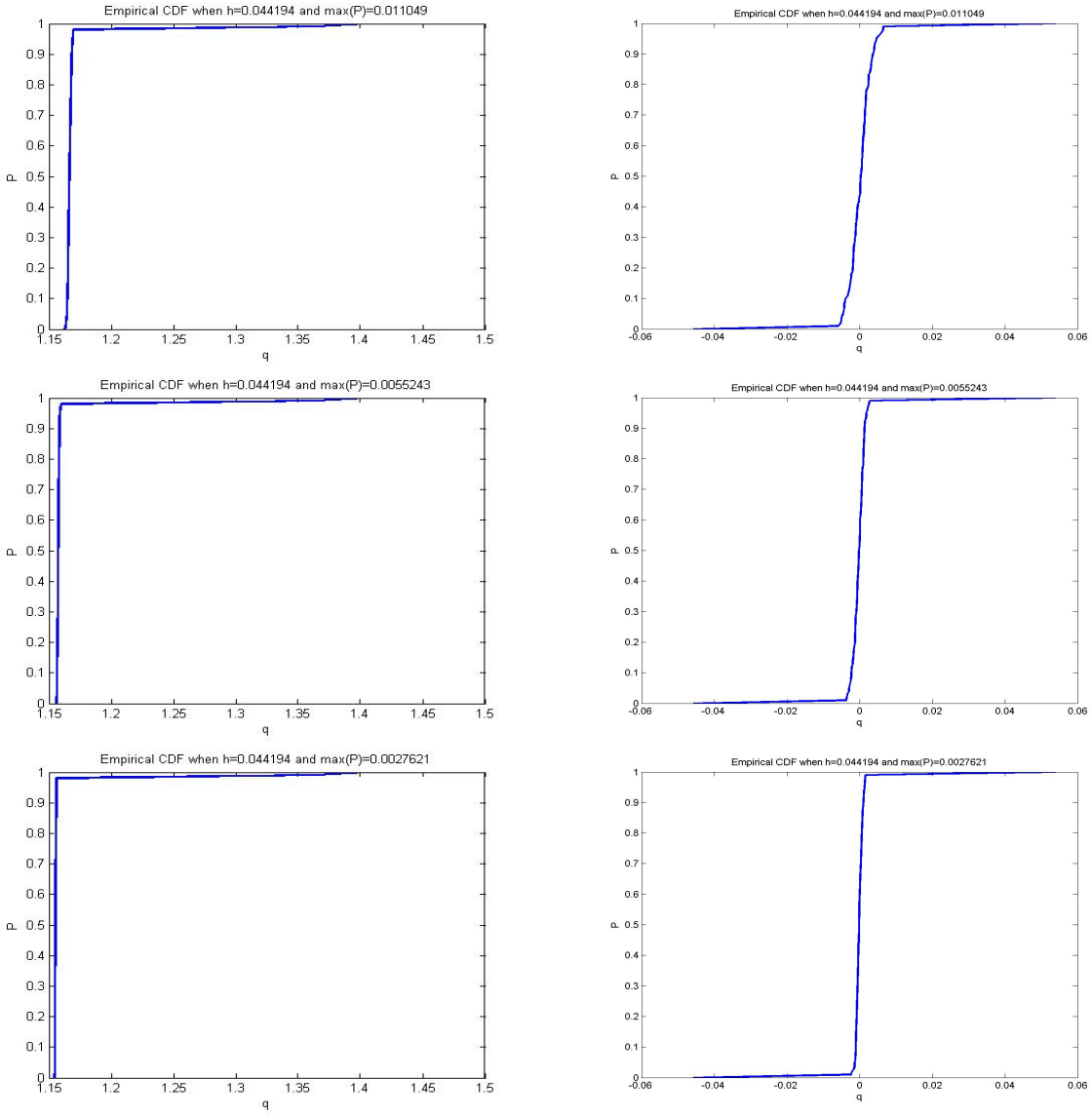

Fig. 8.10: More results for the mesh after four refinements with $f = z^2$ on the left column and $f = \sin(10x)\cos(5y)\sin(10z)$ on the right column. We can see the Cumulative Density Function becoming sharper and sharper as the maximum perturbation decreases.

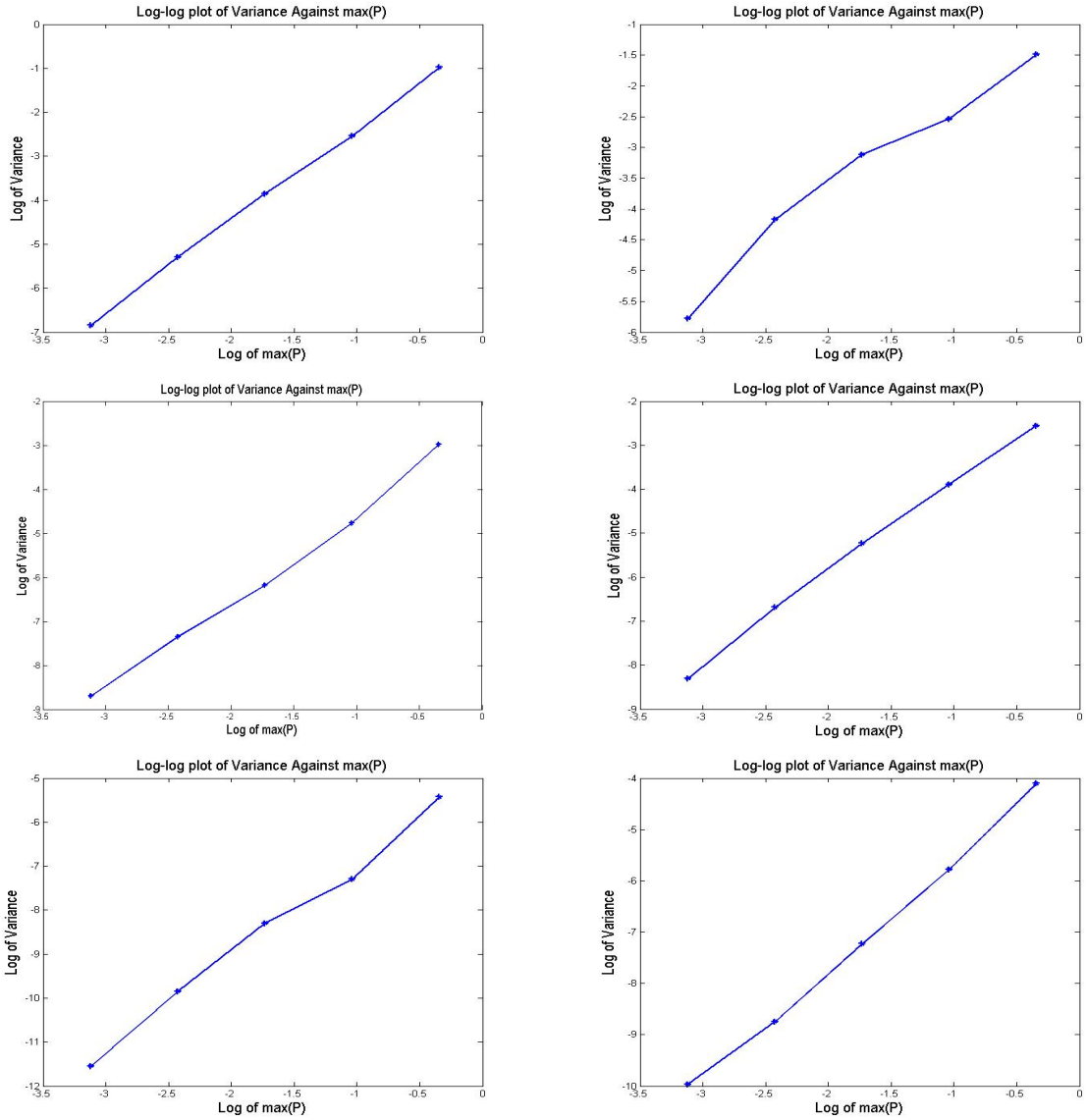

Fig. 8.11: Log-log plots of the variance against the maximum perturbation for the first three meshes. The results for $f = z^2$ are on the left and the results for $f = \sin(10x)\cos(5y)\sin(10z)$ are on the right. Each graph is for the same mesh. At each data point, we have used a different maximum perturbation and show the variance with 100 samples taken using that maximum perturbation.

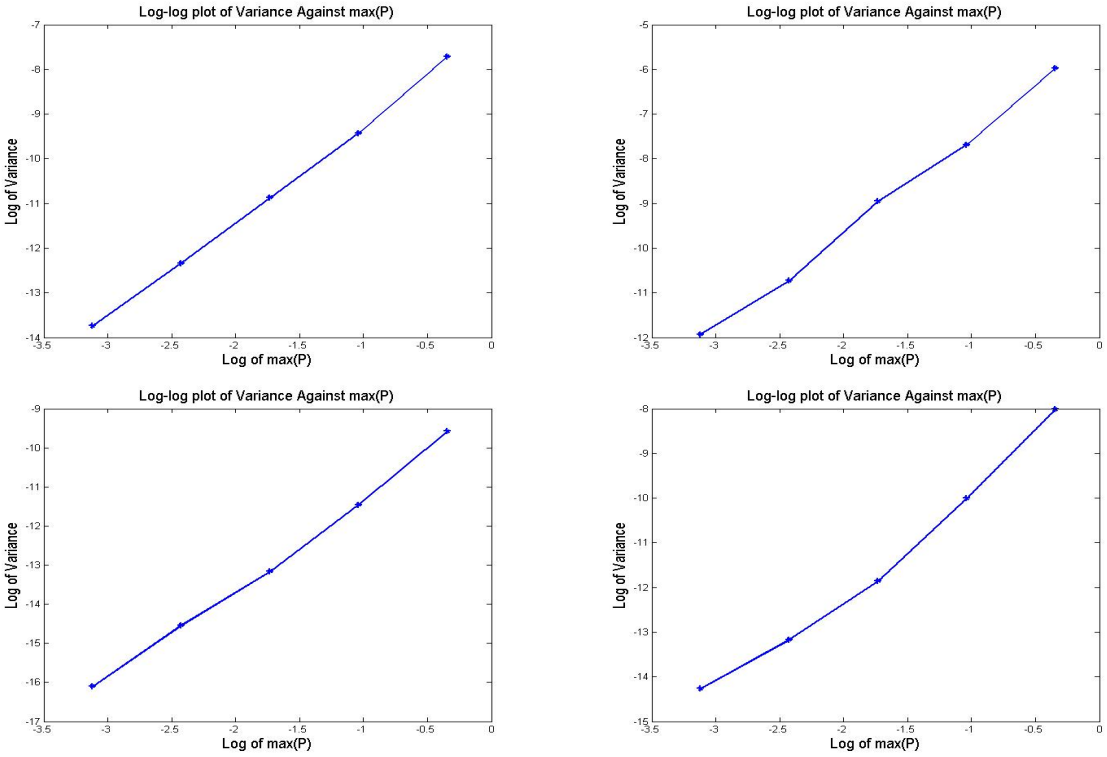

Fig. 8.12: Log-log plots of the variance against the maximum perturbation for the second two meshes. The results for $f = z^2$ are on the left and the results for $f = \sin(10x)\cos(5y)\sin(10z)$ are on the right.

8.2 Implementation on a Complex Biological Molecule

In this section, we implement the same method on a model of a very complicated biological molecule. The surface is shown in Figure 8.13.

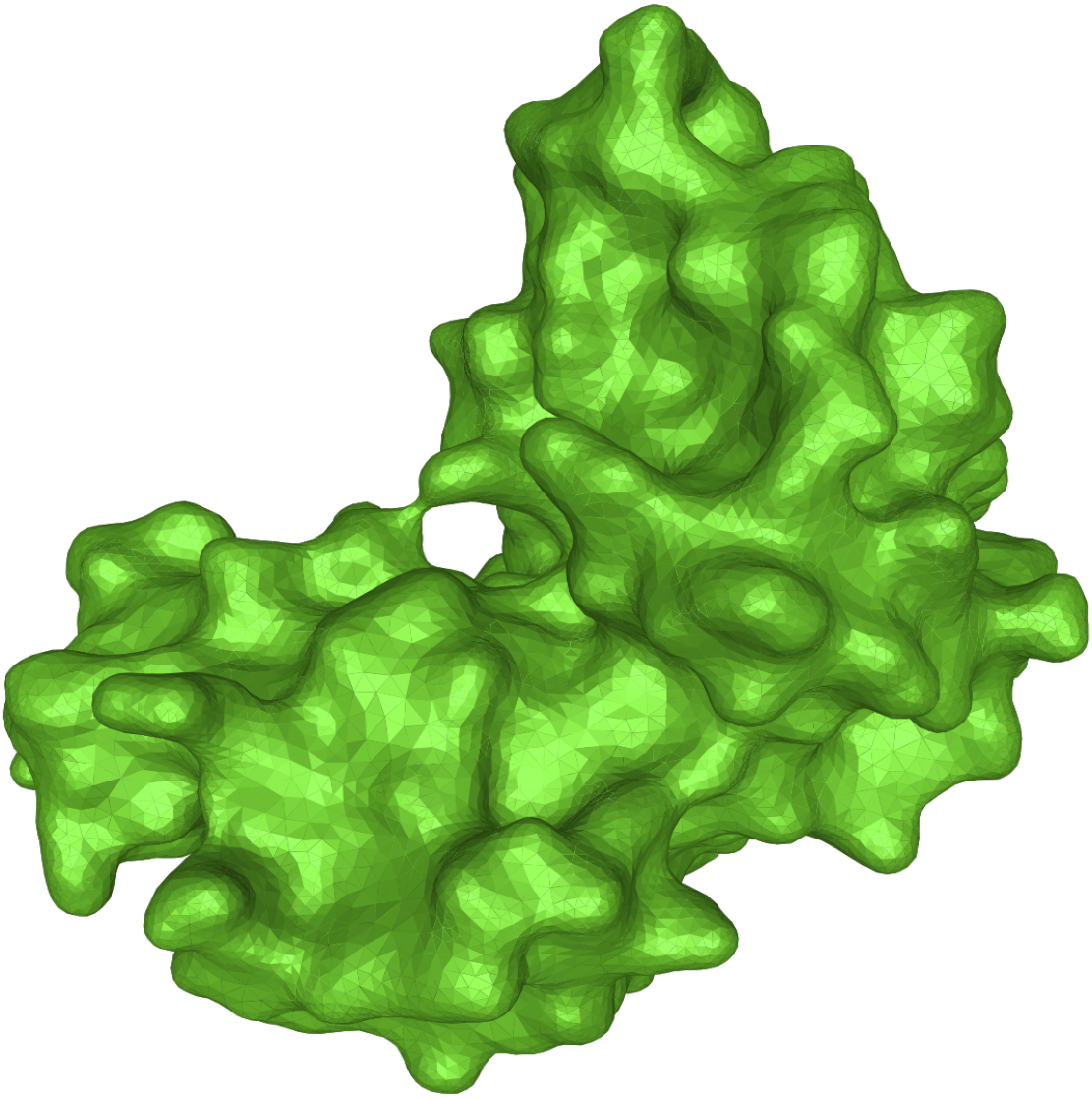

Fig. 8.13: Complex Biological Molecule

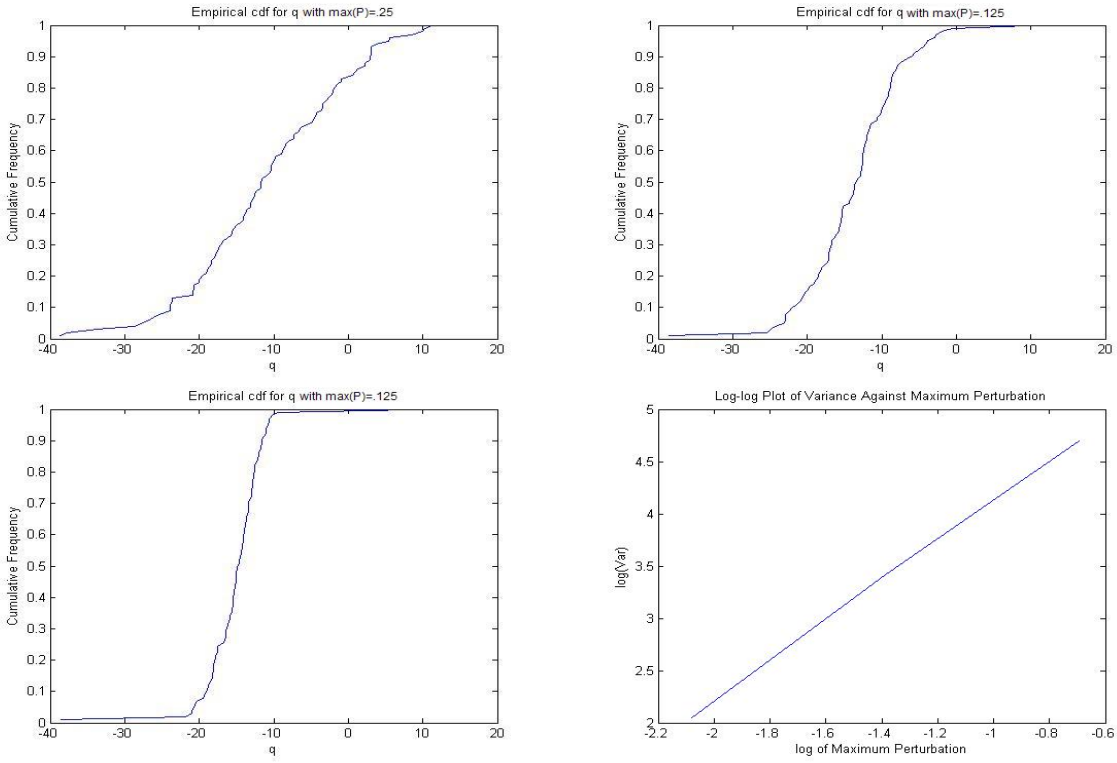

Fig. 8.14: Results of Monte Carlo on the Biological Molecule with $f = \sin(10x)\cos(5y)\sin(10z)$.

We use $\psi = 1$ and $f = \sin(10x)\cos(5y)\sin(10z)$. We solve the problem and approximate the quantity of interest for three different levels of perturbation. Once again, we see that the variance of the empirical distribution goes to zero quadratically with the maximum perturbation. Again, we have quadratic convergence with the maximum perturbation.

8.3 Discussion

Besides all of the deterministic components of error we have discussed in the previous chapters, measurement error always plays an important role when solving a Poisson problem on a surface. It is important to have accurate measurements of the surface in order to have an accurate solution. We can use the Monte Carlo approach we have just considered to represent our uncertainty about the surface. The error in the quantity of interest seems to go to zero quadratically with the level of uncertainty regarding the surface.

9. CONCLUSIONS

We have reviewed the background material necessary for the finite element method for the Poisson Problem on a closed, bounded two-dimensional surface. Our main result is the adjoint-based a posteriori estimate. This estimate measures the amount of error arsing from geometric error, discretization error, quadrature error and measurement error. We have shown how to implement the method and error estimate and shown examples where each type of error is significant.

The finite element method for the problem works by approximating the surface with a simpler piecewise flat surface. This avoids the problem of needing to estimate the First Fundamental Form of the Surface directly. As the approximation gets finer, the map between the approximate surface and the true one becomes closer to an isometry and the Laplacian for the approximate surface converges to the Laplace-Beltrami Operator for the true surface.

Estimating error and how much arises from what source is important for adaptive mesh refinement. It is also useful to know when each type of error is significant. As we have seen, geometric error is low for a surface that is close to being flat and high for a surface with high curvature. Discretization error depends on how close the forcing terms f and ψ are to the space of functions where we seek a solution. Quadrature error can be significant when f or ψ have large derivatves or discontinuities. Measurement error is always a factor and seems to depend quadratically on the uncertainty in the measurements of the surface. The use of the adjoint allows us to focus on the quantity of interest that we are computing from the solution and avoid unnecessary effort that may be necessary if we look at the accuracy of the entire solution.

Accurate estimates of the error will be important when extending this method to time-dependent diffusion problems on surfaces or to problems where the surface itself is evolving.

BIBLIOGRAPHY

[1] T. Apel and C. Pester. 2005. *Clement-type Interpolation on Spherical Domains – Interpolation Error Estimates and Application to a Posteriori Error Estimation*. IMA J. Numer. Anal. vol 25. pp. 310-336.

[2] J.R. Baumgartner and P.O. Frederickson. 1985. *Icosahedral Discretization of the Two-Sphere*. SIAM J. Numer. Anal. vol 22. pp. 1107-1115.

[3] M.A.J. Chaplain, M. Ganesh and I.G. Graham. 2001. *S*patio-temporal Pattern Formation on Spherical Surfaces: Numerical Simulation and Application to Solid Tumour Growth. J. Math. Biol. vol 42. pp 387-423.

[4] L. Cheng. 2000. *The Level Set Method Applied to Geometrically Based Motion, Materials Science, and Image Processing*. Ph.D. Thesis. UCLA.

[5] L. Cheng, P. Burchard, B. Merriman and S. Osher. 2002. *Motion of Curves Constrained on Surfaces Using a Level Set Approach*. J. Comput. Phys. vol 175. pp. 604-644.

[6] D.L. Chopp. *Flow Under Geodesic Curvature.* CAM Report 92-23 (UCLA, 1992).

[7] K. Deckelnick and G. Dziuk. 1995. *Convergence of a Finite Element Method for Non-Parametric Mean Curvature Flow*. Numer. Math. vol 72. pp. 197-222.

[8] K. Decklenick and G. Dziuk. 2003. *Numerical Approximation of Mean Curvature Flow of Graphs and Level Sets*. in Mathematical Aspects of Evolving Interfaces, Lecture Notes in Math. 1812, Springer, Berlin 2003, pp.53-87.

[9] K. Deckelnick, G. Dziuk and C.M. Elliott . 2005. *Computation of Geometric Partial Differential Equations and Mean Curavture Flow.* Acta Numer. vol 14. pp. 139-232.

[10] A. Demlow. 2009. *Higher-order Finite Element Methods and Pointwise Error Estimates for Elliptic Problems on Surfaces*. SIAM J. Numer. Anal. vol 47. pp. 805-827.

[11] A. Demlow and G. Dziuk. 2007. *An Adaptive Finite Element Method for the Laplace-Beltrami Operator on Implicitly Defined Surfaces*. SIAM J. Numer. Anal. vol 45. pp. 421-442.

[12] G. Dziuk. 1988. *Finite Elements for the Beltrami Operator on Arbitrary Surfaces.* in Partial Differential Quations and Calculus of Variations, Lecture Notes in Math. 1357, Springer, Berlin, 1988, pp. 142-155.

[13] G. Dziuk. 1991. *An Algorithm for Evolutionary Surfaces*. Numer. Math. vol 58. pp. 603-611.

[14] G. Dziuk and C.M. Elliott. 2007. *Finite Elements on Evolving Surfaces*. IMA J. Numer. Anal. vol 27. pp. 262-292.

[15] G. Dziuk and J.E. Hutchinson. 2006. *Finite Element Approximations to Surfaces of Prescribed Variable Mean Curvature*. Numer. Math. vol 102. pp. 611-648.

[16] D. Estep, M. Larson and R. Williams. 2000. *Estimating the Error of Numerical Solutions of Systems of Reaction-Diffusion Equations*. Memoirs of the American Mathematical Society. Vol 146. no. 696.

[17] D. Estep and D. Neckels. 2000. *Fast and Reliable Methods for Determining the Evolution of Uncertain Paramters in Differential Equations*. Submitted to Journal on Computational Physics, download at <http://www.math.colostate.edu/∼estep>.

[18] I.M. Gelfand and S.V. Fomin. *Calculus of Variations.* Dover Publications. Mineola, New York. (1991)

[19] M. Holst. 2001. *Adaptive Numerical Treatment of Elliptic Systems on Manifolds.* Adv. Comput. Math. vol 15. pp. 139-191.

[20] M. Holst and A. Stern. 2010. *Geometric Variational Crimes: Hilbert Complexes, Finite Element Exterior Calculus, and Problems on Hypersurfaces.* Submitted for publication.

[21] M. Holst and A. Stern. 2010. *Semilinear Mixed Problems on Hilbert Complexes and Their Numerical Approximation.* Submitted for publication.

[22] R. Kimmel. 1997. *Intrinsic scale space for images on surfaces: The geodesic curvature flow*. Graph. Models Image Process. vol 59. p. 365.

[23] E. Kreyszig. *Differential Geometry*. Dover Publications. Mineola, New York. (1991)

[24] S. Larsson and V. Thomee. *Partial Differential Equations with Numerical Methods.* Springer-Verlag Heidelberg, Germany. (2005)

[25] W. F. Morris and D. F. Doak. *Quantitative Conservation Biology: Theory and Practice of Population Viability Analysis*. Sinauer Associates, Inc. Sunderland, MS (2002)

[26] J.C. Nédélec. 1976. *Curved Finite Element Methods for the Solution of Singular Integral Equations on Surfaces in* $\mathbb{R}^3$. Comput. Methods Appl. Mech. Engrg. vol 8. pp. 61-80.

[27] J.C. Nédélec. 1986. *A New Family of Mixed Finite Elements in* $\mathbb{R}^3$. Numer. Math. vol 50. pp. 57-81.

[28] J.C. Nédélec. 1980. *Mixed Finite Elements in* $\mathbb{R}^3$. Numer. Math. vol 35. pp. 315-341.