THE FINITE ELEMENT METHOD AND CURVED BOUNDARIES

TEODORO FIELDS COLLIN

Abstract. Assuming a full course in real analysis and some basic functional analysis, this paper examines the theory of the finite element method and its implementation on curved boundaries. A review of the theory of the finite element method on polyhedral domains is presented. Then we reveal the deficiencies of the theory on domains with curved boundaries. Code exhibiting these failures is provided. To remedy these deficiencies, we present and apply the theory of isoparametric elements.

Contents

1. Introduction

Although closed form solutions to partial differential equations (PDEs) do exist in some cases, there are remarkably simple examples or remarkably important examples where no useful closed form solution exists. For example, Poisson's equation is widely used in engineering applications yet has no closed form solution (use of a Green's function is not a closed form solution) except in some specific cases. In the

E-mail address: teocollin@uchicago.edu.
Date: Today.

40s, 50s, and 60s computers enter the picture and mathematicians, scientists, and engineers started to numerically solve these problems. One of the key methods that was created is called the finite element method (FEM). By the 60s and 70s, the general mathematical theory was provided for the FEM and software that implemented the FEM became more readily accessible. However, in the same era, serious issues were discovered when the method was applied to curved boundaries. Various solutions were found to these issue in short order. For a more detailed history, see [11]. The objective of this paper will be to provide the theory behind FEM, demonstrate the theory's shortcomings on curved boundaries via computational and theoretical examination, and then remedy the shortcomings.

As I aim to make the paper intelligible to someone who has only taken a year long course in analysis, the first section will briefly introduce some concepts that allow us to formulate the theory. These concepts essentially orbit around the idea of Sobolev spaces. We provide few definitions and no proofs. Instead, we highlight a few technical details and results that hugely influence the theory.

We now make a note about notations and conventions:

Notation 1.1. Unless otherwise stated, all of our domains are subsets of $\mathbb{R}^n$. In particular, Ω is such a domain. If we mention h, it is an element of $(0, 1]$. The Jacobian and the Jacobian determinant are the same thing. If we state a problem, we assume that it is well posed. We will not state $p = \infty$ cases, but they do exist for almost all results that have a p attached to them. Finally, when we define a space of functions V on top of a set Ω, we mean that the functions in V have domain Ω

2. Background and Setting

The first idea that we need is that in PDE we often weaken our notion of what a solution to a PDE is by weakening our notion of derivative.

Remark 2.1 (Weak Derivatives). In particular, we replace the notion of differentiability with the notion of weak differentiability and then require that our solutions satisfy our PDE with derivatives replaces by weak derivatives. Although we do not define the weak derivative, we note that the weak derivative of a function does not need to have pointwise values defined and is only unique up to sets of measure 0. For example, the weak derivative of $f(x) = |x|$ is f' on $\mathbb{R} \setminus \{0\}$ and can be undefined or take any value at 0.

If $g(x) = -1$ on $(-1, 0)$ and 1 on $(0, 1)$ and c_0 at 0 then we know that g will have a weak derivative on $(0, 1)$ and $(-1, 0)$, but not on $(-1, 1)$. Thus, even if a boundary has zero measure, we cannot easily stitch together sets where a function has a weak derivative and still continue to have a weak derivative on the resulting set.

With this notion in place, we can introduce the next idea, the Sobolev space. When you are looking for solutions to a PDE in the weak sense, these are the spaces where you look. We start by defining the Sobolev semi-norm and norm.

Definition 2.2 (Sobolev Norm)**.** Suppose $f \colon \Omega \to \mathbb{R}$. If for every multi-index, $\alpha = (a_1, \ldots, a_n)$ with $a_i \in \mathbb{N}$ and $|\alpha| := \sum a_i = k$, the weak partial derivative $D^\alpha_w f := \frac{\partial f}{\partial x_1^{a_1} \cdots x_n^{a_n}}$ exists, then we say that the Sobolev semi-norm of order (k, p)

on Ω is

$$|f|_{k,p,\Omega} := \Big(\sum_{|\alpha|=k} \|D_w^\alpha f\|_{p,\Omega}^p\Big)^{\frac{1}{p}} \text{ where } \|\cdot\|_{p,\Omega} \text{ is the } L^p(\Omega) \text{ norm.} \tag{2.3}$$

If the same condition is repeated for all α such that $|\alpha| \leq k$ then we say the Sobolev norm of order (k,p) on Ω is

$$\|f\|_{k,p,\Omega} := \Big(\sum_{j=0}^{k} |f|_{j,p,\Omega}^p\Big)^{\frac{1}{p}}. \tag{2.4}$$

We can now define our solution spaces.

Definition 2.5 (Sobolev Space)**.** The Sobolev space on Ω of order (k,p) is the set of all functions $f\colon \Omega \to \mathbb{R}$ such that $\|f\|_{k,p,\Omega} < \infty$. If $p = 2$, then $W_{k,p,\Omega}$ is a Hilbert space notated $H_{k,\Omega}$.

With these spaces in place, we can now talk about the next idea, the weak formulation of a PDE. An example of a PDE that has a weak formulation is Poisson's equation:

Problem 2.6 (Poisson's)**.** Fix a bounded domain Ω. Fix $f, h \in L^2(\Omega)$ and $g \in \mathbb{R}$. Let $\Gamma \subset \partial\Omega$. We wish to find $u \in H_{2,\Omega}$ such that

$$\begin{aligned} -\Delta u &= f \text{ on } \Omega \\ u &= g \text{ on } \Gamma \subset \partial\Omega \\ \frac{\partial u}{\partial n} &= h \text{ on } \partial\Omega \setminus \Gamma. \end{aligned} \tag{2.7}$$

In order to convert this PDE to its weak formulation, we need to make a definition.

Definition 2.8. A bounded bilinear functional on a Hilbert space H is coercive if there is a $\gamma > 0$ such that for all $u \in H$, $a(u,u) \geq \gamma \|u\|_H^2$.

For our purposes, the weak formulation looks like this problem:

Problem 2.9. Fix a bounded domain Ω and a closed subspace $V \subset H$ on top of it. Fix a bounded linear functional G on V and a bounded bilinear functional a on $V \times V$ that is coercive on V. Then we want to find a $u \in V$ such that for all $w \in V$

$$a(u,w) = G(w) \tag{2.10}$$

We note that this is a simplified presentation; there are other things called a weak formulation of a PDE, but we focus in on this particular type of problem. We also note several things about this problem.

Remarks 2.11. First, typically $H \subseteq H_{r,\Omega}$ for some r. Second, via the Lax-Milgram Theorem, a unique solution exists. Third, for Poisson's problem with $g = h = 0$ and $\Gamma = \partial\Omega$, we have that $G(w) = \int_\Omega fw$, $a(u,w) = \int_\Omega \nabla u \cdot \nabla w$, and

$$V = \{v \in H_{1,\Omega}\colon v\,|_\Gamma = 0\}. \tag{2.12}$$

This last item, the definition of V, is crucial to observe because it shows that the boundary conditions on some PDE occasionally affect the definition of V. This observation plays a crucial role in our theory.

We are almost read to begin the general theory, but we need three more facts to grease the wheels. The first two free us from ever thinking about the definition of weak derivative and the final one shows how we can sometimes recover C^k functions from functions in Sobolev spaces. It is known as Sobolev's Inequality. See [5] (Chapter 5) for more information on these facts.

Proposition 2.13. *(1) If a function is C^k then for all weak derivatives corresponding to multi-indices $|\alpha| \leq k$, the standard partial derivatives agree with the weak ones.*

(2) If Ω is a domain, then $C^\infty(\Omega) \cap W_{k,p,\Omega}$ is dense in $W_{k,p,\Omega}$.

(3) Fix a domain $\Omega \subset \mathbb{R}^n$ with a Lipschitz boundary. Fix $k, m \in \mathbb{N}$ with $m < k$. If $p = 1$ and $k - m \geq n$ or if $1 < p < \infty$ and $k - m > n/p$ then there is some constant C such that for all $u \in W_{k,p,\Omega}$, there is a C^m function in the equivalence class of u and $\|u\|_{m,\infty,\Omega} \leq C \|u\|_{k,p,\Omega}$.

We now move to the general theory.

3. The General Theory

In this section we will develop the general theory of the finite element method. The first thing that we do is introduce the finite element abstraction. The next idea is that we reduce Problem 2.9 to a collection of finite dimensional problems using a finite dimensional space indexed by h. We state the finite dimensional problem.

Problem 3.1. Fix a bounded domain Ω_h and a finite dimensional Hilbert subspace $V_h \subset H$ on top of it. Fix a bounded linear function G_h on V_h and a bounded bilinear functional a_h on $V_h \times V_h$ that is coercive on V_h. Then we want to find a $u_h \in V_h$ such that for all $w \in V_h$

$$a_h(u_h, w) = G_h(w). \tag{3.2}$$

Since this problem is finite dimensional, we can use linear algebra to compute a solution to the problem and show that it is well posed. This is exactly how the method is used once you determine what V_h is. Solving it and representing V_h is where most of the computer science happens.

The next idea is to estimate the error of $\|u - u_h\|_H$. To do this, we mandate relationships between Problem 2.9 and Problem 3.1 that allow us to reduce the problem to estimating $\inf_{v \in V_h} \|u - v\|_H$. The final stage is to develop a theory of interpolation on top of our finite element abstractions in order to achieve a bound of similar to $\|u - u_h\|_V \leq Ch^r \|u\|_H$.

For the purpose of this section, fix what is required to state Problem 2.9 with the additional requirement that Ω is polyhedral.

3.1. Finite Elements.

In this section, we almost build V_h by introducing finite elements and the machinery around them. We start with finite elements.

Definition 3.3 (The Finite Element)**.** Let

1. $K \subset \mathbb{R}^n$ be a bounded closed set with nonempty interior and piecewise smooth boundary,
2. P be a finite dimensional space of functions on K, and
3. N be a basis of the dual of P.

Then (K, P, N) is called a finite element.

Now given some Ω, we wish to cover it with finite elements in a way that varies with h. Thus, we need a notion of how to do this.

Definitions 3.4 (Families of Subdivisions and their properties)**.** Given a domain Ω, we say that $\{\mathcal{T}^h\}$ is family of subdivisions indexed by h if for every h, any two $A, B \in \mathcal{T}^h$ have non intersecting interior and $\cup_{T\in\mathcal{T}^h} T = \overline{\Omega}$. We say this collection is good if

$$\max\{\operatorname{diam} T\colon T \in \mathcal{T}^h\} \leq h \operatorname{diam}\Omega \tag{3.5}$$

and regular if there is a $\rho > 0$ such that for all h and all $T \in \mathcal{T}^h$, we have that

$$\operatorname{diam} B_T \geq \rho \operatorname{diam} T \tag{3.6}$$

where B_T is the largest ball in T such that T is star-shaped with respect to B_T.

Now, given an Ω divided up via a particular subdivision, we cover it with finite elements by placing an element on top of every T such that each element on T is just a copy of one particular element, called the reference element. Here is how we copy.

Definition 3.7 (Affine Equivalence)**.** We say that two elements (K, P, N) and (K_F, P_F, N_F) are affine equivalent if there exists an affine invertible map $F = Ax+b$ so that

- $F(K) = K_F$,
- $P_F = \{f \circ F^{-1}\colon f \in P\}$, and
- $N_F = \{f \mapsto n(f \circ F)\colon n \in N\}$.

We note that this is an equivalence relation and almost enough to construct V_h. However, we do not actually build V_h but rather build conditions related to the above into our approximation and interpolation theory and then take advantage of the added structure. In connection with the remaining required constraints and the interpolation theory, we make more definitions.

Definition 3.8 (Local Interpolant and differentiation order)**.** Let (K, P, N) be a finite element. If m is the smallest integer so that $N \subset (C^m(T))'$, then m is the differentiation order of (K, P, N). Let $\{\phi_n\}_{n\in N}$ be the basis of P dual to N. Then the interpolant is a projection from $C^m(K)$ to P defined by

$$I_K f = \sum_{n\in N} \phi_n n(f). \tag{3.9}$$

The idea of the interpolant can be extended and another future requirement on our finite element based V_h emerges.

Definition 3.10 (Global Interpolant and Continuity Order)**.** Given a domain Ω and subdivision $\mathcal{T}$ where each $T \in \mathcal{T}$ belongs to a finite element (T, P, N) of differentiation order m. Then the global interpolant is defined on $f \in C^m(\Omega)$ by

$$I_{\mathcal{T}} f \mid_K = I_K f \text{ for all } K \in \mathcal{T}. \tag{3.11}$$

We say that the interpolant has continuity order r if r is the largest integer so that $I_{\mathcal{T}}(C^m(\Omega)) \subset C^r(\Omega)$.

Now we figure out how we turn estimating $\|u - u_h\|$ into a more tractable problem.

3.2. **Cea's lemma.** The first step in examining the error is the following simple lemma and its corollary due to Jean Cea.

Lemma 3.12. *Given problems Problem 2.9 and Problem 3.1 sharing the same Hilbert super space H and a bounded operator $f\colon V_h \to V$, there exists a constant $C_a > 0$ such that*

$$\|u - f(u_h)\|_H \leq C_a \inf_{v \in f(V_h)} \|u - v\|_H + C_a \sup_{w \in f(V_h)\setminus\{0\}} \frac{|a(u - f(u_h), w)|}{\|w\|_H}. \tag{3.13}$$

Proof. For any $v \in f(V_h)$, we have that

$$\begin{aligned}
\|u - f(u_h)\|_H &\leq \|u - v\|_H + \|v - f(u_h)\|_H \\
&\leq \|u - v\|_H + \frac{1}{\gamma_a} \sup_{w \in f(V_h)\setminus\{0\}} \frac{|a(v - f(u_h), w)|}{\|w\|_H} \\
&\leq \|u - v\|_H + \frac{1}{\gamma_a} \sup_{w \in f(V_h)\setminus\{0\}} \frac{|a(v - u, w) + a(u - f(u_h), w)|}{\|w\|} \\
&\leq \|u - v\|_H + \frac{1}{\gamma_a} \sup_{w \in f(V_h)\setminus\{0\}} \frac{|a(v - u, w)|}{\|w\|_H} + \sup_{w \in f(V_h)\setminus\{0\}} \frac{|a(u - f(u_h), w)|}{\|w\|_H} \\
&\leq \|u - v\|_H + \frac{C_a}{\gamma_a} \|u - v\| + \frac{1}{\gamma_a} \sup_{w \in f(V_h)\setminus\{0\}} \frac{|a(u - f(u_h), w)|}{\|w\|_H}
\end{aligned}$$

where the second line follows via Definition 2.8 and the last via the continuity of a. □

We will revisit this later, but for now its corollary is our focus.

Corollary 3.14 (Cea's Lemma)**.** *Given the conditions of Lemma 3.12 with $a_h = a, G_h = G, \Omega_h = \Omega$, and $V_h \subset V \subset H$ for all $h \in (0, 1]$, we have*

$$\|u - u_h\|_H \leq C_a \inf_{v \in V_h} \|u - v\|_H . \tag{3.15}$$

Proof. Since $u_h \in V_h \subset V$, we have for any $w \in V_h \subset V$, $a(u - u_h, w) = a(u, w) - a(u_h, w) = G(w) - G(w) = 0$. The result follows via Lemma 3.12 with $f = Id$. □

This result is essential as it introduces one of the most important constraints on V_h. Via Remarks 2.11, this essentially requires that V_h inherit information about the boundary or other well-posedness conditions of the PDE.

3.3. **Polynomial Interpolation Theory.** We now have turned estimating $\|u - u_h\|_H$ into estimating $\inf_{v \in V_h} \|u - v\|_H$. The goal of this section is to estimate a bound on this quantity, $\left\|u - I^h u\right\|_H$ where $I^h u$ is a global interpolant for some subdivision in a family indexed by h. As we do this, we are going to introduce more constraints on V_h. In doing so, we are going to pull calculations back to the reference element and so we note how Sobolev norms behave under affine maps.

Lemma 3.16. *Let $F(x) = Bx + b$ be an invertible affine map. If K is a bounded closed set and $F(K) = A$ then for all $f \in W_{m,p,A}$, we have*

$$|f \circ F|_{m,p,K} \leq C_{m,n} \|B\|^m \, |\det(B)|^{-1/p} |f|_{m,p,A} \tag{3.17}$$

and

$$|f|_{m,p,A} \leq C_{m,n} \left\|B^{-1}\right\|^m \, |\det(B)|^{\frac{1}{p}} |f \circ F|_{m,p,K}. \tag{3.18}$$

We do not prove this, but we note the affine map is essential as it makes dealing with the chain rule considerably easier. The next result proves our desired result on a reference element.

Lemma 3.19. *Suppose that (K, P, N) is a finite element of differentiation order l. Suppose K is star shaped with respect to some ball, P contains all polynomials of degree $m-1$ on K, $P \subset W_{m,\infty,K}$. Further, suppose that $p \in (1,\infty)$ is such that $m - l - n/p > 0$. Then for $0 \leq i \leq m$ and $v \in W_{m,p,K}$, we have that*

$$|v - I_K v|_{i,p,K} \leq C_{m,n,K,\|I_{\hat{K}}\|} (\operatorname{diam} K)^{m-i} |v|_{m,p,K} \tag{3.20}$$

where $\hat{K}$ is K under the affine map that divides every coordinate by $\frac{1}{\operatorname{diam} K}$.

Proof. We first prove the case where $\operatorname{diam} K = 1$.

Via hypothesis, Proposition 2.13 (3) applies so that v has a C^l Representative. Thus, I_K is well defined as a map from $W_{m,p,K} \to C^l(K)$. Also, via staring at (3.9), and using that K is bounded, we get that $\{\phi_n\}_{n\in N} \subset P \subset W_{m,\infty,K} \subset W_{m,p,K}$ so that $\|I_K\|_{m,p,K} < \infty$.

Next via approximation theory that we have neglected, under the name of the Bramble-Hilbert lemma (see [1]), K being bounded and star shaped with respect to some ball provides a polynomial w of degree $m-1$ on K. Via these polynomials existing in P and I_K being a projection, we get $I_K w = w$. Then we compute:

$$\begin{aligned}
\|v - Iv\|_{m,p,k} &\leq \|v - w\|_{m,p,K} + \|I_K(w-u)\|_{m,p,K} \\
&\leq \|v - w\| + \left\|I_{\hat{K}}\right\|_{m,p,K} \|v - w\|_{m,\infty,K} \text{ (definition of operator norm)} \\
&\leq (1 + \left\|I_{\hat{K}}\right\|_{m,p,K} C) \|u - w\|_{m,p,K} \text{ (Proposition 2.13 (3))} \\
&\leq (1 + \left\|I_{\hat{K}}\right\|_{m,p,K} C')|v|_{m,p,K} \text{ (Bramble-Hilbert)}
\end{aligned}$$

which concludes the result with $C_{m,n,K,\|I_K\|} = (1 + \left\|I_{\hat{K}}\right\| C')$ after noting that $|u|_{i,p,k} \leq \|u\|_{m,p,K}$ and peeking at the statement of Bramble-Hilbert.

To complete the proof for $\operatorname{diam} K \neq 1$, one uses Lemma 3.16 on (3.20) with the map F sending K to $\hat{K}$. The $(\operatorname{diam} K)^{m-i}$ comes out of the differing powers of the operator norms of F that appear due to the application of Lemma 3.16 on both sides of (3.20). □

This lemma summarizes the result that we want on the reference element. We now study how this result changes as we map into other elements via affine maps.

Lemma 3.21. *Fix a reference element (K, P, N) satisfying the conditions of Lemma 3.19. Suppose (K_F, P_F, N_F) is an affine equivalent element via affine map $F(x) = Ax+b$. Then there is a continuous function $\chi(A)$ such that $\|I_{K_F}\|_{m,p,K_F} < C_{(K,P,N)}\chi(A)$*

Proof Sketch. Via the definition of affine equivalence of elements, we have

$$I_A f = \sum_{n\in N} n(f \circ F)(\phi_n \circ F^{-1}). \tag{3.22}$$

To take the norm of this, you can examine the norms of $n(f \circ F)$ and $(\phi_n \circ F^{-1})$. Applying Lemma 3.16 to both components and doing too much algebra allows one to extract $\|f\|_{m,p,K_F}$, a constant dependent on N and P, and a sum involving norms and determinants of A i.e a continuous function in A. □

With this in place, we can now extend Lemma 3.19 to an entire domain.

Proposition 3.23. *Suppose we have Ω a bounded polyhedral domain with a good and regular family of subdivisions $\mathcal{T}^h$. Suppose there is a reference element (K, P, N) satisfying the conditions of Lemma 3.19 so that for each T which is an element of a subdivision, there is an element (T, P_T, N_T) that is affine equivalent to (K, P, N). Then there exists a constant $C > 0$ dependent on $n, m, p, (K, P, N)$ and the family $\mathcal{T}^h$ such that for $0 \leq s \leq m$*

$$(\sum_{T\in\mathcal{T}^h} \left\|v - I^h v\right\|^p_{s,p,T})^{\frac{1}{p}} \leq Ch^{m-s}|v|_{m,p,\Omega} \tag{3.24}$$

for all $v \in W_{m,p,\Omega}$.

Proof. The proof is in two stages. First, we show the set of $A \in GL(\mathbb{R}^n)$ that we use to make our F is compact. This is needed because our constant in Lemma 3.19 varies as we vary A in a manner that can be controlled via a continuous function.

We pick some T in some triangulation with an associated map $F(x) = Ax + b$. Using the regularity conditions of the family of subdivisions (3.6) and the measure preserving properties of affine maps, we get $0 < C_n\rho^n \leq \mu(B_T) \leq \mu(T) = \int_T dx = |\det A| \int_K \leq |\det A|\mu(K)$. From this, we infer that $A \in \{B\colon |\det B| \geq \epsilon > 0\}$. This set is closed. To trap the A in a bounded set, we note that WLOG we can assume K is positioned so that there is a t_0 such that $\{x\colon \sum x_i \leq t_0, x_i \geq 0\} \subset K$. From this, we infer that $b \in T$ and for all $t \leq t_0$, $Ate_i + b \in T$. Thus, $\|Ate_i\| \leq \operatorname{diam}\Omega$ via (3.5). From this, we can conclude that there is a $t_1 > 0$ such that $A_{ij} \leq t_1$ for all i, j. Thus, $A \in \{B\colon |\det B| \geq \epsilon > 0, |A_{ij}| \leq t_1\}$, a compact set.

With this, we can conclude that the set of constants $C_{m,n,K,\left\|I_{\hat{T}}\right\|}$ from Lemma 3.19 over all subdivisions in the family is bounded above by some $C_{m,n,p,K,\mathcal{T}^h} = C$. Via Lemma 3.16, it is not hard to see that the conditions of Lemma 3.19 are preserved under affine maps. Thus,

$$\begin{aligned}
\sum_{T\in\mathcal{T}^h} \left\|v - I^h v\right\|^p_{s,p,T} &\leq \sum_{T\in\mathcal{T}^h} C^p_{m,n,K,\left\|I_{\hat{T}}\right\|} \sum_{i=0}^{s} (\operatorname{diam} T)^{p(m-i)}|v|^p_{m,p,T} (\textit{Lemma } 3.19) \\
&\leq \sum_{G\in\mathcal{T}^h} C^p_{m,n,K,\left\|I_{\hat{T}}\right\|} \sum_{i=0}^{s} (h \operatorname{diam}\Omega)^{p(m-i)}|v|^p_{m,p,T} \ (3.5) \\
&\leq Ch^{p(m-s)} \sum_{G\in\mathcal{T}^h} |v|^p_{m,p,G}
\end{aligned}$$

With this, we conclude the result.

□

With this in place, we have the interpolation estimates that we need to conclude our convergence theory.

3.4. **Conclusion.** We now have all material in place to state and prove our main convergence result.

Theorem 3.25. *Fixed a bounded polyhedral domain Ω. Fix problems Problem 2.9 and Problem 3.1 sharing the same a, G and Hilbert super space $H = H_{q,\Omega}$. Suppose $V_h \subset V$. Suppose that the conditions of Proposition 3.23 are so with a finite element that has continuity order r and parameters $m \leq q, l, p = 2$. Finally, suppose that*

if I^h is the global interpolant then $I^h(V) \subset V_h$. Then for all $0 < h \leq 1$ and $0 \leq s \leq \min(r+1, m)$, we have

$$\|u - u_h\|_{s,2,\Omega} \leq Ch^{m-s} |u|_{m,2,\Omega} \tag{3.26}$$

where C depends on $a, l, m, n, p, (K, P, N)$, and the family $\mathcal{T}^h$.

Proof. The conditions of Lemma 3.12 are met so $\|u - u_h\|_{s,2,\Omega} \leq \inf_{v \in V_h} \|u - v\|_{s,2,\Omega}$. Since $u \in V$, we know that $I^h u \in V_h$ so we get $\|u - u_h\|_{s,2,\Omega} \leq \left\|u - I^h u\right\|_{s,2,\Omega}$.

Since we have continuity order at least $r + 1 \geq s$, we are justified in writing that $\left\|u - I^h u\right\|_{s,2,\Omega} = (\sum_{T \in \mathcal{T}^h} \left\|v - I^h v\right\|_{s,p,T}^p)^{\frac{1}{p}}$. To see why this is needed, consult Remark 2.1 and note that via the definition of Sobolev Norm, we need the weak derivatives to exist. Then, apply Proposition 3.23 to $(\sum_{T \in \mathcal{T}^h} \left\|v - I^h v\right\|_{s,p,T}^p)^{\frac{1}{p}}$. □

We have completed the result. Note that $I^h(V) \subset V_h$ strongly characterized V_h to the extent that in many practical applications $V_h := I^h(V)$. Now, we break things.

4. The Crime of Solving PDEs on Curved Boundaries

Via computation and theory, we will show the failings of our current theory on curved boundaries.

Example 4.1 (The Computational Example)**.** Let's start out with a version of Poisson's problem. Let Ω be the unit ball in 2 space. Let Γ be the empty set (the pure natural condition). Let $f = 4e^{x^2+y^2}\left(x^2 + y^2 + 1\right)$. Let $h = 2e^{x^2+y^2}\left(x^2 + y^2\right)$ with $V = \left\{v \in H_{1,\Omega} \colon \int_\Omega v = 0\right\}$. We decide that we want to solve this manufactured problem. Without an ability to divide up a unit circle into polyhedral subsets, we settle for approximating a unit circle. In Code Snippet 7.1, we do this with the help of FEnICS (`https://fenicsproject.org`). (Figure 1).

Then we solve the problem on each approximation and to see if we are approaching something, we graph the convergence of the L^2 norm (Figure 2). However, if we plot the solution, we discover that something is seriously wrong (Figure 3). And in case you attribute this to the funky mesh, using a standard polyhedral approximation with this software, we get the same result (Figure 4,Code Snippet 7.3). We seem to converge to a solution, but it is certainly not the correct one. One can manually verify that our solution is $u(x, y) = e^{x^2+y^2} + C$ for some constant C, using that the unit normal on the boundary of the unit ball at (x, y) is (x, y). Thus, the solution should appear to be a vertical translate of Figure 5. It is certainly not this.

The main error here is that the problem as computed does not even make sense. Technically, the boundary conditions as calculated on the unit circle on our approximation only apply on the points of the mesh that touch the unit circle, which is measure zero of the entire boundary. Thus, we need to enforce the correct boundary conditions on the problem. This is rather complicated in the software, but doable (Code Snippet 7.2). We do so and it appears to converge (Figure 6). The solution appears to have at least the right form (Figure 7). It has the correct shape and we note that the distance between a point on the boundary and the center is correct $(u(0, 0) - u(1, 0) = e^0 + C - e - C = 1 - e)$ where as this is not the case in the previous attempt. An analysis of the semi-norm $|\cdot|_{1,2,\Omega}$ compared to the numerical integration of the known solutions shows that this is the correct solution (See Code Snippet 7.4 for an example numerical integration command and Code Snippet 7.2

for numerical integration of the approximate solution). Thus, with some trickery, we were able to adopt our method, but we have no reason to think that this trick should always work.

From all of this analysis, it seems that using our theory on curved boundaries at the minimum requires some trickiness. There does not seem to be much hope for the basic idea of solving the problem on a series of polyhedral domains that approximate it unless we make some new theory. However, it turns out that the basic idea is simply bad. To really get at why this fails even if we properly adjust the boundary conditions to the approximation, we include a lucid theoretical example.

Example 4.2 (Theoretical Example or The Polygon Circle Paradox)**.** We again set Ω to be the unit disk. We fix a problem:

Problem 4.3 (Steklov Problem)**.** Suppose $f \in L^2(\Omega)$ then we wish to find $u \in H_{4,\Omega}$ such that

$$\begin{cases} \Delta^2 u = f & \text{in } \Omega \\ u = \Delta u - (1-\sigma)\kappa \frac{\partial u}{\partial n} = 0 & \text{on } \partial\Omega \end{cases} \tag{4.4}$$

where $\sigma \neq 1$ and κ is the curvature of the domain Ω.

If we then reformulate this problem on a sequence of polyhedral approximations so that $P_n \to \Omega$ in some sense and so that $P_n \subset \Omega$, we note that the boundary condition becomes $u = \Delta u = 0$ because the curvature of a polyhedral domain is 0. If we take a sequence of solutions to the problem phrased on P_n, they converge to the problem on Ω with boundary conditions $u = \Delta u = 0$. For more information on this particular result, see the chapter on the polygon circle paradoxes in [6].

This shows that there is something fundamental about polyhedral domains that makes them unsuited for approximating problems on curved domains. They are particularly bad because they may actually appear to converge i.e they fail silently. For more information on this problem and for some history, see [10],[6], and [11].

With all of this evidence, it is clear that we need to adapt our theory to curved boundaries. Before doing so, we take a moment to view this problem in the context of our general theory where no explicit mention of the boundary is made.

Remark 4.5 (Variational Crimes). To find the exact place where our theory fails with regard to the above examples, we first consult Cea's lemma and Remarks 2.11. The second tells us that boundary conditions for PDEs are represented in the spaces V of Problem 2.9 and V_h of Problem 3.1. Therefore, having differing boundary conditions on the PDE on the polyhedral approximation amounts to $V_h \not\subseteq V$, which is the failure of the main condition of Cea's lemma.

We call the failure of $V_h \subset V$ a variational crime. It can happen in a variety of ways and these are covered in [1].

5. Isoparametric Finite Elements

In this section, we develop isoparametric finite elements and execute a repair of our convergence theory. The interpolation theory will be easy to repair, but the final result will involve new difficulties due to its criminality. We will present a worked case in the end, a version of Problem 2.6.

5.1. **Definition and Examples.** The essential idea of the definition of isoparametric elements is that we still intend to use a nice reference element (K, P, N) so that we can use Lemma 3.19, but we create new elements with sufficiently nice bijections rather than affine maps as the use of affine maps restrains the shapes of our elements.

Definition 5.1 (Isoparametric Finite Elements)**.** Given a finite element (K, P, N) of differentiation order l and a C^l injection F on K with non-vanishing Jacobian, we define a isoparametric element of order l as a finite element (K_F, P_F, N_F) where

- $K_F = F(K)$,
- $P_F = \left\{p \circ F^{-1} \colon p \in P\right\}$, and
- $N_F = \{f \mapsto n(f \circ F) \colon n \in N\}$.

Any element that can be made in this manner from another element (K, P, N) is isoparametrically equivalent to (K, P, N).

We comment on obvious issues.

Remark 5.2. Note that via the chain rule, we know that $f \mapsto n(f \circ F)$ for $f \in C^l(F(K))$ is well defined. We note that F^{-1} exists and is of $C^l(F(K))$ via the inverse function theorem.

We now show how one can easily attempt to manufacture examples and then craft an example of our own.

Construction 5.3 (How to Try to Make an Isoparametric Element)**.** Suppose we have a finite element (K, P, N) where K is defined by some collection of vertices $\{a_i\}_{i=1}^m$ (e.g. it is a simplex) and $N = \{f \to f(a_i)\}_{i=1}^m$. Suppose there is some other collection of points $\{c_i\}_{i=1}^m$ and we would like them to be "vertices" of a curved version of K. Then, it turns out that a good option for F is to define each component for $i = 1, \ldots, n$ by

$$F_i = \sum_{j=1}^{m} \phi_j c_{ji}. \tag{5.4}$$

Via the definition of P and N, we get $n_j(F_i) = c_{ji}$ so $c_j = (n_j(F_1), \ldots, n_j(F_n)) = F(a_j)$. The map is clearly C^∞. Verifying that this map is one to one with non-vanishing Jacobian is more complicated as there is no clear way to do so, but people have for certain choices of K proven simple tests such as only checking that the Jacobian does not vanish at any vertices. (See [4],[3]).

This idea can be expanded to more complicated elements (such as differentiation order 1 elements as in [4]), but more generally when the maps in N are some form of evaluation (at a point, of the directional derivative at a point, etc) trying to craft maps such that $n_j(F_i) = c_{ji}$ is the idea.

We now implement the above construction via code, taking a moment to allude to more software.

Examples 5.5. Let's fix $n = 2$ and imagine a right triangle (Figure 8). We make N evaluation at $(1, 0), (0, 1), (0, 0), (\frac{1}{2}, \frac{1}{2}), (0, \frac{1}{2})$, and $(\frac{1}{2}, 0)$ We let P be polynomials in two variables of degree less than or equal to 2. The corresponding basis dual to N of P is $x(2x-1)$, $y(2y-1)$, $(1-x-y)(2(1-x-y)-1)$, $4xy$, $4y(1-x-y)$, and $4x(1-x-y)$. We implement Construction 5.3 via some matrix multiplication in

Code Snippet 7.5. You can play around with the result in Mathematica by changing the c_i, but we include a sample output in Figure 9.

Now, generating these polynomials to carry out Code Snippet 7.5 is rather annoying. Thankfully, there is software (in this case Firedrake (`firedrakeproject.org`) and FIAT (`https://fenics.readthedocs.io/projects/fiat/en/latest/`), the FInite element Automatic Tabulator) that given the names of several types of finite elements, (K, P, N), will generate P and N for you. Thus, it is possible to automate the process carried out in the last paragraph. Towards that aim, we provide Code Snippet 7.6. This code shows how given an integer d corresponding to the degree of the polynomials and a matrix a of points in $\mathbb{R}^2$, one could automatically generate the coordinates of the map F just as in Code Snippet 7.5.

We now revise our old theory, starting with the interpolation theory.

5.2. **The New Interpolation Theory.** We look back at the old theory and see what needs to change. We can still use Lemma 3.19. However, since we have lost affine maps, we cannot use Lemma 3.21. We proved Lemma 3.21 via Lemma 3.16, which also relies on the use of an affine map. Thus, our first goal in repairing the interpolation theory is a new version of Lemma 3.16. We start with a technical lemma.

Lemma 5.6. *Suppose $f\colon U \to V$ and $g\colon V \to Z$ are both maps from $\mathbb{R}^n \to \mathbb{R}^n$ that are both m times continuously differentiable. Further suppose f is a bijection. For any point $a \in U$, the map $h = g \circ f$ satisfies*

$$\|D^m h(a)\| \leq C_m \sum_{l=1}^{m} \left\|D^l g(f(a))\right\| \Big(\sum_{i \in I_{m,l}} \prod_{j=1}^{l} \left\|D^j f(a)\right\|^{i_j} \Big) \tag{5.7}$$

where $\|\cdot\|$ is the operator norm on the appropriate space of multi-linear functions and $I_{m,l}$ is a finite set of multi-indices that we elide.

Proof. Take any vector $x \in \mathbb{R}^n$ and denote $(x)^m = (x, \ldots, x)$. Via section 7 of [2],

$$D^m h(a) \cdot (x)^m = m! \sum_{l=1}^{m} \sum_{j \in J_{m,l}} \frac{1}{l!} D^l g(f(a)) \cdot \times_{k=1}^{l} \frac{1}{j_k!} D^{j_k} f(a) \cdot (x)^{j_k} \tag{5.8}$$

where $J_{m,l}$ is some other finite set of multi-indices that we elide. We take the sup over all vectors $\|x\| \leq 1$, apply the properties of norms, and then magically reorganize terms:

$$\begin{aligned} \sup_{\|x\| \leq 1} \|D^m h(a) \cdot (x)^m\| &\leq m! \sum_{l=1}^{m} \frac{1}{l!} \left\|D^l g(f(a))\right\| \sum_{j \in I} C_j \prod_{k=1}^{l} \left\|D^k f(a)\right\|^{j_k} \\ &\leq C_m \sum_{l=1}^{m} \left\|D^l g(f(a))\right\| \sum_{i \in I_{m,l}} \prod_{k=1}^{l} \left\|D^k f(a)\right\|^{j_k}. \end{aligned}$$

The result follows via noting that $\|D^m h(a)\| = \sup_{\|x\| \leq 1} \|D^m h(a) \cdot (x)^m\|$ when $h \in C^m$ because then $D^m h$ is a symmetric multi-linear operator via [9]. □

From this we prove the desired replacement of Lemma 3.16.

Lemma 5.9. *Assume the conditions of Lemma 5.6 only instead of $g \in C^m(V)$, we have $g \in W_{m,p,V}$ for $1 \leq p < \infty$ and that the Jacobian of f does not vanish. Then,*

$$|h|^p_{m,p,U} \leq C_{m,p}\chi_{m,p}(\|f\|_{m,\infty,U}) \left\|\det J_f^{-1}\right\|^p_\infty \|g\|^p_{m,p,V} \tag{5.10}$$

where $\chi_{m,p}(a)$ is the max of a^p and a^{mp}.

Proof. Via continuity of norms and Proposition 2.13, it is sufficient to prove this for $g \in C^\infty(V)$. We note that $\|D^m h(a)\|$ is larger than the largest of the order m partial derivatives at the point a. Thus, for some constant dependent on m, we have

$$|h|^p_{m,p,U} \leq C_m \int_U |\, \|D^m h(x)\| \,|^p. \tag{5.11}$$

Since $\left\|D^j f(a)\right\|$ is less than the sum of the absolute values of all j order partial derivative at a and since this in turn is less than the number of such derivatives times the largest single such derivative at a, we get that

$$\begin{aligned}
(\sum_{i\in I_{m,l}} \prod_{j=1}^{l} \left\|D^j f(a)\right\|^{i_j}) &\leq (\sum_{i\in I_{m,l}} \prod_{j=1}^{l} C_j^m \, |f|^{i_j}_{j,\infty,U}) \\
&\leq (\sum_{i\in I_{m,l}} \prod_{j=1}^{l} C_j^m \, \|f\|^{i_j}_{m,\infty,U}) \\
&\leq (\sum_{i\in I_{m,l}} C_i \, \|f\|^{l}_{m,\infty,U}) \\
&\leq C_{m,l} \, \|f\|^{l}_{m,\infty,U}
\end{aligned}$$

We also used that for $i \in I_{m,l}$, $\sum i_k = l$. We combine all of this to complete the proof, starting with an application of Lemma 5.6:

$$\begin{aligned}
|h|^p_{m,p,U} &\leq \int_U |C_m \sum_{l=1}^{m} \left\|D^l g(f(a))\right\| (\sum_{i\in I_{m,l}} \prod_{j=1}^{m} \left\|D^j f(a)\right\|^{i_j})|^p \\
&\leq \int_U |C_m \sum_{l=1}^{m} \left\|D^l g(f(a))\right\| C_{m,l} \, \|f\|^{l}_{m,\infty,U} \,|^p \\
&\leq C_{m,p} \max \|f\|^p_{m,\infty,U} \,, \|f\|^{mp}_{m,\infty,U} \int_U |\sum_{l=1}^{m} \left\|D^l g(f(a))\right\| \,|^p \\
&\leq C'_{m,p} \max \left(\|f\|^p_{m,\infty,U} \,, \|f\|^{mp}_{m,\infty,U}\right) \left\|\det J_f^{-1}\right\|_\infty \|g\|^p_{m,p,V} \,.
\end{aligned}$$

We note that we use that $(a+b)^p \leq 2^{p-1}(a^p + b^p)$ for $a, b \geq 0$ and $p \geq 1$. We also reused a property of the operator norm of the derivative at a point and change of variables. □

We can now repair our convergence theory. With this lemma, a new version of Lemma 3.21 follows easily, but we note the continuous function will be dependent on derivatives of F and F^{-1} as well as the Jacobians of these maps. Consequently, the compactness result at the heart of Proposition 3.23 cannot be replicated. Instead, to replicate Proposition 3.23, we simply add in new conditions.

Proposition 5.12. *Assume the conditions of Proposition 3.23 but replace affine equivalence with isoparametric equivalence. Further, require that there are some $C, c > 0$ such that for every map F from the reference element to isoparametric element we have that*

- $\|F\|_{m,\infty,K} < C$,
- $\left\|F^{-1}\right\|_{m,\infty,F(K)} < C$,
- $c < \left\|\det J_f^{-1}\right\|_\infty < C$, *and*
- $c < \|\det J_f\|_\infty < C$.

Then the result of Proposition 3.23 holds.

The proof is the same only the compactness result is replaced with an application of the new bounds to constrain the continuous function introduced in Lemma 5.9. From this, an obvious replacement of Theorem 3.25 occurs where the conditions of Proposition 3.23 are replaced with those of Proposition 5.12,but we can drop the requirement that the domain is polyhedral. With this result in place, we note that it is unsatisfactory.

Remark 5.13. The essential issue with the proposed revisions to Proposition 3.23 is a practical one in two senses. First, we wish to apply our theory to various domains. How do we know that we can subdivide them into sets that are all isoparametric equivalent to one particular set? Second, via Definition 5.1, if we wish to find the polynomial basis on top of each subdivision, we need to know something about F^{-1}. So, assuming the maps exist, how do we find them? These two issues indicate that our current theory of isoparametric elements is insufficient and we need to revise it for it to be useful.

We tweak our problem. Since it is impractical to exactly approximate Ω, we adopt a new scheme. We state a new family of problems on a family of polyhedral approximations to Ω and then lift this to a family of problems on a family of curved approximations of Ω.

Problem 5.14. Fix a natural k. Fix a bounded Lipschitz domain Ω. Suppose Ω_h is a sequence of inner polyhedral approximations to Ω. Suppose we have a family of finite dimensional Hilbert sub-spaces $W_h \subset H$ on each Ω_h. For each h, fix a map $F^h \colon \Omega_h \to F(\Omega_h)$ that satisfies the following properties:

(1) each map F^h is k times weak differentiable;
(2) each map F^h is one to one;
(3) component-wise each map F^h is a piece-wise polynomial map of degree $k-1$;
(4) there are positive C and c so that for all h, $\left\|F^h\right\|_{k,\infty,\Omega_h} < C$, $\left\|(F^h)^{-1}\right\|_{k,\infty,\Omega_h} < C$, and $|\det D^1 F^h(x)| \in [c, C]$;
(5) for each h, $F^h = I$ outside of Ω_h;
(6) the distance from a point on $\partial\Omega$ to the closet point on $\partial F^h(\Omega_h)$ is $O(h^k)$.

Define $V_h := \left\{v \circ (F^h)^{-1} \colon v \in W_h\right\}$ and suppose it is a Hilbert space. Fix for each h a bounded linear functional G_h on V_h and a bounded bilinear coercive functional a_h on V_h. For each h, we wish to find $u_h \in V_h$ such that for all $w \in V_h$

$$G_h(w) = a_h(u_h, w). \tag{5.15}$$

We must point out a number of things about this problem.

Remarks 5.16. First, we mandate a Lipschitz domain so that we may apply a certain Sobolev extension theorem to elements of H so that they are defined on $F^h(\Omega_h)$ as we do not require that $F^h(\Omega_h) \subseteq \Omega$. Second, we note that the properties of the map F^h ensure that V_h is a Sobolev space of the same order as W_h. Third, we note that the final two properties tell us that $F^h(\Omega_h)$ approximates Ω and its boundary well. They do not matter for interpolation on $F^h(\Omega_h)$, but tell us that this is a sane thing to do. They will matter later on. Finally, we note that V_h is called the isoparametric finite element space and by chopping up Ω_h into finite elements, one can view V_h on top of $F^h(\Omega_h)$ as a collection of isoparametrically equivalent elements.

With this in place, we can create the final isoparametric polynomial approximation proposition.

Proposition 5.17. *Fix a problem Problem 5.14 with some k. Suppose for each Ω_h there is a triangulation $\mathcal{T}^h$ so that the induced family over all h is good and regular. Let (K, P, N) be a continuity order 0 and differentiation order l reference element satisfying the conditions of Lemma 3.19 with $m := k$,l and some p. Further, suppose that each triangle in the family has an associated finite element that is affine equivalent to (K, P, N). Then there is a positive constant not dependent on h such that for $0 \leq s \leq 1$ and $v \in W_{m,p,\Omega}$, we have that*

$$\left\|v - I^h v\right\|_{s,p,F^h(\Omega_h)} \leq Ch^{m-s} \left|v\right|_{m,p,F^h(\Omega_h)} \tag{5.18}$$

where $(I^h v)(F^h(x)) := I^{W_h}(v \circ F^h)(x)$ for $x \in \Omega_h$.

Proof Sketch. Via an extension result, we can view v as being defined on $F^h(\Omega_h)$. Via property (4) of the map F^h, change of variables, and the chain rule, this gets us $\left\|v - I^h v\right\|_{s,p,F^h(\Omega_h)} \leq C' \left\|v \circ F^h - I^{W_h}(v \circ F^h)\right\|_{s,p,\Omega_h}$. Here, we can apply with a bit of modification Proposition 5.17 to find that $\left\|v - I^h v\right\|_{s,p,F^h(\Omega_h)} \leq Ch^{m-s} \left|v \circ F^h\right|_{m,p,\Omega_h}$. Apply Lemma 5.9 and use property 4 to get the desired result.

□

With this result in place, we now turn to the replication of Theorem 3.25.

5.3. **Acquittal.** Suppose that we have Problem 2.9 fixed and an associated Problem 5.14. We have no reason to suspect that $V_h \subset V$ even if we required that $W_h \subset V$. Even though F^h may map a boundary to a boundary, since $F^h(\Omega_h)$ might have a boundary outside of Ω, we cannot be sure that the boundary conditions enforced in V or W_h are passed on to V_h. The lessons of the previous section tell us that this is criminal and that we must not use Corollary 3.14. Instead we use Lemma 3.12 and introduce additional assumptions to bound the extra term.

Assumption 5.19. Suppose that we have Problem 2.9 and an associated Problem 5.14 with $k \in \mathbb{N}$. We assume four things:

(1) There is a k diffeomorphism $\Phi_h \colon \Omega \to F^h(\Omega_h)$ whose partials, Jacobian, and inverse Jacobian satisfy the same properties as F^h such that for all h, $v \mapsto v \circ \Phi^h$ is bounded operator $V_h \to V$ and $v \mapsto v \circ (\Phi^h)^{-1}$ is a bounded operator from $V \to V_h$. We denote these bounded operators by $v \to \hat{v}$ and $v \to \check{v}$. We also assume that

$$\left\|D\Phi^h - I\right\|_{\infty,\Omega} = O(h^{k-1}). \tag{5.20}$$

(2) For all $f, g \in V_h$,

$$\left|a_h(f,g) - a(\hat{f},\hat{g})\right| \leq C_a(h^{k-1}) \left\|\hat{f}\right\|_H \|\hat{g}\|_H . \tag{5.21}$$

(3) For all $g \in V_h$,

$$|F(\hat{g}) - G_h(g)| \leq C_G(h^{k-1}) \|\hat{g}\|_H . \tag{5.22}$$

(4) This could be several items. The needed one is that $\|\hat{u}_h\|_H \leq C \|u\|_H$ with C independent of h, but it could be phrased as the convergence or boundedness of the sequence u_h in H.

With this assumption, the reader is probably pulling their hair and wondering

Remark 5.23 (where are these maps coming from?). We have essentially summoned from nowhere maps F^h and Φ_h that are crucial to our solutions to the problem. Where do they come from? The answer is that it is all due to [8]. This is not particularly satisfying, but the details are actually quite nasty and explicating them in any useful way is another paper's job. We should also note that we only have these maps in $n = 2$ or $n = 3$, but these are where most practical applications happen so this is okay.

We now state and prove the final theorem.

Theorem 5.24. *Fix a bounded domain Ω with a Lipschitz boundary. Fix problems Problem 2.9 and Problem 5.14 that together satisfy Assumption 5.19 with a Hilbert super space $H := H_{q,\omega}$ and $k := q + 1$. Suppose that the conditions of Proposition 5.17 are meet with a finite element that has continuity order $r = 0$ and parameters $m \leq q, l, p = 2$. Finally, suppose that if I^h is the global interpolant then $I^h(V) \subset V_h$. Then for all $0 < h \leq 1$ and $0 \leq s \leq \min(r+1, m)$, we have*

$$\|u - \hat{u_h}\|_{s,2,\Omega} \leq Ch^{m-s} \|u\|_{m,2,\Omega} \tag{5.25}$$

where C does not depend on h.

Proof. We apply Lemma 3.12 with $f(v) = v \circ \Phi_h$:

$$\|u - \hat{u_h}\|_{s,2,\Omega} \leq C_a \inf_{v \in V_h} \|u - \hat{v}\|_{s,2,\Omega} + C_a \sup_{w \in V_h \setminus \{0\}} \frac{|a(u - \hat{u_h}, \hat{w})|}{\|\hat{w}\|_{s,2,\Omega}}. \tag{5.26}$$

Bounding the first term here is the same as in Theorem 3.25, but we use Proposition 5.17 and liberally apply our assumptions on Φ^h along with the chain rule and change of variables:

$$\begin{aligned}\inf_{v \in V_h} \|u - \hat{v}\|_{s,2,\Omega} &\leq C \inf_{v \in V_h} \|\check{u} - v\|_{s,2,F^h(\Omega_h)} \\ &\leq C \left\|\check{u} - I^h\check{u}\right\|_{m,s,F^h(\Omega_h)} \leq Ch^{m-s} \|\check{u}\|_{m,s,F^h(\Omega_h)} \leq Ch^{m-s} \|u\|_{m,s,\Omega} .\end{aligned}$$

Here C does change from step to step, but no dependence on h is introduced.

Then we note that via the definition of the problems ((5.15) and (2.10)) and then via our assumptions, we have

$$\begin{aligned}
|a(u-\hat{u_h},\hat{w})| &= |G(\hat{w}) - a(\hat{u_h},\hat{w})| \\
&= |G(\hat{w}) - G_h(w) + a_h(u_h,w) - a(\hat{u_h},\hat{w})| \\
&\leq |G(\hat{w}) - G_h(w)| + |a_h(u_h,w) - a(\hat{u_h},\hat{w})| \\
&\leq C_{a,G}h^{k-1}(1+\|\hat{u_h}\|_{s,2,\Omega})\|\hat{w}\|_{s,2,\Omega} && (2) \text{ and } (3) \\
&\leq C'_{a,G}h^{k-1}(1+\|u\|_{s,2,\Omega})\|\hat{w}\|_{s,2,\Omega} && (4)
\end{aligned}$$

Combine these two results with (5.26) and the result follows. □

This is the final result, but you might not believe Assumption 5.19 is realistic. Thus, we verify it for Problem 2.6.

Proposition 5.27. *Set Problem 2.6 as problem Problem 2.9 with* $\Gamma = \partial\Omega$, $H := H_{1,\Omega}$, *and* $V := \{v \in H : v\,|_{\partial\Omega} = 0\}$. *We have* $G(v) = \int_\Omega fv$ *and* $a(u,v) = \int_\Omega \nabla u \cdot \nabla v$. *Set* $G_h(v) = \int_{F^h(\Omega_h)} fv$ *and* $a_h(u,v) = \int_{F^h(\Omega_h)} \nabla u \cdot \nabla v$ *for Problem 5.14 with the map* F^h *supplied via* [8]. *Let* $W_h = \{v\,|_{\Omega_h} : v \in H, v\,|_{\partial\Omega_h} = 0\}$. *Then we can verify Assumption 5.19.*

Proof. (1) The existence of the map F^h follows via [8] in the case $n = 2$ or $n = 3$. We ignore the other cases. The boundedness of the defined operators follows from the properties of the map and Lemma 5.9. The mapping from $V \to V_h$ is clear by the boundedness of the operator. The mapping from $V_h \to V$ follows because the continuity of Φ_h and continuity of the inverse of Φ_h ensures that boundaries are mapped to boundaries.

(2) Via change of variables and then the chain rule, we have that

$$a_h(f,g) = \int_\Omega \left|\det D\Phi^h\right| \nabla\hat{f}(D\Phi_h)^{-1} \cdot \nabla\hat{g}(D\Phi_h)^{-1}. \tag{5.28}$$

and

$$a(\hat{f},\hat{g}) = \int_\Omega \nabla\hat{f} \cdot \nabla\hat{g}. \tag{5.29}$$

Then write

$$\begin{aligned}
a_h(f,g) - a(\hat{f},\hat{g}) &= \int_\Omega |\det D\Phi_h| \nabla\hat{f}(D\Phi_h)^{-1} \cdot \nabla\hat{g}(D\Phi_h)^{-1} - \int_\Omega \nabla\hat{f} \cdot \nabla\hat{g} \\
&= \int_\Omega |\det D\Phi_h| \nabla\hat{f}((D\Phi_h)^{-1} - I) \cdot \nabla\hat{g}(D\Phi_h)^{-1} \\
&+ \int_\Omega \nabla\hat{f} \cdot \nabla(\hat{g}(D\Phi_h)^{-1} |\det D\Phi_h|) - \int_\Omega \nabla\hat{f} \cdot \nabla\hat{g} \\
&= \int_\Omega |\det D\Phi_h| \nabla\hat{f}((D\Phi_h)^{-1} - I) \cdot \nabla\hat{g}(D\Phi_h)^{-1} \\
&+ \int_\Omega \nabla\hat{f} \cdot \nabla(\hat{g}(\left|\det D\Phi^h\right| (D\Phi^h)^{-1} - I)).
\end{aligned}$$

Using bounds on the uniform bounds above and below $\left|\det D\Phi^h\right|$, (5.20), and continuity of inversion, the result follows.

(3) This is similar. We we write $G(\hat{w}) = (f, \hat{w})$ and $G(w) = (f, w)_h$. We use change of variables, (5.20), the continuity of det, and Holder's inequality:

$$\begin{aligned}
|(f,\hat{w}) - (f,w)_h| &= |\int_\Omega (f(x) - \hat{f}(x))|\det J_{\Phi_h}|(x)\hat{w}(x)| \\
&\leq |\int_\Omega (f(x) - \hat{f}(x))|\det J_{\Phi_h}|(x)\hat{w}(x) + f\hat{w} - f\hat{w}| \\
&\leq |\int_\Omega f\hat{w}(|\det J_{\Phi_h}| - 1)| + |\int_\Omega f\hat{w} - \hat{f}\hat{w}|\det J_{\Phi_h}|| \\
&\leq Ch^{k-1}\|f\|_{1,2,\Omega}\|\hat{w}\|_{1,2,\Omega} + \|f\|_{1,\infty,\Omega}\int_\Omega |(|\det J_{\Phi_h}| - 1)\hat{w} \\
&\leq Ch^{k-1}\|f\|_{1,2,\Omega}\|\hat{w}\|_{1,2,\Omega} + \|f\|_{1,\infty,\Omega} C\int_\Omega h^k\hat{w} \\
&\leq Ch^{k-1}\|f\|_{1,2,\Omega}\|\hat{w}\|_{1,2,\Omega} + C'\|f\|_{1,\infty,\Omega} h^k\|\hat{w}\|_{1,2,\Omega} \\
&\leq C_G h^{k-1}\|\hat{w}\|_{1,2,\Omega}.
\end{aligned}$$

(4) We skip this item because it involves machinery that has not been developed. One would also typically use techniques that rely more closely on a choice of Ω_h and the last two properties of F^h. See [11] and [8] for an example of how this might be done. Another example lies in the recommended passages for [6]

□

6. Acknowledgments

I'm thankful for the support, bemusement, and banter of my mentors: Claudio Gonzales and Eric Stubley. I'm particularly thankful for their willingness to put confidence in me and follow me into a topic outside their defined areas of interest. Many of the results and proofs of this book are elaborations on or combinations of proofs from [1],[3],[4],and [8]. Of these, [1] is the most deserving of praise. I also enjoyed the material in [7],[11], and [10]. I would also like to thank all the faculty involved in the REU. In particular, I would like to thank Professor Peter May for envisioning, creating, organizing, and hosting the REU.

References

[1] Susanne C. Brenner and L. Ridgway Scott. *The Mathematical Theory of Finite Element Methods*. Springer.
[2] Henri Cartan. *Differential Calculus*.
[3] Philippe G. Ciarlet. *The Finite Element Method for Elliptic Problems*.
[4] Philippe G. Ciarlet and Pierre-Arnaud Raviart. *Interpolation Theory over Curved Elements, with Applications to Finite Element Methods*.
[5] Lawrence C Evans. *Partial Differential Equations*.
[6] Fillippo Gazzola, Hans-Chirstoph Grunau, and Guido Sweers. *Polyharmonic Boundary Value Problems: Positivity Preserving and Nonlinear Higher Order Elliptic Equations in Bounded Domains*.
[7] Thomas J. R. Hughes. *The Finite Element Method Linear Static and Dynamic Finite Element Analysis*.
[8] Marc Lenoir. *Optimal Isoparamametric Finite Elements and Error Estimates For Domains involving Curved Boundaries*.
[9] T. Muramatu and S. Wakabayashi. *On the norms of a symmetric multilinear form*.

[10] H. Chongo Rhee and Satya N. Atluri. *Polygon-Circle Paradox in the Finite Element Analysis of Bending of a Simply Supported Plate.*
[11] L. Ridgway Scott. *Finite Element Techniques for Curved Boundaries.*

7. Appendix

7.1. **Code and Commentary.**

Code Snippet 7.1. This snippet presents python and FEniCS code that solves Poisson's equation on series of polyhedral approximations. Since the boundary condition is natural, some work is done to ensure uniqueness via a Lagrange multiplier like scheme.

```
import math
from dolfin import *
import mshr
import numpy as np
import matplotlib.pyplot as plt
parameters["form_compiler"]["cpp_optimize"] = True
parameters["form_compiler"]["optimize"] = True
parameters["ghost_mode"] = "shared_facet"

results = []

#define the mesh points
def meshPoints(n):
    return((map(lambda x: Point(np.array((math.cos(2.0*math.pi*float
        ↪ (x)/n),(math.sin(2.0*math.pi*float(x)/n))))), range(0,n)))
        ↪ )
n = 500
for q in range(4,n):
    mp = (meshPoints(q))
    dom = mshr.Polygon(mp)
    mesh = mshr.generate_mesh(dom,1)
    #generate and plot mesh
    if q % 50== 0 or q < 8:
        plot(mesh,interactive=True)

    # Create mesh and define function space
    P1 = FiniteElement("Lagrange", mesh.ufl_cell(), 1)
    R = FiniteElement("Real", mesh.ufl_cell(), 0)
    V = FunctionSpace(mesh, P1 * R) #use lagrange to make problem
        ↪ unique




    # Define variational problem
    (u, c) = TrialFunction(V)
    (v, d) = TestFunctions(V)
```

```
    f = Expression("0-4*exp(x[0]*x[0]+x[1]*x[1])*(x[0]*x[0]+x[1]*x
        ↪ [1]+1)",element=P1)
    h = Expression("2*exp(0)",element=P1)
    a = (inner(grad(u), grad(v)) + c*v + u*d)*dx
    L = f*v*dx + h*v*ds

    # Compute solution
    if q % 1 == 0:
        w = Function(V)
        solve(a == L, w)
        (u, c) = w.split()
        eq = norm(u,"L2")
        print("On {0} mesh points the norm is: {1}".format(q,eq))
        a = mp[3]
        print(abs(u(0,0)-u(a)))
        results.append(eq)
        if q % 250 == 0:
            file = File("poisson.pvd")
            file << u
            # Plot solution
            plot(u, interactive=True)
            plot(mesh,interactive=True)



    # Save solution in VTK format
    if q == -10:
        file = File("poisson.pvd")
        file << u
        # Plot solution
        plot(u, interactive=True)


x = range(4,n)
plt.plot(x,results)
plt.savefig("pois1l2.png")
```

Code Snippet 7.2. This snippet presents python and FEniCS code that solves Poisson's equation on series of polyhedral approximations. Unlike the previous code however, this code takes into account the correct boundary conditions on the approximation.

```
# Begin demo
import math
from dolfin import *
import mshr
import numpy as np
import matplotlib.pyplot as plt
import math
```

```
parameters["form_compiler"]["cpp_optimize"] = True
parameters["form_compiler"]["optimize"] = True
parameters["ghost_mode"] = "shared_facet"
# Define Dirichlet boundary x < 0
def boundary(x,on_boundary):
    return on_boundary
results = []



def meshPoints(n):
    return((map(lambda x: Point(np.array((math.cos(2.0*math.pi*float
        ↪ (x)/n),(math.sin(2.0*math.pi*float(x)/n))))), range(0,n)))
        ↪ )
n = 500
meh = range(4,n)
for q in meh:
    mp = (meshPoints(q))
    dom = mshr.Polygon(mp)
    mesh = mshr.generate_mesh(dom,1)
    #operation to make sure that we compute the correct boundary
        ↪ conditions
    class MyExpression0(Expression):
        def eval(self, value, x2):
            x1 = list(x2)
            x = x1[0]
            y = x1[1]

            p = Point(np.array((x,y)))
            minlist = np.array(map(lambda x: p.distance(x),mp))
            minids = minlist.argsort()[-2:]
            p1 = mp[minids[0]]
            p2 = mp[minids[1]]
            nx = 0.5*(p1.x()+p2.x())
            ny = 0.5*(p1.y()+p2.y())

            res = 0-2*(math.e**(x**2 + y**2))*(nx*x+ny*y)
            value[0]=res

        def value_shape(self):
            return (1,)

    # Create mesh and define function space
    # mesh2 = UnitSquareMesh(64, 64)
    P1 = FiniteElement("Lagrange", mesh.ufl_cell(), 3)
    R = FiniteElement("Real", mesh.ufl_cell(), 0)
    V = FunctionSpace(mesh, P1 * R)
```

```
# Define variational problem
(u, c) = TrialFunction(V)
(v, d) = TestFunctions(V)
f = Expression("-4*exp(x[0]*x[0]+x[1]*x[1])*(x[0]*x[0]+x[1]*x
    ↪ [1]+1)",element=P1)
#h = Expression("2*exp(0)",element=P1)
h = MyExpression0(element=P1)
a = (inner(grad(u), grad(v)) + c*v + u*d)*dx
L = f*v*dx + h*v*ds

# Compute solution
if q % 1 == 0:
    w = Function(V)
    solve(a == L, w)
    (u, c) = w.split()
    eq = norm(u,"H10") #nuermical integration of approximate
        ↪ solution
    print("On {0} mesh points the norm is: {1}".format(q,eq))
    a = mp[3]
    print("The value at (0,0) is {0}".format(u((0,0))))
    print("The value at a is {0}".format(u(a)))
    print("The error between the two is {0} and should be {1}".
        ↪ format(abs(u(0,0)-u(a)),abs(1-math.e)))
    results.append(eq)
    #print(c(0,0))
    #print(u(0,0))
    #plot(u,interactive=True)
    if q % 250 == 0:
         file = File("poisson.pvd")
         file << u
         # Plot solution
         plot(u, interactive=True)
         plot(mesh,interactive=True)



#actualu = Expression("exp(x[0]*x[1]*x[0]*x[1])-exp(1)",element=
    ↪ V.ufl_element())
#Iu = interpolate(actualu,V)
#eq = norm(u,"H1")
#print("On {0} mesh points the correct norm is: {1}".format(q,eq
    ↪ ))


# Save solution in VTK format
if q == -10:
    file = File("poisson.pvd")
```

```
        file << u
        # Plot solution
        plot(u, interactive=True)


x = range(4,n)
plt.plot(x,results)
plt.savefig("ploth22.png")
```

Code Snippet 7.3. The solution for our Poisson's equation on the recommended circle approximation.

```
import math
from dolfin import *
import mshr
import numpy as np
parameters["form_compiler"]["cpp_optimize"] = True
parameters["form_compiler"]["optimize"] = True
parameters["ghost_mode"] = "shared_facet"

domain = mshr.Circle(Point(0.,0.),1.0,120)
mesh = mshr.generate_mesh(domain, 120, "cgal")


# Build function space with Lagrange multiplier
P1 = FiniteElement("Lagrange", mesh.ufl_cell(), 1)
R = FiniteElement("Real", mesh.ufl_cell(), 0)
W = FunctionSpace(mesh, P1 * R)

# Define variational problem
(u, c) = TrialFunction(W)
(v, d) = TestFunctions(W)
f = Expression("0-4*exp(x[0]*x[0]+x[1]*x[1])*(x[0]*x[0]+x[1]*x[1]+1)
    ↪ ",element=P1)
g = Expression("2*exp(0)",element=P1)
a = (inner(grad(u), grad(v)) + c*v + u*d)*dx
L = f*v*dx + g*v*ds

# Compute solution
w = Function(W)
solve(a == L, w)
(u, c) = w.split()

# Plot solution
plot(u, interactive=True)
plot(mesh,interactive=True)
```

Code Snippet 7.4. This shows the bit of Mathematica code used to numerically integrate the derivatives. We tested the derivatives because they would be unaffected

by an unknown additive constants that the nature of the problem added to the solution.

```
NIntegrate[((2*y*Exp[x^2 + y^2]))^2, {x, y} \in Disk[{0, 0}, 1]]
```

Code Snippet 7.5. We first do some matrix multiplication to define the map F via a multiplication of C (whose columns are the points c_i) and p (a vector of polynomials that is a basis for P). Then we plot the image of K. By changing the choices of C, you see how you get a different curved element. For the image with the current choice of C, which pinches the mid points of the lines of K inward, see Figure 9.

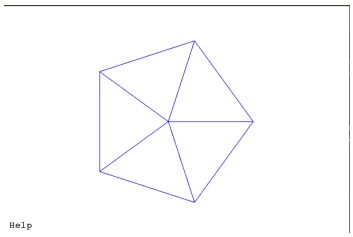
```
p = {x*(2*x - 1), y*(2*y - 1), (1 - x - y)*(2*(1 - x - y) - 1), 4*x*
    ↪ y,
   4*y*(1 - x - y), 4*x*(1 - x - y)}
C = {{1, 0, 0, 0.4, 0.1, 0.5}, {0, 1, 0, 0.4, 0.5, 0.1}}
ir = ImplicitRegion[x >= 0 && y >= 0 && x + y <= 1, {x, y}]
ParametricPlot[
 Dot[C, p], {x, y} \[Element]
  ImplicitRegion[x >= 0 && y >= 0 && x + y <= 1, {x, y}]]
```

Code Snippet 7.6. Using the modern software packages Firedrake and FIAT, this code shows how one can generate the maps assocaited to some isoparametric elements automatically. Here, you set d to request a certain degree polynomial. Then you fill in the matrix a wit the coordinates. This will then output the coordinates of the polynomial map so that one can plot them with software such as Mathematica.

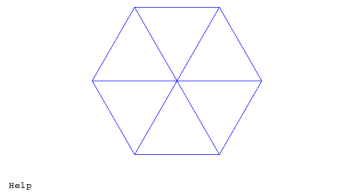
```
from firedrake import *
import tsfc
import sympy as sp
import numpy as np

#set me:
d=3 #the degree of the polynomials
dim = 2


mesh = UnitSquareMesh(1,1)
V = FunctionSpace(mesh,"Lagrange",degree=d)


cord_element = V.ufl_element()
rf = tsfc.fiatinterface.create_element(cord_element,vector_is_mixed=
    ↪ False)
symbols = [[sp.Symbol("x%d" % i) for i in xrange(V.mesh().
    ↪ topological_dimension())]]
basis = np.array((rf.tabulate(0, np.array(symbols)))[(0,0)])

numberOfNodes = len(basis)
```

```
a = np.ones((dim,numberOfNodes),dtype="float")
#overwite me just like C
#the above is the array of coordinates associated to the new map
#overwite it to change the map

polys = np.dot(a,basis)
p=polys.flatten()
print("The X1 cord poly is {0}".format(p[0]))
print("The X2 cord poly is {0}".format(p[1]))
#print("The Z cord poly is {0}".format(p[2])) add if dim =3
```

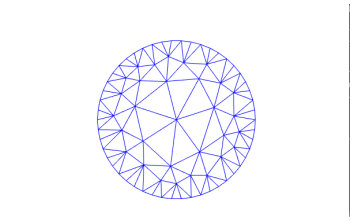
7.2. **Figures.**



(A) The first



(B) The second



(C) Sometime later

FIGURE 1. Polyhedral approximations to the unit disk.

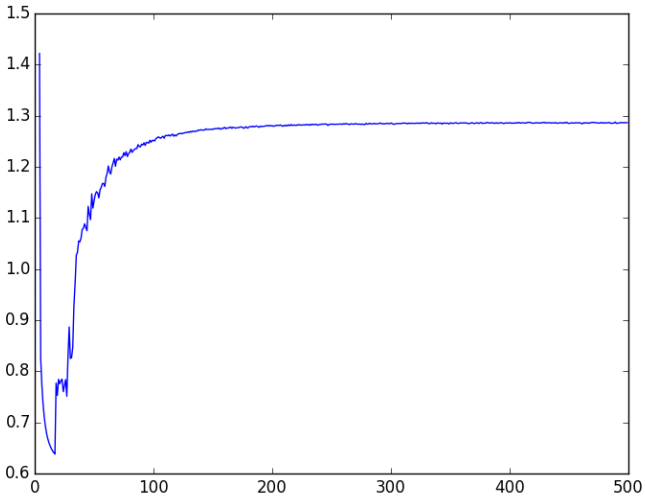


FIGURE 2. Convergence of L^2 norm as the approximation gets better.

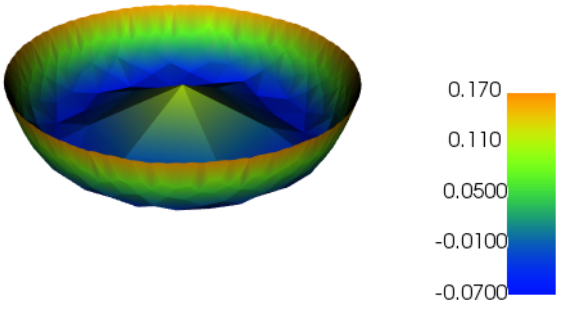


FIGURE 3. Plot of a solution.

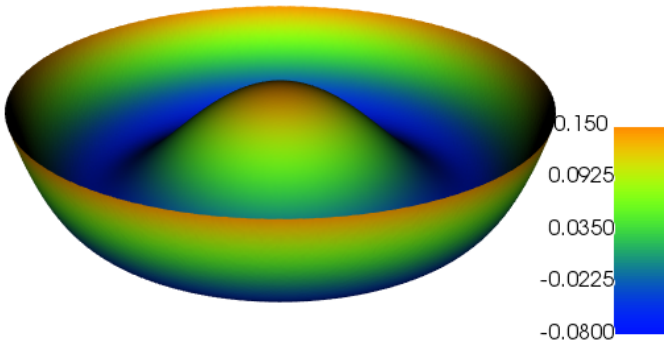


FIGURE 4. Plot of a solution.

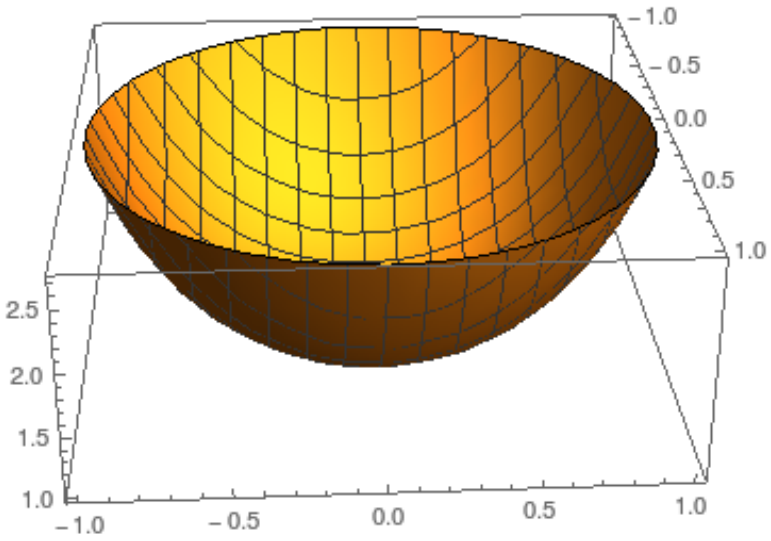


FIGURE 5. Plot of an analytic solution that the previous two images should conform to.

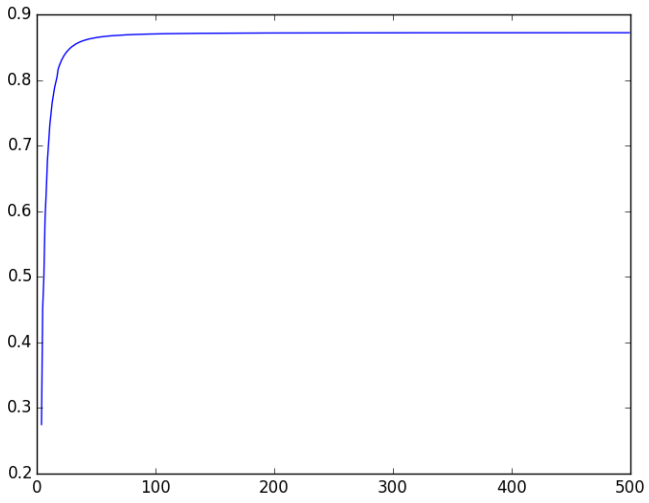


FIGURE 6. Convergence of L^2 norm as the approximation gets better.

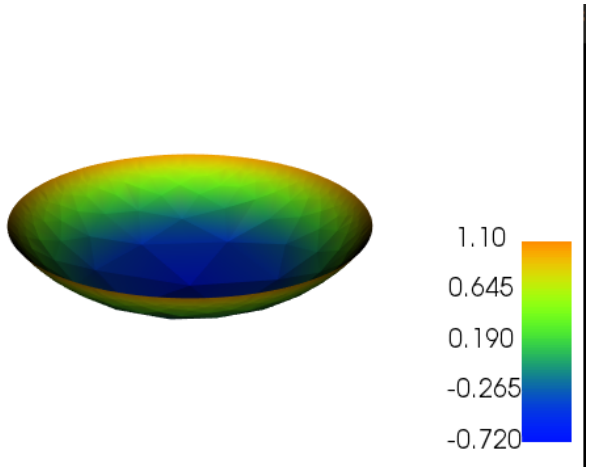


FIGURE 7. Plot of a solution with correct boundary conditions on the approximation.

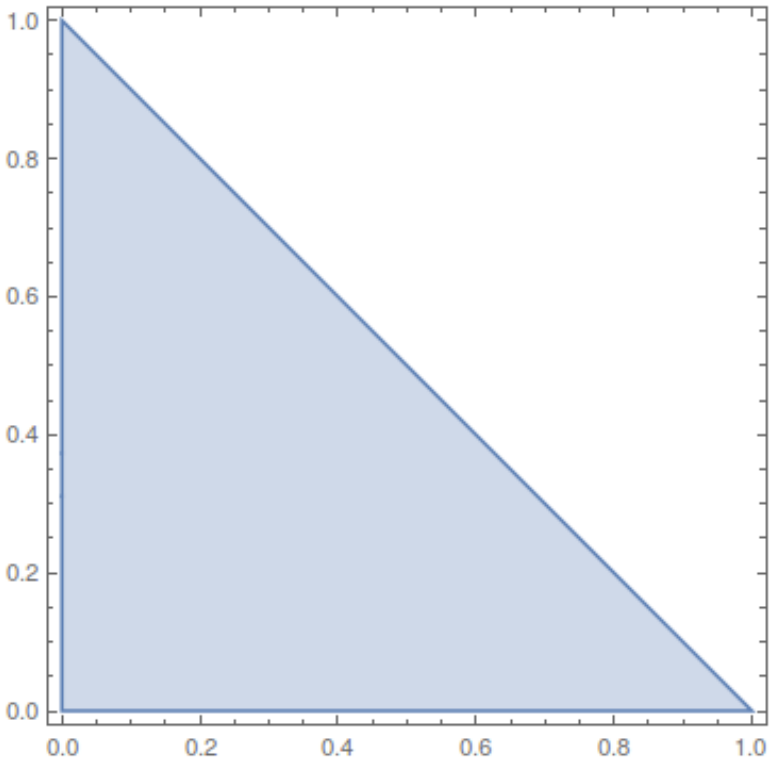


FIGURE 8. A base set for a finite element

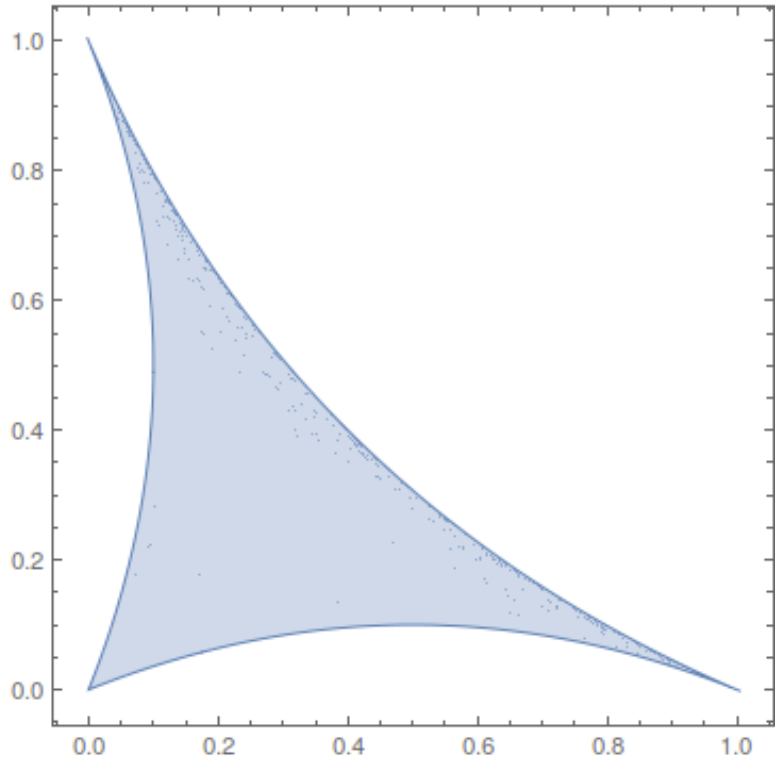


FIGURE 9. A possible image of Figure 8 via the code Code Snippet 7.5.