Numerical Study of Discrete Laplace-Beltrami

Hung D. Nong

Mentor: Jerrold E. Marsden
Anil N. Hirani

SURF 2004
California Institute of Technology

Pasadena, October 19, 2004

Abstract

We study the numerical behavior of a discrete Laplace-Beltrami operator on planar and non-planar triangle meshes. The Laplace-Beltrami operator is a generalization of the usual Laplacian of flat space to arbitrary Riemannian manifolds. It has been studied and used extensively in image processing, signal processing, surface processing, physics, and differential geometry. The discretization we study was proposed in Hirani [2003] as part of the general framework of discrete exterior calculus (DEC), and it had appeared in literature before as well. We show numerical evidence that the solutions to Poisson's equation (using the discrete Laplace-Beltrami) converge. We present this evidence for regular and irregular planar triangle meshes, and for triangle meshes approximating a sphere. For the spherical meshes, in most cases, we show evidence of convergence even when the vertices are perturbed away from the approximated sphere. The main point of this paper is as follows : Xu [2003, 2004]; Garimella and Swartz [2003] have studied Δu on a plane and cylinder. They showed numerical evidence that the error made in computing Δu goes to 0 as meshes are refined, if the mesh is regular (all triangles equilateral) and does not go to 0 for irregular meshes. We study $\Delta u = f$ with f known and u unknown. We show that the error made in computing u *does* go to 0 for regular as well as irregular meshes. This is in agreement with the recent theoretical work of Hildebrandt and Wardetzky [2004] whose work inspired us to do this numerical study. Therefore, the discrete Laplace-Beltrami we study appears to be suitable for use in numerical methods for partial differential equations.

1 Introduction

Discrete Exterior Calculus (DEC), the discrete version of the smooth exterior calculus, is a formulation of numerical methods for solving partial differential equations (PDEs) on piecewise linear manifolds, such as triangle meshes embedded in $\mathbb{R}^3$. See Hirani [2003] for details. DEC defines discrete objects and discrete differential operators, which correspond to objects and operators of exterior calculus. The operators of DEC satisfy many of the properties that their smooth counterparts do. In this project, we study numerical properties of one of these discrete differential operators.

We study the discrete Laplace-Beltrami operator of DEC. The Laplace-Beltrami operator is a generalization of the usual Laplacian of flat space to arbitrary Riemannian manifolds. It has been studied and used extensively in image processing, signal processing, surface processing, physics, and in differential geometry. Discretization of Laplacian on flat regular grids is of course well understood. For example, one just replaces the second derivatives by finite difference approximations. However, for non-planar triangle meshes embedded in $\mathbb{R}^3$, discretizations of Laplace-Beltrami were not well known until recently. Our main aim in this paper is to present our numerical experiments on the use of discrete Laplace-Beltrami for Poisson's equation $\Delta u = f$ on planar and non-planar triangle meshes. In addition, we also repeat some of the experiments of Xu [2003], who studied the use of discrete Laplace-Beltrami for discretizing Δu for known u (as opposed to our main focus on discretizing $\Delta u = f$ for *unknown* u and known f).

1.1 Previous Work

The first useful discretization of Laplace-Beltrami on non-planar triangle meshes appears to have been done by Pinkall and Polthier [1993]. This appeared again in Meyer et al. [2002]. Then Hirani [2003] showed that this formula can be derived by using the discrete definitions of exterior derivative and Hodge star (two operators of exterior calculus). Xu [2003, 2004]; Garimella and Swartz [2003] studied this and other discretizations of the Laplace-Beltrami. Their focus was on computing error in Δu for known u. Thus, they were not solving a PDE. They showed numerical evidence that, if the discrete Laplace-Beltrami of DEC is used to discretize Δ, then the L^∞ norm error made in computing Δu goes to 0 as the mesh is refined if the mesh is regular (consisting of all equilateral triangles). They also showed that however, for even a slightly perturbed mesh (meaning that the triangles are slightly perturbed so that they are no longer congruent and the mesh becomes irregular), the error does not converge to 0. In repeating some of their experiments, we made a small change which is that we study the discrete Laplace-Beltrami directly, where as they studied it indirectly, via discrete curvature.

1.2 Our Methodology

We work with simple surfaces, namely a plane and a sphere, and define smooth functions on these for which we can compute a closed form smooth Laplace-Beltrami. We then discretize the surfaces, by approximating them by triangle meshes. Sampling a smooth function at the mesh vertices gives a discrete function. For computing the L^2 and pointwise error, we compute the discrete Laplace-Beltrami of this discrete function at the vertices and compare it to the value of the smooth Laplace-Beltrami at the vertices. In another study, which is the main focus of our study, we study the pointwise error and L^2 error made in solving Poisson's equation, by comparing the discrete computation with the analytical solution. We study results on both a planar and a spherical mesh. The computations are done for a sequence of meshes, in which the triangles become smaller. For the planar mesh, we also study error as the regularity of mesh is varied at a fixed refinement level, by distorting triangles starting from equilateral. For spherical meshes we consider two types of refinement sequences. In one case, the mesh vertices are kept on the sphere, while in the other, the vertices are moved off the mesh by adding random noise vectors to positions of vertices.

As stated above, our main focus is to study the error made in numerically solving Poisson's equation on a planar and a spherical mesh at various refinements. A recently announced theoretical result by Hildebrandt and Wardetzky [2004] suggests that the error made in solving Poisson's equation decreases, as the mesh is refined, for the discretization of Laplace-Beltrami operator we study. To check this numerically, we consider a given function, apply the Laplace-Beltrami operator to obtain the right hand side, and then numerically solve the corresponding Poisson's equation with this obtained right hand side for a sequence of refined meshes. Thus, since the solution is known, we can compare the numerically obtained function values, with the values of the original function at the mesh vertices.

1.3 Our Results

There are two main parts in results, for planar and spherical cases. For the planar case, we provide a simple Taylor series calculation which shows that the error in computing the solution to Poisson's equation is second-order with respect to the length of the edges if one uses a regular mesh (6 equilateral triangles around each vertex). This calculation also shows that for an irregular mesh the error is first-order. We also provide the numerical results which confirm the analytical proof and seem to give an even better convergence rate in some cases. For the spherical case, we provide numerical results which also suggest a convergence rate as the meshes are refined. As part of our experiments we created a piece of software documented below in Section 5, which can read a mesh, manipulate it in various ways, compute certain properties for the mesh, compute discrete Lapalce-Beltrami on the mesh, solve Poisson's equation on the mesh and so on.

2 Laplace-Beltrami Operator

Let M be a Riemannian manifold with metric g. The Laplace-Beltrami operator of a function $f : M \to \mathbb{R}$ can be computed using the following formula in Abraham et al. [1988]

$$\Delta f = |det[g_{ij}]|^{-\frac{1}{2}} \frac{\partial}{\partial x^k}(g^{lk}|det[g_{ij}]|^{\frac{1}{2}} \frac{\partial f}{\partial x^l}), \tag{1}$$

where g_{ij} are the entries of the matrix which represents the metric g in local coordinates x^i. In this paper, our experiments are on a sphere and a plane. For the plane, we use the formula for usual Laplacian. For the sphere, we use the formula for Laplacian in spherical coordinates after taking the radial coordinate r to be constant. The Laplacian in spherical coordinates is:

$$\Delta f(r,\theta,\phi) = \frac{1}{r^2}\frac{\partial}{\partial r}(r^2\frac{\partial f}{\partial r}) + \frac{1}{r^2 \sin\phi^2}\frac{\partial^2 f}{\partial \theta^2} + \frac{1}{r^2\sin\phi}\frac{\partial}{\partial\phi}(\sin\phi\frac{\partial f}{\partial\phi}). \tag{2}$$

Therefore, the Laplace-Beltrami of a real-valued function $f(\theta, \phi)$ on a unit sphere is given by:

$$\Delta f(\theta, \phi) = \frac{1}{\sin \phi^2} \frac{\partial^2 f}{\partial \theta^2} + \frac{1}{\sin \phi} \frac{\partial}{\partial \phi} (\sin \phi \frac{\partial f}{\partial \phi}). \tag{3}$$

If one does not follow the simple route we are taking which is just simply replacing r in (2) by a constant, but instead, one directly uses (1) with appropriate transformations, then one also obtain (3) for the sphere.

3 Discrete Laplace-Beltrami Operator

In this study, we have to compute the Laplace-Beltrami of a function that is sampled at vertices of a mesh. The mesh is a piecewise affine surface embedded in $\mathbb{R}^3$, in which each piece is a triangle. In practice, the smooth surface approximated by this mesh may not be known. Furthermore, the only information about the function may be its value given at the vertices. Thus, in general, we cannot use the formula for smooth Laplace-Beltrami, which requires the metric of the surface, and requires that the function be smooth. Thus, we must use a discrete Laplace-Beltrami. To compute the Laplace-Beltrami discretely, we use the formula (4) given below. The use of cotangents in this context was first described by Pinkall and Polthier [1993] and appeared again in Meyer et al. [2002]. Then Hirani [2003] showed that this formula can be derived by using the discrete definitions of exterior derivative and Hodge star. Assume that we are given a function f sampled at the vertices of the mesh. Let f_i be the sampled value at vertex i. Then

$$\Delta f_i = \frac{1}{2A} \sum_{j \in N_1(i)} (\cot \alpha_{ij} + \cot \beta_{ij})(f_i - f_j), \tag{4}$$

where α_{ij} and β_{ij} are specified in Figure 1, and A is the area of the circumcentric dual of vertex i. This is shown as shaded region around the central vertex in Figure 1(b). Another type of dual is called the barycentric dual, and this is shown in Figure 1(c). As pointed out by Hirani [2003], (4) is valid only if a circumcentric dual mesh is used. This may place a restriction on the triangles; namely, the triangles cannot have obtuse angles. In our study, we use a barycentric dual mesh as shown in Figure 1(c). This should imply the use of a formula other than (4); however, for this project, we still continue to use (4).

4 Error Types and Their Measurement

In this paper we study Poisson's equation $\Delta u = f$ on the plane and sphere. We first consider the case that u is known and f is unknown and then the case that f is known and u is unknown. When a triangle mesh is used to approximate the domain of interest, then one must discretize the operator Δ as well as the functions u and f.

Now let us consider the case that u is known and f is unknown. If u is known, then using (1), one can apply a smooth Laplace-Beltrami Δ operator on u to obtain Δu which is also f. On the other hand, one can instead, by using (4), apply the discrete Laplace-Beltrami Δ_d on u_d to obtain

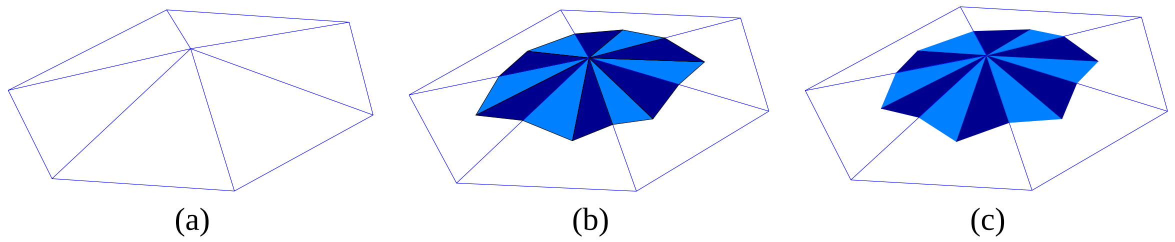

Figure 1: (a) shows a ring for which the discrete Laplace-Beltrami on central vertex is desired. This vertex is referred to as vertex i in (4). Consider a vertex j in the figure. Then the angles α_{ij} and β_{ij} are the two opposite angles in the two adjacent triangles to the edge ij (b) The shaded region around the central vertex is the circumcentric dual of that vertex. (c) The shaded region is the barycentric dual of the central vertex. In (b), the dual is made by starting from x_i going to the center of the edge containing x_i and then to the circumcenter of the triangle containg that edge. In (c), the circumcenter is replaced by the barycenter.

$\Delta_d u_d$ which is also f_d. From there, one can compute errors in using $\Delta_d u_d$ in replacement of Δu in different norms. The errors we consider here contain L^2 and L^∞.

For the case that u is unknown and f is known, it is not always possible to solve Poisson's equation analytically. However, we need to know the analytical solution so that we can measure the errors of the replacement of Δ by Δ_d. To get around this dilemma, we consider a known u, and then use a smooth Laplace-Beltrami to determine f, which means that if we start with this known f, we will get this corresponding smooth u for the analytical solution to Poisson's equation. Now using the same f and replacing the smooth Δ by Δ_d, one obtains a discrete version of Poisson's equation $\Delta_d u_d = f$. Solving this discrete system gives u_d. From there, one can compute errors in using u_d in replacement of u in different norms. Similarly, the errors we consider here contain L^2 and L^∞.

Like the usual definition, L^∞ norm error of Δu and $\Delta_d u_d$ (or u and u_d) is defined as $\|\Delta u - \Delta_d u_d\|_{L^\infty} = \max_i \|\Delta u_i - \Delta_d u_{di}\|$ (or $\|u - u_d\|_{L^\infty} = \max_i \|u_i - u_{di}\|$), where i runs over all the vertices of the mesh.

For L^2 norm error, let us consider the following. Consider two arbitrary functions $u_1, u_2 \in L^2(\mathbb{R})$. By definition, the L^2 norm of the difference of u_1 and u_2 is

$$\begin{aligned} \|u_1 - u_2\|^2_{L^2(\mathbb{R})} &= \int_{-\infty}^{\infty} (u_1 - u_2)^2 dx \\ &= \sum_i \int_{a_i}^{b_i} (u_1 - u_2)^2 dx \\ &= \sum_i (u_1^i - u_2^i)^2 h_i, \end{aligned} \tag{5}$$

where h_i is the region (length) of effectiveness of $u_1^i = u_1(x_i)$ and $u_2^i = u_2(x_i)$.

Thus, if a_i is the effective area of $u_1^i = u_1(x_i)$ and $u_2^i = u_2(x_i)$ in $\mathbb{R}^2$, then the L^2 difference of u_1 and u_2 in $\mathbb{R}^2$ is

$$\|u_1 - u_2\|^2_{L^2(\mathbb{R}^2)} = \int_{R^2} (u_1 - u_2)^2 da$$

$$
\begin{aligned}
&= \sum_i \int_{A_i} (u_1 - u_2)^2 da \\
&= \sum_i (u_1^i - u_2^i)^2 a_i \qquad (6)
\end{aligned}
$$

The effective areas here are taken to be the duals of the vertices of the given mesh. One can find an example of the dual of a vertex in Figure 1(c).

However, since the effective area of a vertex on the given mesh and that of the same vertex on the corresponding smooth surface are different, the L^2 norm error computed by (6) is not accurate. We fix this problem by dividing the whole right hand side of (6) by the the square root of the total area of the mesh. In other words, the effective area is now taken to be unitless, by dividing it by the total area. Therefore, the L^2 norm error of u_1 and u_2 in $\mathbb{R}^2$ (or a surface embedded in $\mathbb{R}^3$) is

$$
\|u_1 - u_2\|_{L^2} = \sqrt{\frac{\sum_i (u_1^i - u_2^i)^2 a_i}{A}}\,. \qquad (7)
$$

Actually this is really the L^2 error, normalized by the square root of the total area, but for the remainder of the paper we will call this the L^2 error.

5 laBel

5.1 Brief Description

In order to aid the numerical studies, we write a software application called laBel in C++ which contains a package of classes and utility functions. The classes specify and capture the ingredients of a given mesh. They hold the information of and the inter-relationship between triangles, edges as well as vertices which form the mesh. The utility functions take responsibilites on performing any given tasks which users wish to do, including from manipulating the mesh, computing Laplace-Beltrami, forming and solving linear systems of equations to measuring appropriate errors.

For solving linear systems of equations, laBel calls to PETSc (Portable, Extensive Toolkit for Scientific Computation). Specifically, laBel obtains information of a given mesh from an input file, builds the corresponding matrix and right hand side, and then pass these two elements to PETSc for PETSc to choose an ultimate and appropriate method to solve the linear system. PETSc then returns the solution back to laBel.

5.2 How to Use laBel

There are five main options for laBel: `-e, -i, -p, -r, -s`, where choosing `-e` results in error computations, `-i` mesh irregularity computation, `-p` mesh perturbation, `-r` mesh rotation, `-s` Poisson's system formulation.

To demonstrate how to use laBel, let us consider the following sequence of calls to laBel. Suppose that we have the following two object files, `oneRing.obj` and `tet.obj`, where `oneRing.obj` specifies a triangle mesh on a plane and `tet.obj` a triangle mesh on a sphere. Therefore, `oneRing.obj` is a $2D$ mesh, while `tet.obj` a $3D$ embedded mesh.

For option `-e`, one can make the following calls to laBel:

```
lb -e l2f f1cart < oneRing.obj
lb -e linfu p f2spher < tet.obj > result.txt,
```

where `l2f` and `linfu` are L^2 norm in Δu (or in f) and L^∞ norm in u errors, f and u taken from Poisson's equation $\Delta u = f$. The other choices for this norm error are `linff` and `l2u`. `p` refers to choosing PETSc for a linear solver application. The other choice for this option is `l`, which stands for LAPACK (Linear Algebra PACKage). One can see that, as for `l2u` and `linfu`, we need to solve a linear system prior to performing any error computation. Therefore, a choice for a linear solver application needs to be specified. This is not the case however if `l2f` or `linff` is chosen. `f1cart` and `f2spher` are two functions one wants to work with. The suffix `cart` is due to `oneRing.obj` being in Cartesian coordinates, `spher` due to `tet.obj` spherical coordinates. There are more choices for this option as well. At the moment, as we are working with six functions, one can choose one from these six functions. `result.txt` is an output file, and it is optional.

The generic call for the option `-e` is

```
lb -e norm [p - if l2u or linfu is chosen] function < inputFilename
[> outputFilename].
```

For option `-i`, one can make the following calls to laBel:

```
lb -i < oneRing.obj
lb -i < tet.obj.
```

There is no special specification for this case. laBel basically computes the irregularity of the mesh(es).

The generic call for the option `-i` is

`lb -i < inputFilename`. (Side note: For this project, we do not use option `-i`.)

For option `-p`, one can make the following calls to laBel:

```
lb -p 2 0.1 0 0 1 < oneRing.obj > oneRingPerturbed.obj
lb -p 3 0.2 2 1 0 < tet.obj,
```

where the first number (`2` in the first call or `3` in the second call) specifies the dimension of the mesh (`oneRing.obj` is in $2D$, `tet.obj` in $3D$). The second number (`0.1` or `0.2`) represents a percentage (10% or 20%, correspondingly) of the length of the average edge in the one-ring about a vertex. laBel will use this percentage as a maximum magnitude to perturb the coordinates of the vertex. The noise is determined randomly to a limit which is this magnitude with a use of the function `rand()` in C. The seed to determine a sequence of random numbers is specified by the third number in the call (`0` in the first call or `2` in the second call). Unfortunately, this perturbation might destroy the topology of the mesh. Therefore, in order to prevent that from happenning, we perform a little test to see if the topology of the mesh is preserved. For the planar case, we check to see if the central vertex is still inside its one-ring. If it is not, then we take a different random number which gives a different noise. This noise will replace the previous one for the vertex's perturbation. We again recheck to see if the topology of the mesh is intact. We continue doing this process until we get a noise which does not destroy the topology of the mesh. For the spherical case (in $3D$), there is no concrete concept of a vertex staying inside its one-ring. Therefore, in order to test to see if the topology of the mesh is preserved, for each one-ring, we project it on a plane which is orthogonal to the ray connecting the central vertex to the origin of the sphere. With simple reasonning, one can see that the topology of the one-ring on the sphere is the same as that of the one-ring on the planar. Then we perform the same test for the one-ring on the plane to see if the topology of the one-ring is preserved. Back to the call(s) to laBel, the fourth number (`0` or `1`) is irrelevant for $2D$ meshes (in this case, for `oneRing.obj`), but is needed for $3D$ embedded meshes, as it specifies whether or not the vertices of the resulting perturbed mesh need to be projected back on the smooth surface so that their projected coordinates are used for smooth function values computations. The last number (`1` in the first call or `0` in the second call) tells laBel to output the resulting perturbed mesh or not to. As for the first call, because `1` is chosen for the last number, meaning that the user wants laBel to output the resulting perturbed mesh, `oneRingPerturbed.obj` is the `outputFilename`.

The generic call for the option `-p` is

`lb -p dimension percentage seed projection flag < inputFilename [> outputFilename]`.

For option `-r`, one can make the following call to laBel:

`lb -r y 0.1 1 < tet.obj > tetRotated.obj`,

where `y` is the axis of rotation. `0.1` is the angle of rotation in radians. `1` is to output the resulting rotated mesh. One can rotate a mesh about any axis as long as one has to decompose this rotation into a sequence of rotations about the standard Cartesian axes. As we can see, this option is not relevant for $2D$ meshes.

The generic call for the option `-r` is

`lb -r axis angle flag < inputFilename [> outputFilename]`.

For option `-s`, one can make the following calls to laBel:

```
lb -s f2cart < oneRing.obj
lb -s f1spher < tet.obj > system.txt,
```

where `f2cart` or `f1spher` is the function one wants laBel to build a Poisson's system upon. The output of these calls are a matrix and a right hand side of a Poisson's system. The suffixes `cart` and `spher` are as above.

The generic call for the option `-s` is

`lb -s function < inputFilename [> outputFilename]`.

Therefore, the most generic call to laBel is as follows:

`lb option(s) < inputFilename [> outputFilename]`.

One convenient feature of laBel is that users can combine options in order. For example, users can make the following call to laBel

`lb -r y 0.1 0 -i -e l2u p f1sphere < tet.obj > result.txt`.

Upon receiving this call, laBel first rotates the mesh about the y axis an angle of 0.1 radian, next computes the irregularity of the resulting rotated mesh, and then computes the L^2 norm error of the solution to Poisson's equation with u being `f1spher`. Last, laBel returns any available output in `result.txt`.

6 Experiments and Results

We consider the following six functions:

$$u_1(x, y, z) = (x^2 + y^2 + z)\sin(x)\cos(y) \tag{8}$$

$$u_2(x, y) = x^2 + y^2 \tag{9}$$

$$u_3(x, y) = \sqrt{4 - (x - 0.5)^2 - (y - 0.5)^2} \tag{10}$$

$$u_4(x, y) = \tanh(9y - 9x) \tag{11}$$

$$u_5(x, y) = \frac{1.25 + \cos(5.4y)}{6 + 6(3x - 1)^2} \tag{12}$$

$$u_6(x, y) = \exp(-\frac{81}{16}[(x - 0.5)^2 + (y - 0.5)^2]) \tag{13}$$

where the last four functions (10), (11), (12) and (13) are taken from Xu [2003].

As mentioned in Section 2, in general, we may not know the smooth surface being approximated by a given triangle mesh. We may also not be given a smooth function, but only values on vertices of the mesh. However, in order to study convergence, we take meshes which triangulate a part of the plane, and meshes which approximate a sphere. We also take smooth functions on plane and sphere so that we know the Laplace-Beltrami of the functions in closed form. This allows us to compare actual values of the Laplace-Beltrami of the function with those computed by the discrete formula 4.

6.1 On Planar Meshes

On a plane, all the functions stay the same except the first function, which becomes

$$u_1(x, y) = (x^2 + y^2)\sin(x)\cos(y) \tag{14}$$

For the planar case, beside performing our own experiments with a different setup, we also repeat some experiments done by Xu [2003] in which he demonstrates that error in Δu goes to 0 for regular meshes and does not for irregular meshes as the meshes are refined. The setup of Xu [2003]'s experiments is as follows: He considers an array of simple regular one-ring meshes over a region on a plane and then keeps these meshes' positions fixed while shrinking them in. In addition to that, we also consider another different setup in which, we start with a regular one-ring mesh and then refine the mesh by subdividing the edges to make the mesh denser and denser.

In addition to study convergence in Δu, for our new study, we also study convergence in u, which is the solution to Poisson's equation. The following is an analytical proof to show that the L^∞ error in u is first-order with respect to the length of the edges for irregular meshes. The error appears to be higher order (at least second-order) for regular meshes.

Consider an arbitrary n-neighbor one-ring. Let us name the central vertex 0, the neighboring vertices i counterclockwise, where $i = 1..n$. Let us also make the following convention: For each vertex i, let θ_{il} and θ_{ir} be the two angles on its left and right according to the central vertex' perspective.

If one revisits the cotangent formula discretization (4), one has the following Poisson's equation expansion at the central vertex:

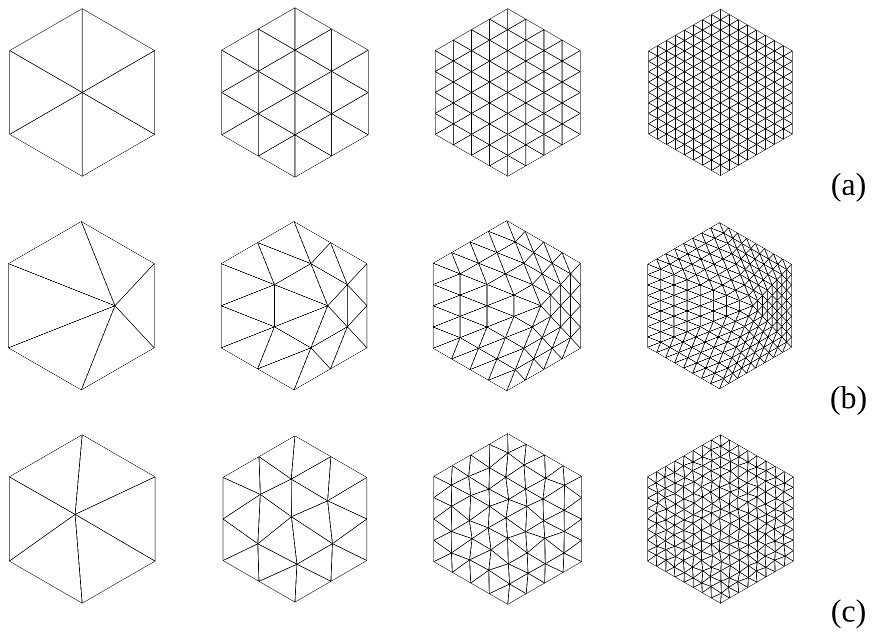

Figure 2: Examples of some of the sequences of meshes used to approximate a region of a plane. Each row shows part of a sequence. The two finest meshes for each sequence are not shown here. The first row (a) represents a sequence of refined meshes P_{d1}, whose base mesh is the first entry which is a regular one-ring mesh. Each subsequence mesh is obtained by subdivision of edges of the prior mesh. The base mesh for the second row (b) P_{d2} is an originally regular one-ring mesh with the central vertex being shifted away. The third row (c) represents another sequence of refined meshes P_{d3}. This sequence is obtained by randomly perturbing each subsequence mesh of the first sequence with a noise of 10% of the length of the edges maximum in the subsequence mesh.

$$\begin{aligned} f_0 &= \frac{1}{2A}((\cot\theta_{2r} + \cot\theta_{nl})(u_{0d} - u_{1d}) \\ &+ (\cot\theta_{3r} + \cot\theta_{1l})(u_{0d} - u_{2d}) \\ &+ (\cot\theta_{4r} + \cot\theta_{2l})(u_{0d} - u_{3d}) \\ &+ \dots \\ &+ (\cot\theta_{nr} + \cot\theta_{(n-2)l})(u_{0d} - u_{(n-1)d}) \\ &+ (\cot\theta_{1r} + \cot\theta_{(n-1)l})(u_{0d} - u_{nd})) \end{aligned} \tag{15}$$

Simplifying and rearranging the equation, one has

$$\begin{aligned} & u_{0d}(\cot\theta_{1l} + \cot\theta_{2l} + \dots + \cot\theta_{nl} + \cot\theta_{1r} + \cot\theta_{2r} + \dots + \cot\theta_{nr} \\ = \; & u_{1d}(\cot\theta_{2r} + \cot\theta_{nl}) + u_{2d}(\cot\theta_{3r} + \cot\theta_{1l}) + \dots + u_{nd}(\cot\theta_{1r} + \cot\theta_{(n-1)l}) \\ + \; & 2Af_0 \end{aligned} \tag{16}$$

Now let θ_i be the angle formed by the positive x-axis and the edge $(0, i)$, where the edge $(0, i)$ is

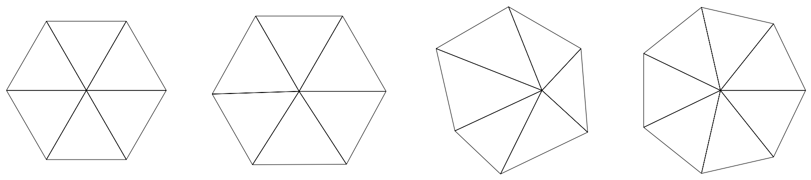

Figure 3: Each of these meshes is shifted over a region and then is shrunk. Xu [2003] used this mechanism to investigate the convergence study of a discrete Laplace-Beltrami. His meshes are the first two ones. We use the last two meshes for convergence study to see if the convergence property still holds for more agressive perturbed and non-congruent meshes.

formed by the central vertex and vertex i. Let r_i be the length of the edge $(0, i)$. Taylor expansion gives:

$$\begin{aligned} u_{id} &= u_0 + u_{0x} r_i \cos\theta_i + u_{0y} r_i \sin\theta_i + u_{0xx} r_i^2 \cos{\theta_i}^2 \\ &+ 2u_{0xy} r_i^2 \cos\theta_i \sin\theta_i + u_{0yy} r_i^2 \sin{\theta_i}^2 + O(r_i^3), \end{aligned} \tag{17}$$

where u_0 is the smooth function value at vertex 0, and u_{0x}, u_{0y}, u_{0xx}, u_{0xy} and u_{0yy} are partial derivatives of u with respect to x and y at vertex 0.

Substituting (17) into (16) gives:

$$\begin{aligned} & u_{0d}(\cot\theta_{1l} + \cot\theta_{2l} + \ldots + \cot\theta_{nl} + \cot\theta_{1r} + \cot\theta_{2r} + \ldots + \cot\theta_{nr} \\ = \;& u_0(\cot\theta_{1l} + \cot\theta_{2l} + \ldots + \cot\theta_{nl} + \cot\theta_{1r} + \cot\theta_{2r} + \ldots + \cot\theta_{nr}) + O(r_i), \end{aligned} \tag{18}$$

where we have used the fact that the area $A = O(r^2)$.

Therefore, $\|u_{0d} - u_0\| = O(r_i)$.

If the one-ring is regular, meaning that it has 6 neighbors and that all the triangles are equilaterals, then (15) becomes:

$$\begin{aligned} f_0 &= \frac{1}{2A}(\frac{2}{\sqrt{3}}(u_{0d} - u_{1d}) + \frac{2}{\sqrt{3}}(u_{0d} - u_{2d}) \\ &+ \frac{2}{\sqrt{3}}(u_{0d} - u_{3d}) + \frac{2}{\sqrt{3}}(u_{0d} - u_{4d}) \\ &+ \frac{2}{\sqrt{3}}(u_{0d} - u_{5d}) + \frac{2}{\sqrt{3}}(u_{0d} - u_{6d})) \end{aligned} \tag{19}$$

And also (17) becomes:

$$\begin{aligned} u_{id} &= u_0 + u_{0x} r \cos(\theta_1 + (i-1)\frac{\pi}{3}) + u_{0y} r \sin(\theta_1 + (i-1)\frac{\pi}{3}) \\ &+ u_{0xx} r^2 \cos(\theta_1 + (i-1)\frac{\pi}{3})^2 + 2u_{0xy} r^2 \cos(\theta_1 + (i-1)\frac{\pi}{3}) \sin(\theta_1 + (i-1)\frac{\pi}{3}) \end{aligned}$$

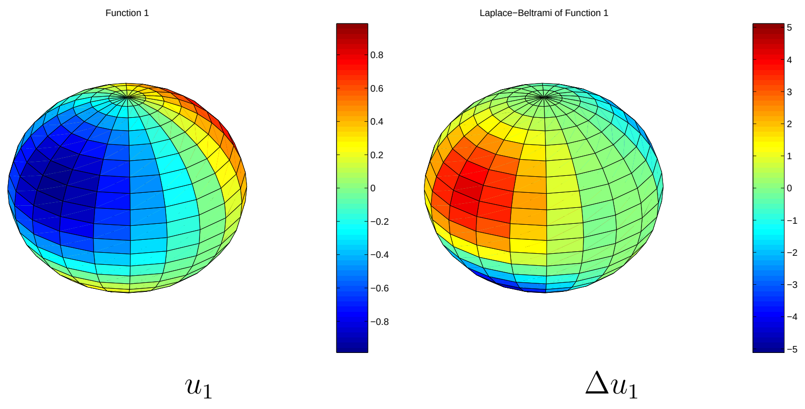

Figure 4: Function $u_1 : S^2 \to \mathbb{R}$ and its Laplace-Beltrami

$$+ \quad u_{0yy} r^2 \sin\left(\theta_1 + (i-1)\frac{\pi}{3}\right)^2 + O(r^3), \tag{20}$$

where $i = 1..6$ and r is the length of the edges.

Substituting (20) into (19) and simplifying, one obtains:

$$\|u_{0d} - u_0\| = O(r^2), \tag{21}$$

where we also have used the fact that the area $A = O(r^2)$.

To conduct numerical studies, we consider three different kinds of sequences of meshes: For the first kind, we consider a simple regular one-ring mesh. To form a sequence of meshes, we then keep refining it according to the scheme specified above. For the second kind, we start with the same simple regular one-ring mesh, but then we shift the central vertex away to form an irregular mesh. Using this irregular mesh, we keep refining it to form another sequence of meshes. For the third kind, we use the whole first sequence of meshes. However, for each mesh in the sequence, we add a random noise of 10% of the length of the edges maximum to each vertex of the mesh (except for the ones on the boundary). Refer to Figure 2 for the meshes.

Refer to Tables 1, 2 and 3 in Appendix B for numerical results of experiments with our setup. For Xu [2003]'s setup, beside repeating experiments done by Xu [2003], we also consider some even more extremely perturbed meshes to confirm his results. One can also find Tables 4, 5, 6 and 7 in Appendix B for these numerical results. Figure 3 shows the meshes we use to perform Xu [2003]'s experiments. The first two meshes are taken from Xu [2003].

6.2 On Spherical Meshes

For the spherical case, we consider only the first two functions (8) and (9). Their representations in spherical coordinates are as follows:

$$\begin{aligned} \tilde{u_1}(r,\theta,\phi) &= (r^2\cos(\theta)^2\sin(\phi)^2 + r^2\sin(\theta)^2\sin(\phi)^2 \\ &+ r\cos(\phi))\sin(r\cos(\theta)\sin(\phi))\cos(r\sin(\theta)\sin(\phi))\,. \end{aligned} \tag{22}$$

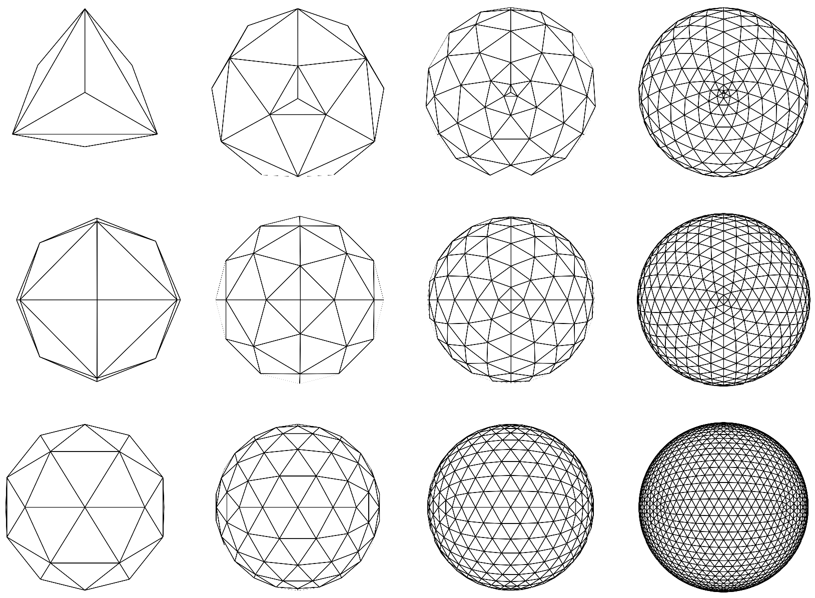

Figure 5: Examples of some of the sequences of meshes used to approximate a sphere. Each row shows part of a sequence. The finest meshes are not shown here. The first entry in each row was constructed from a polyhedron whose vertices lie on a sphere. We used a tetrahedron, octahedron and icosahedron as the starting mesh for the first, second and third rows respectively. Subsequent meshes in each row were obtained by subdivision of the triangles followed by normalization to unit sphere. The meshes in row 1 will be called the $S^2_{d,T}$ sequence, those in row 2 the $S^2_{d,O}$ sequence and the last row meshes will be called $S^2_{d,I}$ sequence. In computations we used one more level of refinement which we do not show here.

$$\tilde{u_2}(r,\theta,\phi) = r^2\cos(\theta)^2\sin(\phi)^2 + r^2\sin(\theta)^2\sin(\phi)^2 \,. \tag{23}$$

Restricting (22) and (23) to a unit sphere yields:

$$\begin{aligned}\tilde{u_1}|_{S^2}(\theta,\phi) &= (\cos(\theta)^2\sin(\phi)^2 + \sin(\theta)^2\sin(\phi)^2 \\ &+ \cos(\phi))\sin(\cos(\theta)\sin(\phi))\cos(\sin(\theta)\sin(\phi))\,. \end{aligned}\tag{24}$$

$$\tilde{u_2}|_{S^2}(\theta,\phi) = \cos(\theta)^2\sin(\phi)^2 + \sin(\theta)^2\sin(|phi)^2 \,. \tag{25}$$

In Appendix A, we give the expressions for the Laplacians of the specified functions (24) and (25) in spherical coordinates. Figure 4 shows the function (24) and its Laplace-Beltrami.

For the spherical case, we consider three different kinds of meshes on a unit sphere: $S^2_{d,T}$, $S^2_{d,O}$ and $S^2_{d,I}$, where $S^2_{d,T}$ is a mesh built and refined from a tetrahedron, $S^2_{d,O}$ an octahedron and $S^2_{d,I}$ an icosahedron. We refine these meshes and conduct a series of convergence studies of the Laplace-Beltrami of the functions (8) and (9) on these refined meshes in Figure 5.

For function (8), as there is a singularity at the vertex $(x,y,z) = (0,0,-1)$ or $(r,\theta,\phi) = (1,0,\pi)$, in order to avoid that, we rotate the whole mesh an angle of $0.1^0 = 0.00175rad$ about the y axis. In addition to these standard meshes, we also perform the same convergence studies on

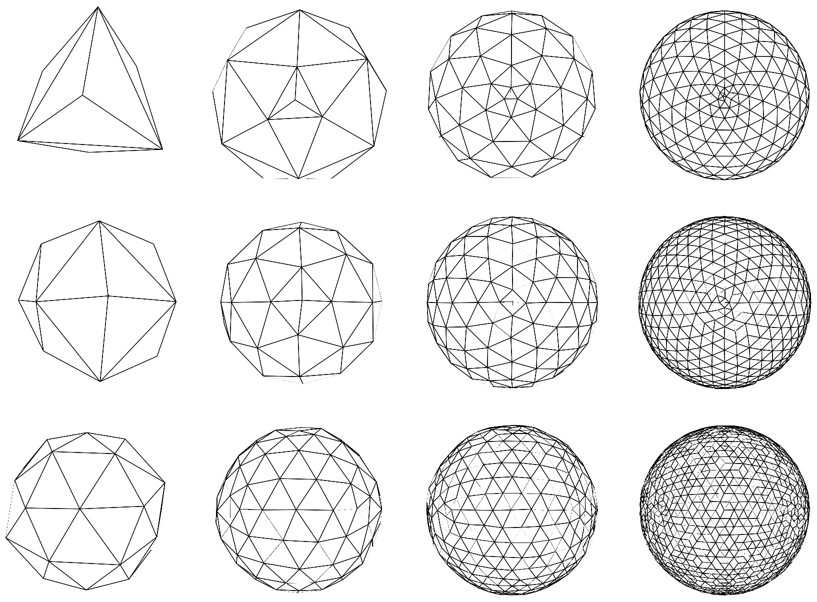

Figure 6: These meshes are similar to those in (5). The only difference is that each vertex of each mesh in this case has been randomly perturbed with a noise of a magnitude of 10% of the average edge of one ring about the vertex. The finest level of refinement used in our computations has not been shown here.

perturbed meshes which we obtain by adding some random noise to the vertices. The meshes are shown in Figure 6. Refer to Tables 8, 9, 10, 11, 12, and 13 in Appendix B for numerical results of the convergence studies. As representatives, we present the plots for some of the error studies in Figure 7.

7 Conclusions, Issues and Future Work

In addition to studying convergence behaviors of a discrete Laplace-Beltrami on different surfaces by looking at errors in Δu as Xu [2003, 2004]; Garimella and Swartz [2003] have done (even though we directly study a discrete Laplace-Beltrami; while they study it indirectly via discrete curvature), we also study convergence behaviors of a discrete Laplace-Beltrami by looking at errors in the solution u to Poisson's equation. The latter study is a more appropriate approach if one considers solving PDEs. For error in Δu, we basically obtain what Xu [2003, 2004]; Garimella and Swartz [2003] claim. However, there are a few features and trends of the error that we cannot explain (Refer to appropriate figures for further explanation). For error in u, our studies show analytical and numerical convergence evidence for a discrete Laplace-Beltrami for any kinds of planar meshes (regular or irregular) as the meshes get refined. We also show numerical convergence evidence for spherical meshes. As for spherical meshes whose vertices are perturbed off the spherical surface, in most cases, we also observe convergence.

For the planar case, numerical results seem to express a better convergence rate. One of the reasons to which we ascribe this feature is that, if one recalls our proof for the order of convergence rate in L^∞, one can see that the convergence rate can be second-order for irregular meshes if, for some special functions, the values of the first-order terms get canceled out mutually. Analogously, the convergence rate can be fourth-order for regular meshes if, for some special functions, the values of the second-order terms get canceled out mutually. Therefore, the convergence rate does depend on what function one works with. One more issue which we also think would effect the numerical convergence study of a discrete Laplace-Beltrami is that for a given linear solver application (PETSc or LAPACK), there is a certain built-within numerical error tolerance. This error might influence the actual error of a discrete and smooth values comparison. Moreover, the machine number precision also plays a role in contributing to the total error. One evidence of this is that, if one takes a look at $||u - u_d||_{L^\infty}$ for u_2 in Table 1, one can see that, as all the edges of the mesh are splitted in halves, the error should also decrease at a rate of at least 2 according to the analytical proof. However, this is not the case. These error numbers must be influenced by the machine number precision and/or the built-within numerical error tolerance of PETSc. Another issue is with very large linear systems obtained from the finest meshes. It would be best to solve these using iterative linear solvers. Currently, we do not know enough about PETSc to know what linear solver it is using. It would also be interesting to study the behavior of the condition numbers of the matrices obtained from discretization.

As a result, for future work, we would like to look for methods to isolate these influences. In addition to this, we would like to investigate some different features and trends in numerical errors for which we haven't had explanations. We also would like to find the convergence rate for the spherical case by presenting an analytical proof. For the software application laBel, we would like to add more options to make laBel more productive and powerful.

Acknowledgements I would like to thank my mentors, Prof. Jerrold Marsden at Caltech's Control & Dynamical Systems department and Dr. Anil Hirani at Caltech's Jet Propulsion Laboratory for their belief in me and for allowing me to be part of their research teams. Dr. Anil Hirani has encouraged and guided me all the way along the SURF. Without the help of these persons, hours of struggling with the project would not have been paid off.

I would like to thank Mr. Nathan Litke, a graduate student at Caltech's Computer Science department, for his generosity of providing us with the spherical meshes and Dr. Christophe Geuzaine, a postdoctoral scholar at Caltech's Applied & Computational Mathematics department, for his `gl2ps` routine.

I would like to thank Mr. and Mrs. Krown for their generous sponsorship for the SURF.

I would like to thank the SURF office for its excellent job of organizing and making SURF possible.

References

R. Abraham, J. E. Marsden, and T. Ratiu. *Manifolds, tensor analysis, and applications*, volume 75 of *Applied Mathematical Sciences*. Springer–Verlag, New York, second edition, 1988.

Rao V. Garimella and Blair K. Swartz. Curvature estimation for unstructured triangulations of surfaces. 2003.

Klaus Hildebrandt and Max Wardetzky. Personal communication. 2004.

Anil N. Hirani. *Discrete Exterior Calculus*. PhD thesis, California Institute of Technology, May 2003. URL http://resolver.caltech.edu/CaltechETD:etd-05202003-095403.

Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *International Workshop on Visualization and Mathematics, VisMath*, 2002.

Ulrich Pinkall and Konrad Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2(1):15–36, 1993.

Guoliang Xu. Convergent discrete laplace-beltrami operators over triangle surfaces. In *GMP*, 2003.

Guoliang Xu. Convergent discrete laplace-beltrami operators over triangle surfaces. In *GMP*, 2004.

A Laplacian Expressions

$$\begin{aligned}
\Delta u_1|_{S^2} = & - && \sin(\phi)(2\sin(\phi)\cos(\cos(\theta)\sin(\phi))\cos(\theta)\cos(\phi)^3\sin(\sin(\theta)\sin(\phi))\sin(\theta) \\
& + && 6\sin(\cos(\theta)\sin(\phi))\sin(\sin(\theta)\sin(\phi))\sin(\theta)\cos(\phi)^3 \\
& - && 6\cos(\cos(\theta)\sin(\phi))\cos(\theta)\cos(\phi)^3\cos(\sin(\theta)\sin(\phi)) \\
& + && \sin(\phi)\sin(\cos(\theta)\sin(\phi))\cos(\sin(\theta)\sin(\phi)\cos(\phi)^3 \\
& - && 2\cos(\cos(\theta)\sin(\phi))\cos(\theta)\cos(\phi)^2\cos(\sin(\theta)\sin(\phi)) \\
& + && 2\sin(\cos(\theta)\sin(\phi))\sin(\sin(\theta)\sin(\phi))\sin(\theta)\cos(\phi)^2 \\
& - && 4\cos(\phi)\cos(\cos(\theta)\sin(\phi))\sin(\theta)\sin(\phi)\sin(\sin(\theta)\sin(\phi))\cos(\theta) \\
& - && 6\sin(\cos(\theta)\sin(\phi))\sin(\sin(\theta)\sin(\phi))\sin(\theta)\cos(\phi) \\
& + && 6\cos(\phi)\sin(\cos(\theta)\sin(\phi))\cos(\sin(\theta)\sin(\phi))\sin(\phi) \\
& + && 6\cos(\cos(\theta)\sin(\phi))\cos(\theta)\cos(\phi)\cos(\sin(\theta)\sin(\phi)) \\
& + && 3\sin(\cos(\theta)\sin(\phi))\cos(\sin(\theta)\sin(\phi))\sin(\phi) \\
& - && 2\sin(\cos(\theta)\sin(\phi))\sin(\sin(\theta)\sin(\phi))\sin(\theta) \\
& - && 2\cos(\cos(\theta)\sin(\phi))\sin(\theta)\sin(\phi)\sin(\sin(\theta)\sin(\phi))\cos(\theta) \\
& + && 2\cos(\cos(\theta)\sin(\phi))\cos(\theta)\cos(\sin(\theta)\sin(\phi)))/(\cos(\phi)+1)
\end{aligned} \tag{26}$$

$$\Delta u_2|_{S^2} = 4\cos(\phi)^2 - 2\sin(\phi)^2 \tag{27}$$

B Tables of Numerical Results

u_i	Error type	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5
u_1	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	4.582 e-0	9.253 e-0	1.926 e+1	3.916 e+1	7.877 e+1	1.579 e+2
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	3.054 e-0	4.091 e-0	5.624 e-0	7.819 e-0	1.094 e+1	1.538 e+1
	$\|\|u - u_d\|\|_{L^\infty}$	9.480 e-18	2.081 e-2	5.456 e-3	1.364 e-3	3.450 e-4	8.638 e-5
	$\|\|u - u_d\|\|_{L^2}$	5.473 e-18	1.120 e-2	2.898 e-3	7.172 e-4	1.789 e-4	4.471 e-5
u_2	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	6.000 e-0	1.200 e+1	2.400 e+1	4.800 e+1	9.600 e+1	1.920 e+2
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	4.899 e-0	6.325 e-0	8.485 e-0	1.116 e+1	1.625 e+1	2.280 e+1
	$\|\|u - u_d\|\|_{L^\infty}$	7.708 e-9	6.424 e-9	6.261 e-9	6.243 e-9	6.241 e-9	3.833 e-8
	$\|\|u - u_d\|\|_{L^2}$	4.450 e-9	3.681 e-9	3.584 e-9	3.571 e-9	3.569 e-9	2.315 e-8
u_3	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	6.419 e-0	1.053 e+1	1.943 e+1	3.797 e+1	7.562 e+1	1.513 e+2
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	2.860 e-0	3.031 e-0	3.793 e-0	5.104 e-0	7.069 e-0	9.905 e-0
	$\|\|u - u_d\|\|_{L^\infty}$	3.774 e-2	1.196 e-2	3.173 e-3	8.275 e-4	2.093 e-4	5.258 e-5
	$\|\|u - u_d\|\|_{L^2}$	2.179 e-2	6.268 e-3	1.709 e-3	4.409 e-4	1.113 e-4	2.789 e-5
u_4	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	3.997 e-0	2.933 e+1	9.742 e+1	7.307 e+1	1.172 e+2	2.289 e+2
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	2.352 e-0	1.295 e+1	2.913 e+1	1.564 e+1	7.624 e-0	8.192 e-0
	$\|\|u - u_d\|\|_{L^\infty}$	3.049 e-18	1.970 e-0	1.770 e-0	2.234 e-1	3.357 e-2	7.878 e-3
	$\|\|u - u_d\|\|_{L^2}$	1.760 e-18	8.688 e-1	4.901 e-1	5.471 e-2	9.635 e-3	2.282 e-3
u_5	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	9.071 e-1	1.698 e-0	5.532 e-0	1.470 e+1	3.143 e+1	6.391 e+1
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	4.591 e-1	1.043 e-0	1.222 e-0	2.001 e-0	2.971 e-0	4.258 e-0
	$\|\|u - u_d\|\|_{L^\infty}$	5.726 e-2	1.071 e-1	3.135 e-2	1.211 e-2	3.018 e-3	7.477 e-4
	$\|\|u - u_d\|\|_{L^2}$	3.306 e-2	4.907 e-2	8.678 e-3	2.465 e-3	6.035 e-4	1.501 e-4
u_6	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	2.392 e-0	5.004 e-0	1.635 e+1	3.787 e+1	7.802 e+1	1.567 e+2
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	1.639 e-0	1.651 e-0	2.549 e-0	3.933 e-0	5.698 e-0	8.090 e-0
	$\|\|u - u_d\|\|_{L^\infty}$	5.979 e-1	1.524 e-1	5.824 e-2	1.335 e-2	3.412 e-3	8.504 e-4
	$\|\|u - u_d\|\|_{L^2}$	3.452 e-1	5.467 e-2	1.274 e-2	2.983 e-3	7.340 e-4	1.828 e-4

Table 1: Errors for planar meshes created from a regular one-ring mesh, i.e. sequence P_{d1} of Figure 2 (a). From the data, one can see that, as the mesh is refined, the L^∞ norm (as well as L^2 norm) error in u eventually decreases with a second order rate with respect to the length of the edges as analytical results confirm. For the errors in Δu, there is no convergence observed. For the moment, we do not know why these errors go up.

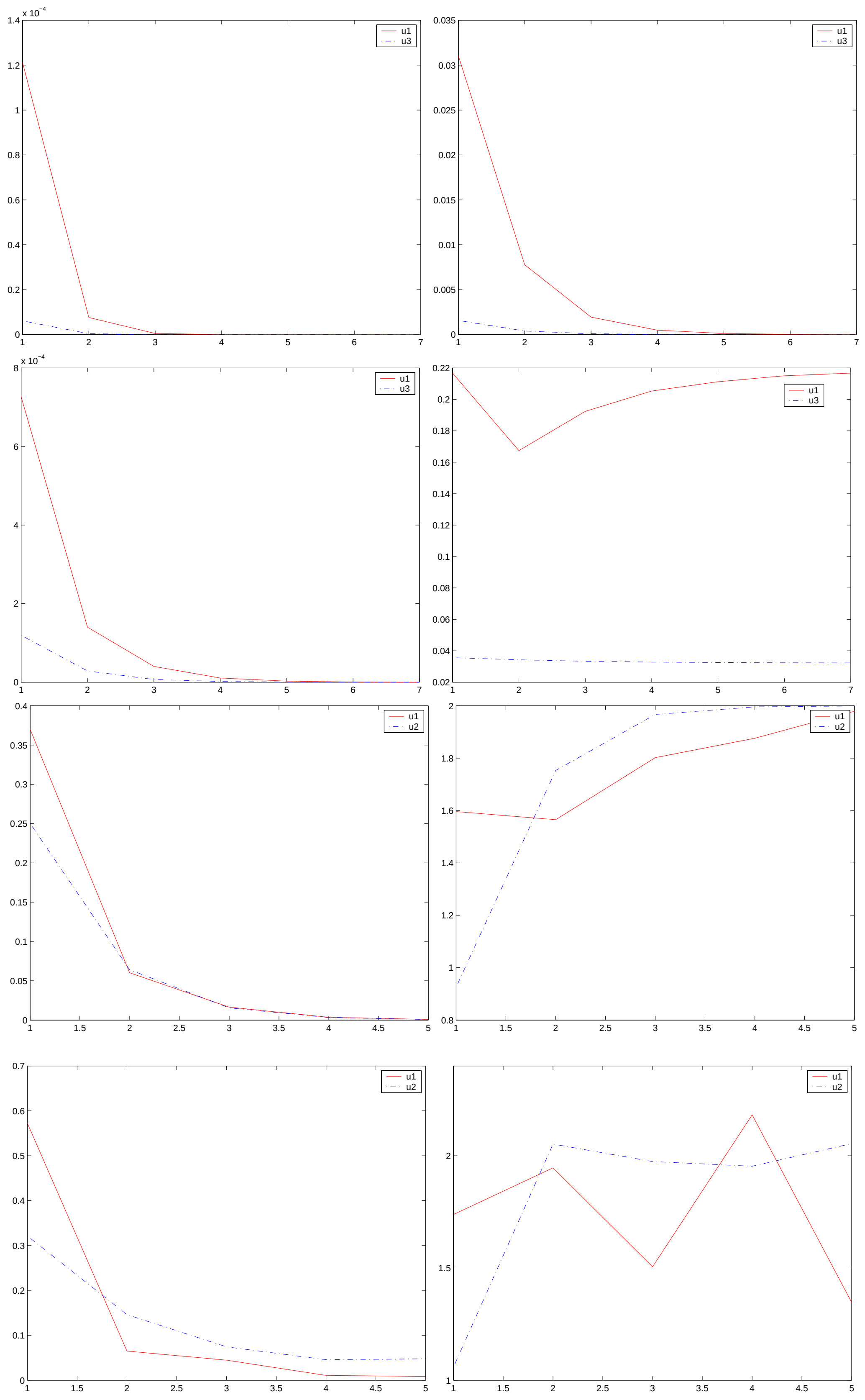

Figure 7: The left column shows L^∞ in u. The right column shows L^∞ in Δu. The data for the first row are taken from Table (4). The data for the second row are taken from Table (7). The data for the third row are taken from Table (9). The data for the last row are taken from Table (12).

u_i	Error type	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5
u_1	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	7.176 e-0	1.553 e+1	3.096 e+1	6.195 e+1	1.239 e+2	2.479 e+2
	$\|\Delta u - \Delta_d u_d\|_{L^2}$	3.624 e-0	4.515 e-0	6.178 e-0	8.606 e-0	1.207 e+1	1.698 e+1
	$\|u - u_d\|_{L^\infty}$	3.978 e-1	8.831 e-2	2.239 e-2	5.887 e-3	1.557 e-3	4.113 e-4
	$\|u - u_d\|_{L^2}$	2.230 e-1	3.156 e-2	7.008 e-3	1.688 e-3	4.178 e-4	1.042 e-4
u_2	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	9.217 e-0	1.840 e+1	3.676 e+1	7.348 e+1	1.469 e+2	2.938 e+2
	$\|\Delta u - \Delta_d u_d\|_{L^2}$	5.034 e-0	6.617 e-0	83950 e-0	1.235 e+1	1.723 e+1	2.420 e+1
	$\|u - u_d\|_{L^\infty}$	4.656 e-2	1.585 e-2	5.158 e-3	1.600 e-3	4.782 e-4	1.391 e-4
	$\|u - u_d\|_{L^2}$	2.688 e-2	5.218 e-3	1.152 e-3	2.742 e-4	6.738 e-5	1.675 e-5
u_3	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	5.592 e-0	8.621 e-0	1.520 e+1	2.913 e+1	5.837 e+1	1.171 e+2
	$\|\Delta u - \Delta_d u_d\|_{L^2}$	2.834 e-0	2.897 e-0	3.554 e-0	4.741 e-0	6.546 e-0	9.162 e-0
	$\|u - u_d\|_{L^\infty}$	1.922 e-2	1.575 e-2	4.923 e-3	1.344 e-3	3.484 e-4	8.766 e-5
	$\|u - u_d\|_{L^2}$	1.110 e-2	7.200 e-3	2.334 e-3	6.318 e-4	1.613 e-4	4.055 e-5
u_4	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	2.175 e-0	4.703 e+1	9.057 e+1	9.196 e+1	1.500 e+2	2.962 e+2
	$\|\Delta u - \Delta_d u_d\|_{L^2}$	2.412 e-0	1.653 e+1	2.805 e+1	1.658 e+1	8.761 e-0	8.514 e-0
	$\|u - u_d\|_{L^\infty}$	3.234 e-1	4.472 e-0	2.687 e-0	7.583 e-1	5.127 e-2	1.201 e-2
	$\|u - u_d\|_{L^2}$	1.867 e-1	1.667 e-0	7.269 e-1	1.616 e-1	1.257 e-2	2.891 e-3
u_5	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	8.559 e-0	5.304 e-0	8.293 e-0	1.982 e+1	4.126 e+1	8.331 e+1
	$\|\Delta u - \Delta_d u_d\|_{L^2}$	4.955 e-0	1.928 e-0	1.519 e-0	2.343 e-0	3.428 e-0	4.896 e-0
	$\|u - u_d\|_{L^\infty}$	1.897 e-0	3.214 e-1	3.382 e-2	8.281 e-3	2.806 e-3	7.084 e-4
	$\|u - u_d\|_{L^2}$	1.095 e-0	1.002 e-1	1.102 e-2	2.264 e-3	5.634 e-4	1.403 e-4
u_6	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	2.189 e-0	8.548 e-0	2.662 e+1	5.870 e+1	1.257 e+2	2.566 e+2
	$\|\Delta u - \Delta_d u_d\|_{L^2}$	1.515 e-0	1.974 e-0	3.352 e-0	4.942 e-0	7.064 e-0	1.000 e+1
	$\|u - u_d\|_{L^\infty}$	4.851 e-1	1.265 e-1	5.004 e-2	1.196 e-2	3.133 e-3	7.810 e-4
	$\|u - u_d\|_{L^2}$	2.801 e-1	4.103 e-2	1.225 e-2	2.795 e-3	6.817 e-4	1.694 e-4

Table 2: Errors for planar meshes created from an irregular one-ring mesh, i.e. sequence P_{d2} of Figure 2 (b). The irregular one-ring mesh is formed by shifting the central vertex away towards the midpoint of an edge. The L^∞ (as well as L^2) error in u is eventually decreasing with a second order rate with respect to the length of the edges of the mesh which is better than analytical results. For the errors in Δu, there is no convergence observed. For the moment, we do not know why these errors go up.

u_i	Error type	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5
u_1	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	4.978 e-0	9.968 e-0	2.122 e+1	4.340 e+1	8.759 e+1	1.748 e+2
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	3.082 e-0	4.095 e-0	5.789 e-0	7.837 e-0	1.094 e+1	1.541 e+1
	$\|\|u - u_d\|\|_{L^\infty}$	1.016 e-1	2.467 e-2	5.560 e-3	1.782 e-3	4.591 e-4	1.198 e-4
	$\|\|u - u_d\|\|_{L^2}$	5.865 e-2	1.098 e-2	2.616 e-3	7.698 e-4	1.894 e-4	4.751 e-5
u_2	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	6.688 e-0	1.317 e+1	2.663 e+1	5.318 e+1	1.068 e+2	2.125 e+2
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	4.906 e-0	6.390 e-0	8.685 e-0	1.165 e+1	1.627 e+1	2.280 e+1
	$\|\|u - u_d\|\|_{L^\infty}$	3.311 e-3	3.931 e-3	1.100 e-3	3.256 e-4	8.383 e-5	2.372 e-5
	$\|\|u - u_d\|\|_{L^2}$	1.912 e-3	1.416 e-3	3.688 e-4	9.945 e-5	2.889 e-5	7.122 e-6
u_3	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	6.576 e-0	1.116 e+1	2.097 e+1	3.828 e+1	7.228 e+1	1.586 e+2
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	2.865 e-0	3.094 e-0	3.854 e-0	5.141 e-0	7.102 e-0	9.909 e-0
	$\|\|u - u_d\|\|_{L^\infty}$	3.515 e-2	8.877 e-3	3.373 e-3	8.581 e-4	2.465 e-4	5.939 e-5
	$\|\|u - u_d\|\|_{L^2}$	2.029 e-2	5.134 e-3	1.643 e-3	4.315 e-4	1.094 e-4	2.789 e-5
u_4	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	7.481 e+1	1.174 e+2	8.945 e+1	6.712 e+1	1.213 e+2	2.384 e+2
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	4.326 e+1	4.005 e+1	2.656 e+1	1.626 e+1	7.852 e-0	8.301 e-0
	$\|\|u - u_d\|\|_{L^\infty}$	1.858 e+1	1.037 e+1	1.969 e-0	6.924 e-1	5.590 e-2	1.041 e-2
	$\|\|u - u_d\|\|_{L^2}$	1.072 e+1	3.891 e-0	5.861 e-1	1.744 e-1	1.161 e-2	2.421 e-3
u_5	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	1.047 e-0	3.407 e-0	6.361 e-0	1.399 e+1	3.505 e+1	7.019 e+1
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	5.221 e-1	1.298 e-0	1.315 e-0	1.982 e-0	3.003 e-0	4.255 e-0
	$\|\|u - u_d\|\|_{L^\infty}$	5.285 e-3	2.328 e-1	5.648 e-2	1.221 e-2	2.906 e-3	8.231 e-4
	$\|\|u - u_d\|\|_{L^2}$	3.051 e-3	7.408 e-2	1.494 e-2	2.455 e-3	5.936 e-4	1.564 e-4
u_6	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	2.176 e-0	4.774 e-0	1.893 e+1	3.724 e+1	8.710 e+1	1.737 e+2
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	1.533 e-0	1.444 e-0	2.627 e-0	3.922 e-0	5.689 e-0	8.126 e-0
	$\|\|u - u_d\|\|_{L^\infty}$	5.404 e-1	1.436 e-1	6.101 e-2	1.315 e-2	3.811 e-3	9.035 e-4
	$\|\|u - u_d\|\|_{L^2}$	3.120 e-1	5.046 e-2	1.440 e-2	3.040 e-3	7.677 e-4	1.891 e-4

Table 3: Errors for planar meshes created from a regular one-ring mesh, i.e. sequence P_{d3} of Figure 2 (c). Each vertex of each mesh is randomly perturbed with a noise of a magnitude of 10% of the average edge of one ring about the vertex. The L^∞ (as well as L^2) error in u is eventually decreasing with a second order rate with respect to the length of the edges of the mesh which is even better than analytical results (first order rate). For the errors in Δu, there is no convergence observed. For the moment, we do not know why these errors go up.

u_i	Error type	n = 8	n = 16	n = 32	n = 64	n = 128	n = 256	n = 512
u_1	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	3.105 e-2	7.773 e-3	1.944 e-3	4.860 e-4	1.215 e-4	3.035 e-5	7.764 e-6
	$\|u - u_d\|_{L^\infty}$	1.213 e-4	7.591 e-6	4.746 e-7	2.966 e-8	1.854 e-9	1.158 e-10	7.213 e-12
u_2	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	3.083 e-8	3.084 e-8	3.084 e-8	3.084 e-8	3.084 e-8	3.086 e-8	3.099 e-8
	$\|u - u_d\|_{L^\infty}$	1.204 e-10	3.011 e-11	7.529 e-12	1.882 e-12	4.714 e-13	1.181 e-13	3.020 e-14
u_3	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	1.567 e-3	3.905 e-4	9.757 e-5	2.440 e-5	6.105 e-6	1.533 e-6	3.896 e-7
	$\|u - u_d\|_{L^\infty}$	6.116 e-6	3.813 e-7	2.382 e-8	1.489 e-9	9.316 e-11	5.847 e-12	3.717 e-13
u_4	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	5.694 e+1	2.182 e+1	6.222 e-0	1.610 e-0	4.060 e-1	1.017 e-1	2.544 e-2
	$\|u - u_d\|_{L^\infty}$	2.224 e-1	2.131 e-2	1.519 e-3	9.826 e-5	6.195 e-6	3.880 e-7	2.426 e-8
u_5	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	8.841 e-1	2.367 e-1	4.024 e-2	1.513 e-2	3.786 e-3	9.468 e-4	2.367 e-4
	$\|u - u_d\|_{L^\infty}$	3.453 e-3	2.311 e-4	1.471 e-5	9.234 e-7	5.778 e-8	3.612 e-9	2.257 e-10
u_6	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	7.802 e-1	1.989 e-1	4.997 e-2	1.251 e-2	3.128 e-3	7.819 e-4	1.954 e-4
	$\|u - u_d\|_{L^\infty}$	3.047 e-3	1.942 e-4	1.220 e-5	7.634 e-7	4.773 e-8	2.983 e-9	1.863 e-10

Table 4: Errors for planar meshes created from a regular one-ring mesh. The regular one-ring mesh is then shrunk (the edge's length is $\frac{1}{n}$), and shifted over a specified region. For these data, one can see that L^∞ norm error in Δu is second order (which agrees with Xu [2003]), while that in u is fourth order. This fourth order is much better than the analytical proof (which is first order). However, for the second function u_2, one can see that L^∞ norm error in Δu seems to be influenced by some other source of error as it does not agree with analytical proof. For the moment, we do not have an interpretation for this phenomenon. And L^∞ norm error in u is second order.

u_i	Error type	n = 8	n = 16	n = 32	n = 64	n = 128	n = 256	n = 512
u_1	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	3.858 e-2	4.129 e-2	4.265 e-2	4.306 e-2	4.308 e-2	4.303 e-2	4.300 e-2
	$\|u - u_d\|_{L^\infty}$	1.503 e-4	4.022 e-5	1.039 e-5	2.622 e-6	6.556 e-7	1.637 e-7	4.090 e-8
u_2	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	1.007 e-2	1.007 e-2	1.007 e-2	1.008 e-2	1.008 e-2	1.008 e-2	1.009 e-2
	$\|u - u_d\|_{L^\infty}$	3.925 e-5	9.813 e-6	2.453 e-6	6.134 e-7	1.534 e-7	3.837 e-8	9.601 e-9
u_3	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	4.666 e-3	3.473 e-3	3.202 e-3	3.137 e-3	3.120 e-3	3.116 e-3	3.116 e-3
	$\|u - u_d\|_{L^\infty}$	1.818 e-5	3.383 e-6	7.798 e-7	1.910 e-7	4.749 e-8	1.186 e-8	2.964 e-9
u_4	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	5.871 e+1	2.308 e+1	6.904 e-0	2.025 e-0	7.080 e-1	3.528 e-1	2.525 e-1
	$\|u - u_d\|_{L^\infty}$	2.288 e-1	2.248 e-2	1.681 e-3	1.233 e-4	1.078 e-5	1.342 e-6	2.402 e-7
u_5	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	8.293 e-1	1.798 e-1	8.547 e-2	7.475 e-2	7.192 e-2	7.103 e-2	7.072 e-2
	$\|u - u_d\|_{L^\infty}$	3.231 e-3	1.752 e-4	1.081 e-5	4.551 e-6	1.095 e-6	2.703 e-7	6.728 e-8
u_6	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	7.312 e-1	1.697 e-1	1.315 e-1	1.206 e-1	1.171 e-1	1.156 e-1	1.149 e-1
	$\|u - u_d\|_{L^\infty}$	2.849 e-3	1.653 e-4	3.201 e-5	7.342 e-6	1.782 e-6	4.398 e-7	1.093 e-7

Table 5: Errors for planar meshes created from a slightly perturbed one-ring mesh. The slightly perturbed one-ring mesh is then shrunk (the edge's length is $\frac{1}{n}$), and shifted over a specified region. For these data, one can see that L^∞ norm error in Δu is eventually zeroeth order (which agrees with Xu [2003]), while that in u is second order. This second order is much better than the analytical proof (which is first order).

u_i	Error type	n = 8	n = 16	n = 32	n = 64	n = 128	n = 256	n = 512
u_1	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	3.438 e-1	3.235 e-1	3.185 e-1	3.172 e-1	3.169 e-1	3.168 e-1	3.168 e-1
	$\|u - u_d\|_{L^\infty}$	1.453 e-3	3.420 e-4	8.415 e-5	2.095 e-5	5.233 e-6	1.308 e-6	3.270 e-7
u_2	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	3.043 e-1	3.043 e-1	3.043 e-1	3.043 e-1	3.043 e-1	3.043 e-1	3.043 e-1
	$\|u - u_d\|_{L^\infty}$	1.286 e-3	3.216 e-4	8.040 e-5	2.010 e-5	5.025 e-6	1.256 e-6	3.141 e-7
u_3	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	8.331 e-2	8.439 e-2	8.466 e-2	8.473 e-2	8.475 e-2	8.475 e-2	8.475 e-2
	$\|u - u_d\|_{L^\infty}$	3.522 e-4	8.920 e-5	2.237 e-5	5.597 e-6	1.400 e-6	3.499 e-7	8.748 e-8
u_4	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	6.117 e+1	2.873 e+1	1.430 e+1	1.003 e+1	8.922 e-0	8.641 e-0	8.603 e-0
	$\|u - u_d\|_{L^\infty}$	2.586 e-1	3.037 e-2	3.778 e-3	6.629 e-4	1.473 e-4	3.568 e-5	8.880 e-6
u_5	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	1.672 e-0	1.072 e-0	9.087 e-1	8.670 e-1	8.565 e-1	8.539 e-1	8.532 e-1
	$\|u - u_d\|_{L^\infty}$	7.070 e-3	1.133 e-3	2.401 e-4	5.727 e-5	1.415 e-5	3.526 e-6	8.807 e-7
u_6	$\|\Delta u - \Delta_d u_d\|_{L^\infty}$	2.261 e-0	1.724 e-0	1.586 e-0	1.552 e-0	1.543 e-0	1.541 e-0	1.540 e-0
	$\|u - u_d\|_{L^\infty}$	9.560 e-3	1.822 e-3	4.192 e-4	1.025 e-4	2.549 e-5	6.363 e-6	1.590 e-6

Table 6: Errors for planar meshes created from a seven-neighbor one-ring mesh. The seven-neighbor one-ring mesh is then shrunk (the edge's length is $\frac{1}{n}$), and shifted over a specified region. For these data, one can see that L^∞ norm error in Δu is eventually zeroeth order (which agrees with Xu [2003]), while that in u is second order. This second order is much better than the analytical proof (which is first order).

u_i	Error type	n = 8	n = 16	n = 32	n = 64	n = 128	n = 256	n = 512
u_1	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	2.167 e-1	1.674 e-1	1.924 e-1	2.053 e-1	2.112 e-1	2.150 e-1	2.167 e-1
	$\|\|u - u_d\|\|_{L^\infty}$	7.267 e-4	1.403 e-4	4.033 e-5	1.075 e-5	2.774 e-6	7.042 e-7	1.774 e-7
u_2	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	1.023 e-1	1.023 e-1	1.023 e-1	1.023 e-1	1.023 e-1	1.023 e-1	1.024 e-1
	$\|\|u - u_d\|\|_{L^\infty}$	3.430 e-4	8.576 e-5	2.144 e-5	5.360 e-6	1.340 e-6	3.351 e-7	8.380 e-8
u_3	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	3.559 e-2	3.428 e-2	3.334 e-2	3.280 e-2	3.251 e-2	3.236 e-2	3.229 e-2
	$\|\|u - u_d\|\|_{L^\infty}$	1.193 e-4	2.874 e-5	6.988 e-6	1.719 e-6	4.259 e-7	1.060 e-7	2.644 e-8
u_4	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	7.282 e+1	3.932 e+1	2.037 e+1	1.361 e+1	1.134 e+1	1.051 e+1	1.018 e+1
	$\|\|u - u_d\|\|_{L^\infty}$	2.442 e-1	3.297 e-2	4.268 e-3	7.129 e-4	1.485 e-4	3.443 e-5	8.332 e-6
u_5	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	1.423 e-0	7.262 e-1	4.867 e-1	3.879 e-1	3.404 e-1	3.182 e-1	3.219 e-1
	$\|\|u - u_d\|\|_{L^\infty}$	4.772 e-3	6.088 e-4	1.020 e-4	2.033 e-5	4.459 e-6	1.042 e-6	2.635 e-7
u_6	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	1.879 e-0	1.085 e-0	7.841 e-1	6.608 e-1	6.077 e-1	5.832 e-1	5.712 e-1
	$\|\|u - u_d\|\|_{L^\infty}$	6.303 e-3	9.098 e-4	1.643 e-4	3.462 e-5	7.961 e-6	1.910 e-6	4.676 e-7

Table 7: Errors for planar meshes created from an aggressively perturbed one-ring mesh. The aggressively perturbed one-ring mesh is then shrunk (the edge's length is $\frac{1}{n}$) and shifted over a specified region. For these data, one can see that L^∞ norm error in Δu is eventually zeroeth order (which agrees with Xu [2003]), while that in u is second order. This second order is much better than the analytical proof (which is first order).

u_i	Error type	Level 0	Level 1	Level 2	Level 3	Level 4
u_1	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	1.186 e-0	1.554 e-0	1.693 e-0	2.196 e-0	3.327 e-0
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	6.693 e-1	5.696 e-1	1.686 e-1	4.599 e-2	1.258 e-2
	$\|\|u - u_d\|\|_{L^\infty}$	5.529 e-1	2.520 e-1	4.821 e-2	1.189 e-2	3.069 e-3
	$\|\|u - u_d\|\|_{L^2}$	2.613 e-1	8.441 e-2	1.356 e-2	3.113 e-3	7.817 e-4
u_2	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	1.792 e-0	7.312 e-0	7.951 e-0	8.001 e-0	8.020 e-0
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	6.216 e-1	5.480 e-1	1.600 e-1	4.435 e-2	1.383 e-2
	$\|\|u - u_d\|\|_{L^\infty}$	3.095 e-1	2.297 e-1	4.000 e-2	9.827 e-3	2.905 e-3
	$\|\|u - u_d\|\|_{L^2}$	2.082 e-1	1.082 e-1	2.349 e-2	5.970 e-3	1.399 e-3

Table 8: Errors for spherical meshes created from an tetrahedral mesh, i.e sequence $S^2_{d,T}$ of Figure 5. For u_1, each mesh has been rotated an angle of 0.1 degree about the y-axis to avoid having a vertex at the poles. The data tell us that only L^∞ in Δu does not converge. One note we would like to make here is that in fact, these meshes are not regular. Some of its one-rings are not congruent. However, because the number of such one-rings is very small, these one-rings do not influence much the behavior of the whole mesh. Therefore, we observe convergence in L^2 norm error.

u_i	Error type	Level 0	Level 1	Level 2	Level 3	Level 4
u_1	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	1.596 e-0	1.565 e-0	1.802 e-0	1.876 e-0	1.979 e-0
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	6.099 e-1	2.887 e-1	9.475 e-2	3.035 e-2	1.091 e-2
	$\|\|u - u_d\|\|_{L^\infty}$	3.702 e-1	6.014 e-2	1.638 e-2	3.529 e-3	8.199 e-4
	$\|\|u - u_d\|\|_{L^2}$	1.367 e-1	2.714 e-2	7.214 e-3	1.577 e-3	3.854 e-4
u_2	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	9.253 e-1	1.753 e-0	1.967 e-0	1.996 e-0	1.999 e-0
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	5.976 e-1	2.949 e-1	1.000 e-1	3.502 e-2	1.328 e-2
	$\|\|u - u_d\|\|_{L^\infty}$	2.505 e-1	6.401 e-2	1.549 e-2	3.179 e-3	6.962 e-4
	$\|\|u - u_d\|\|_{L^2}$	1.339 e-1	3.261 e-2	7.801 e-3	1.695 e-3	4.191 e-4

Table 9: Errors for spherical meshes created from an octahedral mesh, i.e sequence $S^2_{d,O}$ of Figure 5. For u_1 each mesh has been rotated an angle of 0.1 degree about the y-axis to avoid having a vertex at the poles. The data tell us that only L^∞ in Δu does not converge. One note we would like to make here is that in fact, these meshes are not regular. Some of its one-rings are not congruent. However, because the number of such one-rings is very small, these one-rings do not influence much the behavior of the whole mesh. Therefore, we observe convergence in L^2 norm error.

u_i	Error type	Level 0	Level 1	Level 2	Level 3	Level 4
u_1	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	6.064 e-1	4.431 e-1	6.888 e-1	7.398 e-1	7.520 e-1
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	2.918 e-1	1.269 e-1	4.498 e-2	1.788 e-2	7.892 e-3
	$\|\|u - u_d\|\|_{L^\infty}$	7.683 e-2	2.812 e-2	6.943 e-3	1.780 e-3	not available
	$\|\|u - u_d\|\|_{L^2}$	3.221 e-2	8.918 e-3	2.195 e-3	5.527 e-4	not available
u_2	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	4.695 e-1	2.784 e-1	3.278 e-1	3.366 e-1	3.364 e-1
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	1.941 e-1	7.765 e-2	3.035 e-2	1.275 e-2	5.893 e-3
	$\|\|u - u_d\|\|_{L^\infty}$	9.730 e-2	2.330 e-2	6.092 e-3	1.525 e-3	not available
	$\|\|u - u_d\|\|_{L^2}$	3.949 e-2	9.859 e-3	2.688 e-3	6.751 e-4	not available

Table 10: Errors for spherical meshes created from an icosahedral mesh, i.e sequence $S^2_{d,I}$ of Figure 5. For u_1, each mesh has been rotated an angle of 0.1 degree about the y-axis to avoid having a vertex at the poles. The data tell us that only L^∞ in Δu does not converge. One note we would like to make here is that in fact, these meshes are not regular. Some of its one-rings are not congruent. However, because the number of such one-rings is very small, these one-rings do not influence much the behavior of the whole mesh. Therefore, we observe convergence in L^2 norm error.

u_i	Error type	Level 0	Level 1	Level 2	Level 3	Level 4
u_1	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	1.891 e-0	1.539 e-0	4.686 e-0	2.265 e+1	4.842 e+1
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	9.330 e-1	5.856 e-1	2.333 e-1	1.201 e-1	1.257 e-1
	$\|\|u - u_d\|\|_{L^\infty}$	9.454 e-1	3.086 e-1	8.060 e-2	4.042 e-2	1.228 e-2
	$\|\|u - u_d\|\|_{L^2}$	4.168 e-1	1.547 e-1	4.190 e-2	2.126 e-2	5.446 e-3
u_2	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	2.536 e-0	8.455 e-0	6.978 e-0	1.384 e+1	2.287 e+1
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	7.145 e-1	6.084 e-1	2.122 e-1	9.661 e-2	9.733 e-2
	$\|\|u - u_d\|\|_{L^\infty}$	3.119 e-1	2.117 e-1	7.313 e-2	3.297 e-2	4.430 e-2
	$\|\|u - u_d\|\|_{L^2}$	1.535 e-1	1.048 e-1	3.299 e-2	1.439 e-2	3.424 e-2

Table 11: Errors for spherical meshes created from an tetrahedral mesh, i.e sequence $S^2_{d,T}$ of Figure 6. Each vertex of each mesh has been randomly perturbed with a noise of a magnitude of 10% of the average edge of one ring about the vertex. The seed used for perturbation is 0. For function values computation at the vertices, the perturbed vertices are projected back on the sphere. The data tell us that there is a bizarre trend for L^2 error in Δu. As the mesh gets refined, this error goes down and then goes up. We do not know how to interpret that. This will be investigated in future work.

u_i	Error type	Level 0	Level 1	Level 2	Level 3	Level 4
u_1	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	1.738 e-0	1.946 e-0	1.505 e-0	2.182 e-0	1.346 e-0
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	5.797 e-1	3.274 e-1	1.232 e-1	9.589 e-2	1.535 e-1
	$\|\|u - u_d\|\|_{L^\infty}$	5.724 e-1	6.493 e-2	4.473 e-2	1.063 e-2	8.394 e-3
	$\|\|u - u_d\|\|_{L^2}$	2.023 e-1	2.889 e-2	2.396 e-2	4.398 e-3	3.525 e-3
u_2	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	1.058 e-0	2.052 e-0	1.974 e-0	1.953 e-0	2.054 e-0
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	6.706 e-1	3.492 e-1	1.368 e-1	8.377 e-2	1.269 e-1
	$\|\|u - u_d\|\|_{L^\infty}$	3.213 e-1	1.461 e-1	7.423 e-2	4.563 e-2	4.769 e-2
	$\|\|u - u_d\|\|_{L^2}$	1.526 e-1	6.570 e-2	4.858 e-2	3.343 e-2	3.820 e-2

Table 12: Errors for spherical meshes created from an octahedral mesh, i.e sequence $S^2_{d,O}$ of Figure 6. Each vertex of each mesh has been randomly perturbed with a noise of a magnitude of 10% of the average edge of one ring about the vertex. The seed used for perturbation is 0. For function values computation at the vertices, the perturbed vertices are projected back on the sphere. The data tell us that there is a bizarre trend for L^2 error in Δu. As the mesh gets refined, this error goes down and then goes up. We do not know how to interpret that. This will be investigated in future work.

u_i	Norm	Level 0	Level 1	Level 2	Level 3	Level 4
u_1	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	1.033 e-0	7.053 e-1	6.941 e-1	8.302 e-1	1.440 e-0
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	3.460 e-1	1.669 e-1	9.614 e-2	1.169 e-1	2.232 e-1
	$\|\|u - u_d\|\|_{L^\infty}$	2.001 e-1	8.660 e-2	2.929 e-2	8.351 e-3	not available
	$\|\|u - u_d\|\|_{L^2}$	8.551 e-2	4.475 e-2	1.652 e-2	3.291 e-3	not available
u_2	$\|\|\Delta u - \Delta_d u_d\|\|_{L^\infty}$	7.352 e-1	4.695 e-1	3.685 e-1	5.950 e-1	1.215 e-0
	$\|\|\Delta u - \Delta_d u_d\|\|_{L^2}$	2.395 e-1	1.159 e-1	8.228 e-2	1.105 e-1	2.041 e-1
	$\|\|u - u_d\|\|_{L^\infty}$	1.852 e-1	9.572 e-2	8.099 e-2	4.714 e-2	not available
	$\|\|u - u_d\|\|_{L^2}$	6.898 e-2	6.758 e-2	6.502 e-2	3.710 e-2	not available

Table 13: Errors for spherical meshes created from an icosahedral mesh, i.e sequence $S^2_{d,I}$ of Figure 6. Each vertex of each mesh has been randomly perturbed with a noise of a magnitude of 10% of the average edge of one ring about the vertex. The seed used for perturbation is 0. For function values computation at the vertices, the perturbed vertices are projected back on the sphere. The data tell us that there is a bizarre trend for L^2 error in Δu. As the mesh gets refined, this error goes down and then goes up. We do not know how to interpret that. This will be investigated in future work.