PARAMETRIC POLYNOMIAL PRESERVING RECOVERY ON MANIFOLDS

GUOZHI DONG* AND HAILONG GUO†

Abstract. This paper investigates gradient recovery schemes for data defined on discretized manifolds. The proposed method, parametric polynomial preserving recovery (PPPR), does not ask for the tangent spaces of the exact manifolds which have been assumed for some significant gradient recovery methods in the literature. Another advantage of the proposed method is that it removes the symmetric requirement from the existing methods for the superconvergence. These properties make it a prime method when meshes are arbitrarily structured or generated from high curvature surfaces. As an application, we show that the recovery operator is capable of constructing an asymptotically exact posteriori error estimator. Several numerical examples on 2–dimensional surfaces are presented to support the theoretical results and make comparisons with methods in the state of the art, which show evidence that the PPPR method outperforms the existing methods.

AMS subject classifications. Primary 65N50, 65N30; Secondary 65N15, 53C99

Key words. Gradient recovery, manifolds, superconvergence, parametric polynomial preserving, function value preserving, curvature stable.

1. Introduction. Numerical methods for approximating variational problems or partial differential equations (PDEs) with solutions defined on surfaces or manifolds are of growing interests over the last decades. Finite element methods, as one of the main streams in numerical simulations, are well established for those problems. A starting point can be traced back to [17], which is the first to investigate a finite element method for solving elliptic PDEs on surfaces. Since then, there have been a lot of extensions both in analysis and in algorithms, see for instance [9–11, 18, 26–28] and the references therein. In the literature, most of the works consider the *a priori* error analysis of various surface finite element methods, and only a few works, up to our best knowledge, take into account the *a posteriori* error analysis and superconvergence of finite element methods in a surface setting, see [5, 7, 8, 11, 12, 16, 30]. Recently, there is an approach proposed in [19] which merges the two types of analysis to develop a higher order finite element method on an approximated surface, where a gradient recovery scheme plays a key role. Gradient recovery techniques, which are important in *post processing* solutions or data for improving the accuracy of numerical simulations, have been widely studied and applied in many aspects of numerical analysis. In particular for planar problems, the study of gradient recovery methods has reached already a mature stage, and there is a massive of works in the literature, to name but only a few [1,4,21,24,31–34]. We point out some significant methods among them, like the classical Zienkiewicz–Zhu (ZZ) patch recovery method [33], and a later method called polynomial preserving recovery (PPR) [32]. The two approaches work with different philosophies in methodology. The former method first locates positions of certain points in the given mesh, and then recovers the gradients themselves at those points to achieve a higher order approximation accuracy, while the latter one first recovers the function values by polynomial interpolations in a local patch at each nodal points, and then takes gradients at the nodal points from the previously recov-

*Computational Science Center, University of Vienna, Oskar-Morgenstern-Platz 1, Wien, 1090 (guozhi.dong@univie.ac.at).

†Department of Mathematics, University of California, Santa Barbara, CA, 93106 (hlguo@math.ucsb.edu).

ered functions. Both the methods can produce comparable superconvergence results, but do not require the same assumptions on the discretized meshes.

Gradient recovery methods for data defined on curved spaces have only recently been investigated. In [30], several gradient recovery methods have been adapted to a general surface setting for linear finite element solutions which are defined on polyhedrons by triangulation. There a surface is concerned to be a zero level set of a smooth function defined in a higher dimensional space, which is from the point of view of an ambient space of the surface. It has been shown that most of the properties of the gradient recovery schemes for planar problems are maintained in their counterparts for surface problems. In particular, in their implementation and analysis, the methods ask for exact knowledge of the surface, e.g. the nodal points are located on the exact surface, and the tangent spaces or in another word the normal vector field are given. However, this information is usually not available in reality, where we have only the approximations of surfaces, for instance, polyhedrons, splines or polynomial surfaces. On the other hand, the generalized ZZ scheme for gradient recovery with surface elements gives the most competitive results in [30], including several other methods, their superconvergence are proved with the assumption that the local patch is $\mathcal{O}(h^2)-$symmetric on the discretized surfaces, which is restrictive in applications. In the planar case, the $\mathcal{O}(h^2)-$symmetric condition is also asked for the superconvergence by these methods which have been generalized to a surface setting in [30], but it is not necessary for the PPR method.

This difference gives us the motivation to generalize the PPR method for problems with data defined on manifolds. A follow-up question would be what are the polynomials in the domains of curved manifolds. Using the idea from the literature, e.g. [16], one could consider polynomials locally on the tangent spaces of the manifolds. Obviously, a direct generalization of PPR to a manifold setting based on tangent spaces will again fall into the awkward situation: The exact manifold and its tangent spaces are unknown.

To overcome these difficulties, we go back to the original definition of a manifold. We take the manifold as patches locally parametrized by Euclidean planar domains, but not necessarily by their tangent spaces. This has no interruption for us to define patch-wise polynomials in such planar parameter domains. In this manner, we are able to recover the unknown surfaces from the given sampling points in these local domains, as well as the finite element solutions iso-parametrically. Our proposed method is thus called parametric polynomial preserving recovery (PPPR) which *does not* rely on the $\mathcal{O}(h^2)-$ symmetric condition for the superconvergence, just like its genetic father PPR. To this end, it will be revealed that PPPR is particularly useful to *address the issue of unavailable tangent spaces*, and thus it enables us to solve the open issues in [30]. Another benefit of the PPPR method for data on a surface is that it is relatively *curvature stable* in comparing with the methods proposed in [30]. This is verified by our numerical examples, but a quantitative analysis will be open in the paper. Moreover, the original PPR method [32] does not preserve the function values at the nodal points in its pre-recovery step. In this paper, we provide an alternative method which can achieve this goal. With this option, the PPPR can not only preserve *parametric polynomial*, but also preserve the *surface sampling points* and the *function values* at the given points simultaneously. That means the given data is invariant during the recovery by using the PPPR method.

The rest of the paper is organized as follows: Section 2 gives a preliminary account on relevant differential geometry concepts and an analytic PDE problem. Section 3

introduces discretized function spaces and collects some geometric notations used in the paper. Section 4 presents the new algorithms especially the PPPR for gradient recovery on manifolds. There we make remarks on the comparison of algorithms and the idea of preserving function values, and provide an argument for its curvature stable property. Section 5 gives a brief analysis of the superconvergence properties of the proposed method. Section 6 analyze the recovery-based *a posteriori* estimator using the new gradient recovery operators. Finally, we present some numerical results and the comparisons with existing methods in Section 7. We have a proof of a basic lemma in Appendix A.

2. Background. We will only show some basic concepts which are relevant to our paper. For a more general overview on the topic of Riemannian geometry or differential geometry, one could refer to for instance [13, 25]. In the context of the paper, we shall consider $(\mathcal{M}, g)$ as an oriented, connected, C^3 smooth and compact Riemannian manifold without boundary, where g denotes the Riemann metric tensor. The idea we are going to work should be no restriction for general n dimensional manifolds, but we will focus on the case of two dimensional ones, which are also called surfaces, in the later applications and numerical examples.

Our concerns are some quantities $u : \mathcal{M} \to \mathbb{R}$ which are scalar functions defined on manifolds. First, let us mention the differentiation of a function u in a manifold setting, which is called covariant derivatives in general. It is defined as the directional derivatives of the function u along an arbitrarily selected path γ on the manifold

$$D_{\mathbf{v}} u = \frac{du(\gamma(t))}{dt}|_{t=0},$$

here $\mathbf{v} = \gamma(t)'|_{t=0}$ is a tangential vector field.

The gradient then is an operator such that

$$(\nabla_g u(x), \mathbf{v}(x))_g = D_{\mathbf{v}} u, \quad \text{for all } \mathbf{v} \in T_x\mathcal{M} \text{ and all } x \in \mathcal{M}.$$

It is not harm to think of the gradient as a tangent vector field on the manifold $\mathcal{M}$. In a local coordinate, the gradient has the form

$$\nabla_g u = \sum_{i,j} g^{ij} \partial_j u \partial_i, \tag{2.1}$$

where g^{ij} is the entries of the inverse of the metric tensor g, and ∂_i denotes the tangential basis. Let $\mathbf{r} : \Omega \to S \subset \mathcal{M}$ be a local geometric mapping, then we can rewrite (2.1) into a matrix form with this local parametrization, that is

$$(\nabla_g u) \circ \mathbf{r} = \nabla \hat{u} (g \circ \mathbf{r})^{-1} \partial \mathbf{r}. \tag{2.2}$$

In (2.2), $\hat{u} = u \circ \mathbf{r}$ is the pull back of function u to the local planar parameter domain Ω, ∇ denotes the gradient on the planar domain Ω, $\partial \mathbf{r}$ is the Jacobian matrix of $\mathbf{r}$, and $g \circ \mathbf{r} = \partial \mathbf{r} (\partial \mathbf{r})^T$ on this patch.

REMARK 2.1. *$\mathbf{r}$ is not specified here, and we will make it clear when it becomes necessary later. We actually have a relation that*

$$(\partial \mathbf{r})^\dagger = (g \circ \mathbf{r})^{-1} \partial \mathbf{r}, \tag{2.3}$$

where $(\partial \mathbf{r})^\dagger$ denotes the Moore-Penrose inverse of $\partial \mathbf{r}$. See [14, Appendix] for a detailed explanation.

Note that the parametrization map $\mathbf{r}$ is not unique, typical ones can be constructed through function graphs which will be used in our later algorithms. We have the following lemma of which the proof is given in Appendix A.

LEMMA 2.1. *The gradient* (2.2) *is invariant under different chosen of regular isomorphic parametrization functions* $\mathbf{r}$.

Let $\omega = dvol$ be the volume form on $\mathcal{M}$, and ∂_j $(j = 1, \cdots, n)$ be the tangential bases. For every tangent vector field $\mathbf{v} : \mathcal{M} \to T\mathcal{M}$, $\mathbf{v} = v^i \partial_i$, we have a $n-1$ form defined by the interior product of $\mathbf{v}$ and the volume form ω through the following way

$$i_{\mathbf{v}}\omega = \sum_k \omega(\mathbf{v}, \partial_{k_1}, \cdots, \partial_{k_{n-1}}),$$

where $k_1, \cdots, k_{n-1}$ are $n-1$ indexes with k taking out from $1, \cdots, n$. The divergence of the vector field $\mathbf{v}$ then satisfies

$$d(i_{\mathbf{v}}\omega) = \mathrm{div}_g(\mathbf{v})\omega, \tag{2.4}$$

where d denotes the exterior derivative. Since both the left hand side and the right hand side of (2.4) are n forms, $\mathrm{div}_g(\mathbf{v})$ is a scalar field. Using the local coordinates, we can write the volume form explicitly

$$\omega = \sqrt{|\det g|} dx^1 \wedge \cdots \wedge dx^n.$$

Applying equation (2.4), the divergence of the vector field $\mathbf{v}$ can be computed by

$$\mathrm{div}_g \mathbf{v} = \frac{1}{\sqrt{|\det g|}} \partial_i (v^i \sqrt{|\det g|}).$$

It is revealed that the divergence operator is actually the dual of the gradient operator. With the above preparation, we can now given the definition of the Laplace-Beltrami operator, which is denoted by Δ_g in our paper, as the divergence of the gradient, that is

$$\Delta_g u = \mathrm{div}_g(\nabla_g u) = \frac{1}{\sqrt{|\det g|}} \partial_i (g^{ij} \sqrt{|\det g|} \partial_j u).$$

We mention that if the manifold $\mathcal{M}$ is a hyper-surface, that is $\mathcal{M} \subset \mathbb{R}^{n+1}$ which has co-dimension 1. The gradient and divergence of the function u can be equally calculated through projecting the gradient and divergence of an extended function in ambient space $\mathbb{R}^{n+1}$ to the tangent spaces of $\mathcal{M}$ respectively. This type of definitions has been applied in many references which consider problems in an ambient space setting, i.e.

$$\nabla_g u = (\mathcal{P}_T \nabla_e) u_e, \text{ and } \mathrm{div}_g \mathbf{v} = (\mathcal{P}_T \nabla_e) \cdot \mathbf{v}_e,$$

where u_e and $\mathbf{v}_e$ are the extended scalar and vector fields defined in the ambient space of the hypersurface, which satisfies $u_e(x) = u(x)$ and $\mathbf{v}_e(x) = \mathbf{v}(x)$ for all $x \in \mathcal{M}$. Note that $\nabla_e \cdot$ is the gradient operator defined in the ambient Euclidean space $\mathbb{R}^{n+1}$, $\mathcal{P}_T$ is the tangential projection operator

$$\mathcal{P}_T = \mathrm{Id} - \mathbf{n} \otimes \mathbf{n},$$

and $\mathbf{n}$ is a unit normal vector field of $\mathcal{M}$. It can be showed that the gradient and divergence by projections are independent of the way of the extension of the scalar or vector fields, and they are equivalent to the former definitions.

With the generalized notions of the differentiation on manifolds, the function spaces based on manifold domains can be studied analogously to Euclidean domains. Sobolev spaces on manifolds [23] are one of the mostly investigated spaces, which provide a breeding ground to study PDEs. We are interested in numerically approximating PDEs of which the solutions are defined on $\mathcal{M}$. Even though our methods are *problem independent*, in this paper, our analysis will be mainly based on the Laplace-Beltrami operator (2.5), and its generated PDEs. For the purpose of both analysis and applications, we consider an exemplary problem [17] the *Laplace-Beltrami equation*, that is for a given f satisfying $\int_{\mathcal{M}} f \, dvol = 0$ to solve the equation

$$-\Delta_g u = f \ \text{on} \ \mathcal{M}, \ \ \text{with} \int_{\mathcal{M}} u \, dvol = 0, \tag{2.5}$$

where *dvol* denotes the manifold volume measure.

3. Function Spaces on Discretized Manifolds. The discretization of a smooth manifold $\mathcal{M}$ has been widely studied in many settings, especially in terms of surfaces [18]. A discretized surface, in most cases, is a piecewise polynomial surface. One of the most simple case is the polygonal approximation to a given smooth surface, especially with triangulations. Finite element methods for triangulated meshes on surfaces have firstly been studied in [17] by using linear elements. In [10], a generalization of [17] to high order finite element method is proposed based on triangulated surfaces. In order to have an optimal convergence rates, it is showed that the geometric approximation error and the function approximation error has to be compatible with each other. In fact, the balance of geometric approximation errors and function approximation errors is also the key point in the development of our recovery algorithm.

In this paper, we will denote $\mathcal{M}_h = \bigcup_{j\in J_h} \mathcal{T}_{h,j}$ the triangulated surface, where $(\mathcal{T}_{h,j})_{j\in J_h}$ is the set of triangles, and $h = \max_{j\in J_h} D(\mathcal{T}_{h,j})$ is the maximum diameter. We restrict ourselves to the first order finite element methods, thus the nodes consist of simply the vertices of $\mathcal{M}_h$, which we denote by $(x_i)_{i\in I_h}$.

In the following, we define transform operators between the function spaces on $\mathcal{M}$ and function spaces on $\mathcal{M}_\delta$, where $\mathcal{M}_\delta$ denotes some perturbation of $\mathcal{M}$.

$$\begin{aligned} T_\delta : \mathcal{V}(\mathcal{M}) &\to \mathcal{V}_\delta(\mathcal{M}_\delta); \\ v &\mapsto v \circ P_\delta, \end{aligned} \tag{3.1}$$

and its inverse

$$\begin{aligned} (T_\delta)^{-1} : \mathcal{V}_\delta(\mathcal{M}_\delta) &\to \mathcal{V}(\mathcal{M}); \\ v_\delta &\mapsto v_\delta \circ P_\delta^{-1}, \end{aligned} \tag{3.2}$$

where P_δ is a continuous and bijective projection map from $\mathcal{M}_\delta$ to $\mathcal{M}$.

We have the following lemma with triangulated approximation $\mathcal{M}_\delta = \mathcal{M}_h$.

LEMMA 3.1. *For $\mathcal{V}(\mathcal{M}) \hookrightarrow W^{k,p}(\mathcal{M})$ $k \geq 0$, $p \geq 1$, the transform operators $(T_h)^{\pm 1}$ are uniformly bounded between the spaces $\mathcal{V}(\mathcal{M})$ and $\mathcal{V}_h(\mathcal{M}_h)$ as long as the space $\mathcal{V}_h(\mathcal{M}_h)$ is compatible with the smoothness of $\mathcal{M}_h$.*

Proof. For every $v \in \mathcal{V}(\mathcal{M})$, denote $\check{v}_h := T_h v$. Each triangular faces $\mathcal{T}_{h,j}$ of $\mathcal{M}_h$ corresponding to a curved triangle faces on $\mathcal{M}$, and we denote it as $\mathcal{T}_j$. If $p = \infty$, every function v and its derivatives are uniformly bounded on $\mathcal{M}$, then we can always find constants c and C such that

$$c \left\| \check{v}_h \right\|_{k,\infty,\mathcal{M}_h} \leq \left\| v \right\|_{k,\infty,\mathcal{M}} \leq C \left\| \check{v}_h \right\|_{k,\infty,\mathcal{M}_h}.$$

For $1 \leq p < \infty$, using the results in [10, page 811], we have the equivalence of $\|v\|_{k,p,\mathcal{T}_j}$ and $\|\check{v}_h\|_{k,p,\mathcal{T}_{h,j}}$. That is there exists positive and uniformly bounded constants $c_{h,j}$ and $C_{h,j}$, such that

$$c_{h,j} \|\check{v}_h\|^p_{k,p,\mathcal{T}_{h,j}} \leq \|v\|^p_{k,p,\mathcal{T}_j} \leq C_{h,j} \|\check{v}_h\|^p_{k,p,\mathcal{T}_{h,j}},$$

holds on each pair of the triangular faces. Since $\|v\|^p_{k,p,\mathcal{M}} = \sum_{j\in J_h} \|v\|^p_{k,p,\mathcal{T}_j}$ we have then the estimate

$$\min\{c_{h,j}\} \|\check{v}_h\|^p_{k,p,\mathcal{M}_h} \leq \|v\|^p_{k,p,\mathcal{M}} \leq \max\{c_{h,j}\} \|\check{v}_h\|^p_{k,p,\mathcal{M}_h},$$

which gives the conclusion. □

REMARK 3.1. *The statement of Lemma 3.1 hold also for higher order continuous piece-wise polynomial approximation of $\mathcal{M}$.*

We give here an assumption on the triangulations of surfaces, which is a common condition to have the so-called supercloseness.

ASSUMPTION 3.2. *$\mathcal{M}_h$ is a quasi-uniform and shape regular triangulation of $\mathcal{M}$, and it satisfies the $\mathcal{O}(h^{2\sigma})$ irregular condition (cf. [4, Definition 2.4], or [30, Definition 3.2]).*

For convenience, Table 3.1 collects some notations in the paper.

Table 3.1: Notations

Notation	Remark
$(\mathcal{M}, g)$	a smooth, connected, oriented and close manifold with metric g
$(\mathcal{M}_h, g_h)$	a polyhedral approximation of $\mathcal{M}$ with piece-wise smooth metric g_h
$\mathbf{n}$	a unit normal vector field on $\mathcal{M}$
∇_g	gradient operator with respect to the metric g
Δ_g	Laplace-Beltrami operator with respect to the metric g
T_x	the tangent space at a position $x \in \mathcal{M}$
$(P_h)^{\pm 1}$	bijective maps between $\mathcal{M}_h$ and $\mathcal{M}$
$(T_h)^{\pm 1}$	operators map between function spaces on $\mathcal{M}$ and on $\mathcal{M}_h$
$(T_\delta)^{\pm 1}$	operators map between function spaces on $\mathcal{M}$ and on $\mathcal{M}_\delta$
h	the diameter of the triangulation mesh in $\mathcal{M}_h$
Ω	a planar domain which locally parametrize a patch of $\mathcal{M}$
ζ	a position variable in the parameter domain Ω
$\mathbf{r}$(or $\mathbf{r}_h$)	a local parametrization map from Ω to a patch of $\mathcal{M}$ (or $\mathcal{M}_h$)
vol(or vol_h)	the volume (area) measure of $\mathcal{M}$ (or $\mathcal{M}_h$)
$\|\cdot\|_{k,p,\mathcal{M}}$	$W^{k,p}$ norm of functions defined on $\mathcal{M}$
$\|\cdot\|_{k,p,\mathcal{M}}$	$W^{k,p}$ semi-norm of functions defined on $\mathcal{M}$
$\|\cdot\|_{k,\mathcal{M}}$	H^k norm of functions defined on $\mathcal{M}$
I_h	the total number of the nodal points (vertices) of $\mathcal{M}_h$
J_h	the total number of the triangles on $\mathcal{M}_h$
$P^2(\Omega)$	the 2^{nd} order polynomial space over a planar domain Ω

4. Parametric Polynomial Preserving Recovery on Manifolds. Our developments are based on the so-called PPR method proposed in [32] for planar problems. It is a robust and high accuracy approach for recovering gradient on mildly

structured triangular meshes. This idea has been developed to recover Hessian in a recent paper [22]. In the current paper, we show the possibility of some generalizations into a manifold domain. To simplify the presentation, we shall restrict to the case of 2 dimensional manifolds here and after.

We will mainly consider $\mathcal{M}_h$ a triangulated polyhedron approximation of $\mathcal{M}$, and post-process with first order finite element solutions. At each node x_i, let $B(x_i)$ the set of vertices of the triangles which share the node x_i as a common vertex. For a discretized manifold, a main difficulty is that the vertices in $B(x_i)$ are in general not located on a common plane, another difficulty is that there is no trivial definition of polynomials in a manifold setting. Some idea appeared in the literature is using the tangent space T_{x_i} at every vertex x_i as a local parameter domain, and projecting the neighboured vertices of x_i onto this common planar plane, then defining polynomials locally by the coordinates of the tangent space. This idea has been applied in [16] and also in [30] to generalize the ZZ method and several other methods. However, the exact manifold $\mathcal{M}$ is usually not given in real problems, therefore the tangent spaces $(T_{x_i})_{i\in I_h}$ of $\mathcal{M}$ are blind to users, which makes the idea not much reliable in practice. This problem has also been claimed as an open issue in [30].

As a starting point, we first provide an direct generalization of the PPR method based on given tangent spaces of the exact manifold $\mathcal{M}$. In this case, the algorithm is pretty much as the same as the planar one. We sketch it in Algorithm 1.

Algorithm 1 PPR Method (*with Information of Exact Surfaces*)

Let the discretized triangular surface $\mathcal{M}_h$ and the data (FEM solutions) $(u_{h,i})_{i\in I_h}$ be given, and the vertices $(x_i)_{i\in I_h}$ be located on the exact surface $\mathcal{M}$. Also, we have the the normal vector $(\mathbf{n}_i)_{i\in I_h}$ of $\mathcal{M}$ at each vertex x_i. Then repeat steps $(1)-(3)$ for all $i \in I_h$.

(1) At each vertex x_i, select all the neighbored vertices within a sufficiently big geodesic ball $B(x_i)$ such that the rank condition[a] is satisfied. Shift x_i to be the origin of T_{x_i}, and choose an orthonormal basis (τ_1^i, τ_2^i) of T_{x_i}, then project the vertices $x_j \in B(x_i)$ to T_{x_i} with the new coordinates denoted by ζ_{i_j}. I_i denotes the indexes of the selected vertices in $B(x_i)$.

(2) Reconstruct an interpolation polynomial p_i by the coordinates of the tangent plane from the FEM solutions at the given vertices, where p_i is the minimizer

$$p_i = \arg\min_p \sum_{j\in I_i} |p(\zeta_{i_j}) - u_{h,j}|^2 \ \text{ for } p \in P^2(T_{x_i}).$$

(3) Calculate the partial derivatives of the approximated polynomial functions, then we have the recovered gradient at each vertex x_i

$$G_{1,h}u_h(x_i) = \partial_1 p_i(0,0)\tau_1^i + \partial_2 p_i(0,0)\tau_2^i. \tag{4.1}$$

For the recovery of the gradient $G_{1,h}u_h$ on the whole $\mathcal{M}_h$, we propose to interpolate the values $\{G_{1,h}u_h(x_i)\}_{i\in I_h}$ by using linear finite element basis on each triangles.

[a] The rank condition is asked in the step (2) in order to have a stable solution of the least square problem, see [32] for a detailed discussion.

In fact, Algorithm 1 continues with the ambient space setting for surfaces. We point out that a straight forward remedy for missing exact normal fields is to find a

way to approximate normal vectors at every vertex x_i, for instance, by simple average or weighted average of the normal vectors of each faces adjunct to x_i. However, in this way, the recovery error will very likely be dominated by the error of the approximation of the normal vector fields. See also the numerical results in Section 7.

In the following, we shall provide another algorithm which is neither relying on the information of the tangent spaces, nor the knowledge of the exact surface. Our idea goes to the gradient formulation (2.2), which is able to calculate the gradient from an arbitrary local parametrization (either exact or approximated). It is in fact an intrinsic point of view for manifolds.

Algorithm 2 PPPR Method (*without Asking for Exact Surfaces*)

Let the discretized triangular surface $\mathcal{M}_h$ and the data (FEM solutions) $(u_{h,i})_{i\in I_h}$ be given. Then repeat steps $(1)-(4)$ for all $i \in I_h$.

(1) At each vertex x_i, select all the neighbored vertices within a sufficiently big geodesic ball $B(x_i)$, such that the rank condition is satisfied. Construct a local parameter domain Ω_i, and shift x_i to be the origin of Ω_i, and choose (ϕ_1^i, ϕ_2^i) the orthonormal basis of Ω_i, and ϕ_3^i a unit orthogonal vector to Ω_i, then project all selected vertices $x_j \in B(x_i)$ into the parameter domain Ω_i, with the new coordinates denoted by ζ_{i_j}. The principle of finding such Ω_i is to make sure the triangles are shape regular after projection. I_i denotes the indexes of the selected vertices in $B(x_i)$.

(2) Reconstruct a 2^{nd} order polynomial surface S_i over Ω_i by interpolating the chosen vertices in the new coordinates. Typically, we can approximate the surface locally as a function graph parametrized by Ω_i. That is $S_i = \mathbf{r}_{h,i}(\Omega_i) = \bigcup_{\zeta\in\Omega_i}(\zeta, s_i(\zeta))$, where s_i is given as

$$s_i = \arg\min_s \sum_{j\in I_i} |s(\zeta_{i_j}) - \langle x_j, \phi_3^i\rangle|^2 \quad \text{for } s \in P^2(\Omega_i).$$

(3) Reconstruct a 2^{nd} order polynomial p_i over Ω_i to interpolate the solutions at the given vertices of each $B(x_i)$

$$p_i = \arg\min_p \sum_{j\in I_i} |p(\zeta_{i_j}) - u_{h,j}|^2 \quad \text{for } p \in P^2(\Omega_i).$$

(4) Calculate the partial derivatives of both the polynomial approximated surface function in the second step and the approximated polynomial function of FEM solution in the third step, then we can approximate the gradient which is given in (2.2). In the local coordinates,

$$G_{2,h}u_h(x_i) = \begin{pmatrix} 1 & 0 & \partial_1 s_i(0,0) \\ 0 & 1 & \partial_2 s_i(0,0) \end{pmatrix}^{\dagger} \begin{pmatrix} \partial_1 p_i(0,0) \\ \partial_2 p_i(0,0) \end{pmatrix} \begin{pmatrix} \phi_1^i & \phi_2^i & \phi_3^i \end{pmatrix}. \tag{4.2}$$

The last equation here is given by (2.3) in the remark 2.1 for calculating (2.2). To multiply with the orthonormal basis $\{\phi_1^i, \phi_2^i, \phi_3^i\}$ is because we have to unify the coordinates from local ones to a global one.

For the recovery of the gradient $G_{2,h}u_h$ on the whole $\mathcal{M}_h$, we propose to interpolate the values $\{G_{2,h}u_h(x_i)\}_{i\in I_h}$ by using linear finite element basis on each triangles.

REMARK 4.1. *We point out that if $\Omega_i = T_{x_i}$ for all $i \in I_h$, and we shift x_i to be the origin of T_{x_i}, then $\partial_1 s_i(0,0) = \partial_2 s_i(0,0) \equiv 0$ for all $i \in I_h$, and $\phi_1^i = \tau_1^i$, $\phi_2^i = \tau_2^i$, $\phi_3^i = \mathbf{n}_i$. It is easy to find that the recovered gradient in* (4.2) *is equal to the one recovered in* (4.1). *That is Algorithm 2 actually generalizes Algorithm 1. Thus in a later analysis in Section 5, we do not distinguish the notation G_h for the recovery operator given either by Algorithm 1 or by Algorithm 2. However, we stress that the two algorithms are* not *equivalent if Algorithm 1 use* approximated *normal vector fields, in this case, we will use $G_{1,h}^a$ for the operator.*

REMARK 4.2. *In fact, in both Algorithm 1 and Algorithm 2, we have another alternative way for the polynomial reconstruction instead of the one presented in the algorithms, which is originally proposed in [32]. The method in [32] assumes that a second order polynomial has a form*

$$p(y) = a_0 + a_1 y_1 + a_2 y_2 + a_3 y_1^2 + a_4 y_1 y_2 + a_5 y_2^2, \ \textit{for} \ \ y = (y_1, y_2) \in \Omega_i$$

then solving the linear system $A\mathbf{a} = \mathbf{b}$ for $\mathbf{a} = (a_0, a_1, \cdots, a_5)^T$, where

$$A = \begin{pmatrix} 1 & \zeta_{i_1,1} & \zeta_{i_1,2} & \zeta_{i_1,1}^2 & \zeta_{i_1,1}\zeta_{i_1,2} & \zeta_{i_1,2}^2 \\ 1 & \zeta_{i_2,1} & \zeta_{i_2,2} & \zeta_{i_2,1}^2 & \zeta_{i_2,1}\zeta_{i_2,2} & \zeta_{i_2,2}^2 \\ \vdots & \cdots & & & & \\ 1 & \zeta_{i_{|I_i|},1} & \zeta_{i_{|I_i|},2} & \zeta_{i_{|I_i|},1}^2 & \zeta_{i_{|I_i|},1}\zeta_{i_{|I_i|},2} & \zeta_{i_{|I_i|},2}^2 \end{pmatrix} \ \textit{and} \ \mathbf{b} = \begin{pmatrix} u_{h,i_1} \\ u_{h,i_2} \\ \vdots \\ u_{h,i_{|I_i|}} \end{pmatrix}. \tag{4.3}$$

The solution of the least square approximation in the algorithms is given by

$$\mathbf{a} = (A^T A)^{-1} A^T \mathbf{b},$$

which tells that $\partial_1 p(0,0) = a_1$ and $\partial_2 p(0,0) = a_2$.

Our observation is that there are some extra freedom can be reduced in the reconstruction of the polynomials. Since the polynomial recovery can not improve the accuracy of the solution itself, it is unnecessary to adapt the solution in gradient recovery. We can fix this problem by using the following polynomial equation locally

$$\bar{p}(y) = u_{h,i_1} + \bar{a}_1 y_1 + \bar{a}_2 y_2 + \bar{a}_3 y_1^2 + \bar{a}_4 y_1 y_2 + \bar{a}_5 y_2^2, \ \textit{for} \ \ y = (y_1, y_2) \in \Omega_i$$

where u_{h,i_1} is the finite element solution at the vertex x_i. Let $\zeta_{i_1} = (\zeta_{i_1,1}, \zeta_{i_1,2})$ be the origin $(0,0)$ of the plane Ω_i, then the matrix and the vector in (4.3) *can be simplified to*

$$\bar{A} = \begin{pmatrix} \zeta_{i_2,1} & \zeta_{i_2,2} & \zeta_{i_2,1}^2 & \zeta_{i_2,1}\zeta_{i_2,2} & \zeta_{i_2,2}^2 \\ \vdots & \cdots & & & \\ \zeta_{i_{|I_i|},1} & \zeta_{i_{|I_i|},2} & \zeta_{i_{|I_i|},1}^2 & \zeta_{i_{|I_i|},1}\zeta_{i_{|I_i|},2} & \zeta_{i_{|I_i|},2}^2 \end{pmatrix} \ \textit{and} \ \bar{\mathbf{b}} = \begin{pmatrix} u_{h,i_2} - u_{h,i_1} \\ \vdots \\ u_{h,i_{|I_i|}} - u_{h,i_1} \end{pmatrix}. \tag{4.4}$$

Solving the problem in the least square sense

$$\bar{\mathbf{a}} = (\bar{A}^T \bar{A})^{-1} \bar{A}^T \bar{\mathbf{b}},$$

then we have $\partial_1 \bar{p}(0,0) = \bar{a}_1$ and $\partial_2 \bar{p}(0,0) = \bar{a}_2$.

Using (4.4) *in stead of* (4.3), *the gradient recovery algorithms not only preserve polynomials but also preserve the function values at the recovered nodal points. This*

idea can be applied to replace the polynomial reconstruction in the 2^{nd} step of Algorithm 1, and also to replace the polynomial reconstruction in both the 2^{nd} and the 3^{rd} step of Algorithm 2.

REMARK 4.3. *We claim that the PPPR (also the PPR with exact normal field) gives the most competitive results for the recovery of gradient when the approximated surface is featured with some high curvature. Our argument is that, in the planar case, the PPR is the most robust method with respect to unstructured meshes in comparing with the other methods. For a surface with complicated curvature, a well structured triangulation after projecting to the parametric domains or tangent spaces, is very likely not keeping the good structure any more. The PPPR method is in fact using the PPR to reconstruct both the tangent vectors of the surface and the gradient of the solutions in local parametric domains, which is more stable than the other methods for those mildly structured meshes projected from the high curvature areas. This claim has been supported by our numerical tests, see Numerical Example 2 in Section 7, however, a quantitative analysis on this property is open for future.*

Lemma 2.1 indicates that for every fixed x_i, taking arbitrary Ω_i, the gradient operator is analytically invariant. We then suppose the PPPR should work properly for different choice of Ω_i at a fixed x_i. The main issue is that, numerically, the shape of the triangles must not be destroyed after projecting them to the domain Ω_i. Thus, we still have to find a proper way for this projection. In practice, we can use the approximating normal vectors to help us to locate and orient a suitable parameter domain Ω_i, and this is what we have done in our numerical test. Our numerical results show that the accuracy of the approximations of normal vectors has very little influence to the recovery accuracy of the gradient by the PPPR, which is contrast to the case for Algorithm 1 where the recovery accuracy highly relies on the error of the approximating normal vectors. The nature of Algorithm 2 allows us to apply the analysis of the PPR which has been developed for planar problems.

5. Superconvergence Analysis. In the following, we shall show the superconvergence property of the proposed algorithms. Although our algorithms are problem independent, in order to make the discussion simple, we will take the equation (2.5) as our example, and discuss with the approximation by the first order finite element methods on triangulated surfaces. The variational formulation of problem (2.5) is given as follows: Find $u \in H^1(\mathcal{M})$ such that

$$\int_{\mathcal{M}} \nabla_g u \cdot \nabla_g v \; dvol = \int_{\mathcal{M}} fv \; dvol \text{ for all } \; v \in H^2(\mathcal{M}). \tag{5.1}$$

The regularity of the solutions has been proved in [2, Chapter 4]. Using finite element methods, the surface $\mathcal{M}$ is approximated by the triangulation $\mathcal{M}_h$ which satisfy Assumption 3.2, and the solution is simulated in the piecewise linear function spaces $\mathcal{V}_h$ defined over $\mathcal{M}_h$,

$$\int_{\mathcal{M}_h} \nabla_{g_h} u_h \cdot \nabla_{g_h} v_h \; dvol_h = \int_{\mathcal{M}_h} f_h v_h \; dvol_h \text{ for all } \; v_h \in \mathcal{V}_h(\mathcal{M}_h). \tag{5.2}$$

LEMMA 5.1. *Let G_h be the gradient recovery operator by Algorithm 1 or 2, then it is a bounded linear operator and preserves polynomial in every parametric domain Ω_i for all $i \in I_h$. Moreover, if $u \in H^3(\mathcal{M}) \bigcap W^{2,\infty}(\mathcal{M})$, then we have the estimate*

$$\left\| \nabla_g u - T_h^{-1} G_h (T_h u) \right\|_{0,\infty,\mathcal{M}} \leq C h^2 |u|_{3,\mathcal{M}}. \tag{5.3}$$

Proof. Since on the parametric domain Ω_i the operator G_h is using PPR method both for surface interpolation and for function interpolation, which are $\mathbf{r}$ and $u \circ \mathbf{r}$ defined on the local parametric domain Ω_i respectively. Thus, we refer to [32, Theorem 2.1] (See [20] for proof of sharp estimate for PPR) for the statement of preserving polynomial on every parametric domain Ω_i for both the local geometric mapping $\mathbf{r}$ and the function $u \circ \mathbf{r}$. Then we can have the estimate (5.3) by using Hilbert–Bramble lemma and triangle inequality on each Ω_i. ∎

THEOREM 5.2. *Let Assumption 3.2 hold for $\mathcal{M}_h$, and let $u \in H^3(\mathcal{M}) \bigcap W^{2,\infty}(\mathcal{M})$ be the solution of* (5.1)*, and u_h be the solution of* (5.2)*, then*

$$\left\|\nabla_g u - T_h^{-1} G_h u_h\right\|_{0,\mathcal{M}} \lesssim h^{1+\min\{1,\sigma\}}(\|u\|_{3,\mathcal{M}} + \|u\|_{2,\infty,\mathcal{M}}) + h^2 \|f\|_{0,\mathcal{M}}. \tag{5.4}$$

Proof. By triangle inequality, we have

$$\left\|\nabla_g u - T_h^{-1} G_h u_h\right\|_{0,\mathcal{M}} \leq \left\|\nabla_g u - T_h^{-1} G_h (T_h u)\right\|_{0,\mathcal{M}} + \left\|T_h^{-1} G_h (T_h u - u_h)\right\|_{0,\mathcal{M}}.$$

We apply Lemma 5.1 with the embedding of $L^\infty(\mathcal{M}) \hookrightarrow L^2(\mathcal{M})$ for compact manifold to the first term on the right hand side, and apply [30, Theorem 3.5] and the boundedness of both the operators T_h^{-1} and G_h to the second term. Then we get the conclusion. ∎

Due to Lemma 3.1, we have the following result immediately.

COROLLARY 5.3. *Let the same assumptions as Theorem 5.2 hold, then*

$$\|T_h \nabla_g u - G_h u_h\|_{0,\mathcal{M}_h} \lesssim h^{1+\min\{1,\sigma\}} \|u\|_{3,\mathcal{M}} + \|u\|_{2,\infty,\mathcal{M}}) + h^2 \|f\|_{0,\mathcal{M}}. \tag{5.5}$$

6. Recovery-based a posteriori error estimator. The gradient recovery operator G_h naturally provides an *a posteriori* error estimator. We define a local *a posteriori* error estimator on each triangular element $\mathcal{T}_{h,j}$ as

$$\eta_{h,\mathcal{T}_{h,j}} = \|(G_h u_h - T_h \nabla_g T_h^{-1} u_h)\|_{0,\mathcal{T}_{h,j}}, \tag{6.1}$$

and the corresponding global error estimator as

$$\eta_h = \left(\sum_{j \in J_h} \eta_{h,\mathcal{T}_{h,j}}^2 \right)^{1/2}. \tag{6.2}$$

With the previous superconvergence result, we are ready to prove the asymptotic exactness of error estimators based on the recovery operator G_h.

THEOREM 6.1. *Assume the same conditions in Theorem 5.2 and let u_h be the IFE solution of discrete variational problem* (5.2)*. Further assume that there is a constant $C(u) > 0$ such that*

$$\left\|\nabla_g (u - T_h^{-1} u_h)\right\|_{0,\mathcal{M}} \geq C(u) h. \tag{6.3}$$

Then it holds that

$$\left| \frac{\eta_h}{\left\|T_h \nabla_g (u - T_h^{-1} u_h)\right\|_{0,\mathcal{M}_h}} - 1 \right| \lesssim h^\rho. \tag{6.4}$$

Proof. It follows from Theorem 5.2, (6.3), and the triangle inequality. ∎

REMARK 6.1. *Theorem 6.1 implies that* (6.1) *(or* (6.2)*) is an asymptotically exact a posteriori error estimator for surface finite element methods.*

7. Numerical Results. In this section, we present numerical examples to demonstrate the superconvergence property of the proposed gradient recovery operators and make a comparison with existing gradient recovery operators. The first example is to show the superconvergence results of the proposed gradient recovery operators even though the element patch is not $\mathcal{O}(h^2)$-symmetric. The second one is to compare the results on a more complicated surface, and to demonstrate the superiority of the PPPR method for surfaces with high curvature. The last two are to show the asymptotic exactness of the recovery-based *a posterior* error estimator introduced in Section 6. Some of our numerical tests are performed based on MATLAB package *i*FEM [6]. Except for the first example, the initial meshes for the other three examples are generated using the three-dimensional surface mesh generation module of the Computational Geometry Algorithms Library [29].

Let G_h^{SA}, G_h^{WA}, and G_h^{ZZ} be recovery operators by simple averaging, weighted averaging, and Zienkiewicz-Zhu schemes on tangent planes [30], respectively. Note we use the exact normal vectors for G_h^{ZZ} in the numerical examples. We denote $G_{1,h}$, $G_{2,h}$, and $G_{1,h}^a$ to be the recovery operators given by Algorithm 1, Algorithm 2 and Algorithm 1 with approximations of normal vectors, respectively. The approximating normal vectors are computed by weighted averaging for the tests with $G_{1,h}^a$ in our examples, which are also used to implement Algorithm 2 to construct the local parametric domains Ω_i. Another remark is that we use the function value preserving skill in the PPPR $G_{2,h}$, but not for $G_{1,h}$. For the reason of making comparison, we define:

$$
\begin{aligned}
&De = \|T_h\nabla_g u - \nabla_{g_h} u_h\|_{0,\mathcal{M}_h}, && De^I = \|\nabla_{g_h} u_I - \nabla_{g_h} u_h\|_{0,\mathcal{M}_h},\\
&De^{r_1} = \|T_h\nabla_g u - G_{1,h} u_h\|_{0,\mathcal{M}_h}, && De^{r_2} = \|T_h\nabla_g u - G_{2,h} u_h\|_{0,\mathcal{M}_h},\\
&De^{r_3} = \left\|T_h\nabla_g u - G_{1,h}^a u_h\right\|_{0,\mathcal{M}_h}, && De^{SA} = \left\|T_h\nabla_g u - G_h^{SA} u_h\right\|_{0,\mathcal{M}_h},\\
&De^{WA} = \left\|T_h\nabla_g u - G_h^{WA} u_h\right\|_{0,\mathcal{M}_h}, && De^{ZZ} = \left\|T_h\nabla_g u - G_h^{ZZ} u_h\right\|_{0,\mathcal{M}_h}.
\end{aligned}
$$

where u_h is the finite element solution, u is the analytical solution and u^I is the interpolation of u at the nodal points x_i.

In Numerical Example 2, we shall compare the discrete maximal error of the above six discrete gradient recovery methods. For that reason, we introduce the following notations

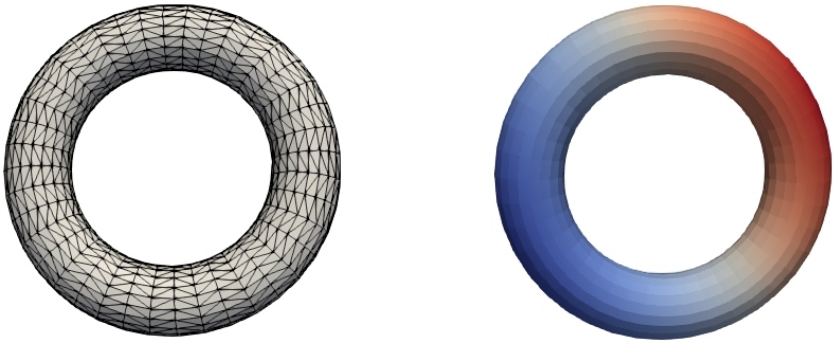
$$
\begin{aligned}
&De_0^{r_1} = \|T_h\nabla_g u - G_{1,h} u_h\|_{0,\infty,\mathcal{M}_h}, && De_0^{r_2} = \|T_h\nabla_g u - G_{2,h} u_h\|_{0,\infty,\mathcal{M}_h},\\
&De_0^{r_3} = \left\|T_h\nabla_g u - G_{1,h}^a u_h\right\|_{0,\infty,\mathcal{M}_h}, && De_0^{SA} = \left\|T_h\nabla_g u - G_h^{SA} u_h\right\|_{0,\infty,\mathcal{M}_h},\\
&De_0^{WA} = \left\|T_h\nabla_g u - G_h^{WA} u_h\right\|_{0,\infty,\mathcal{M}_h}, && De_0^{ZZ} = \left\|T_h\nabla_g u - G_h^{ZZ} u_h\right\|_{0,\infty,\mathcal{M}_h}.
\end{aligned}
$$

where $\|\cdot\|_{0,\infty,\mathcal{M}_h}$ means the maximum absolute value at all vertices.

In the following tables, all convergence rates are listed with respect to the degree of freedom(DOF). Noticing Dof $\approx h^{-2}$, the corresponding convergence rates with respect to the mesh size h are double of what we present in the tables.

7.1. Numerical Example 1. Our first example is to consider Laplace-Beltrami equation on a torus surface. The right hand function f is chosen to fit the exact solution $u(x, y, z) = x - y$. The signed distance function of torus surface is

$$
d(x, y, z) = \sqrt{(4 - \sqrt{x^2 + y^2})^2 + z^2} - 1. \tag{7.1}
$$



(a) (b)

Fig. 7.1: Numerical Solution on Torus Surface: (a) Mesh; (b) Solution.

To construct a series meshes on torus without $\mathcal{O}(h^2)$ symmetric property of their element patches, we firstly make a series of uniform meshes of Chevron pattern and map the mesh onto the torus. Figure 7.1 plots the uniform mesh with 800 Dof and the corresponding finite element solution.

Table 7.1 lists the numerical results. As expected, H^1 error of finite element solution is of $\mathcal{O}(h)$. The generated uniform meshes satisfy the $\mathcal{O}(h^{2\sigma})$ condition. For that reason, $\mathcal{O}(h^2)$ supercloseness for De^I is observed. Concerning the convergence of recovered gradients, both the recovered gradient by PPR with exact normal field and by the PPPR superconverges have a superconvergence rate of order $\mathcal{O}(h^2)$; while the recovered gradient using PPR with recovered normal field and the other three methods in [30] only converge at the optimal rate $\mathcal{O}(h)$.

Table 7.1: Numerical Results for equation (5.1) on torus surface.

Dof	De	order	De^I	order	De^{r_1}	Order	De^{r_2}	order
200	2.52e+00	–	9.43e-01	–	1.50e+00	–	1.59e+00	–
800	1.26e+00	0.50	2.65e-01	0.92	4.12e-01	0.93	4.37e-01	0.93
3200	6.29e-01	0.50	6.92e-02	0.97	1.06e-01	0.98	1.13e-01	0.98
12800	3.14e-01	0.50	1.75e-02	0.99	2.67e-02	0.99	2.84e-02	0.99
51200	1.57e-01	0.50	4.40e-03	1.00	6.70e-03	1.00	7.12e-03	1.00
204800	7.86e-02	0.50	1.10e-03	1.00	1.67e-03	1.00	1.78e-03	1.00
819200	3.93e-02	0.50	2.75e-04	1.00	4.19e-04	1.00	4.45e-04	1.00
3276800	1.97e-02	0.50	6.88e-05	1.00	1.05e-04	1.00	1.11e-04	1.00
Dof	De^{r_3}	order	De^{SA}	order	De^{WA}	Order	De^{ZZ}	order
200	1.52e+00	–	2.27e+00	–	2.28e+00	–	2.27e+00	–
800	4.74e-01	0.84	7.22e-01	0.83	7.25e-01	0.83	6.91e-01	0.86
3200	1.68e-01	0.75	2.48e-01	0.77	2.49e-01	0.77	2.19e-01	0.83
12800	7.18e-02	0.61	1.03e-01	0.63	1.03e-01	0.63	8.39e-02	0.69
51200	3.42e-02	0.54	4.86e-02	0.54	4.86e-02	0.54	3.80e-02	0.57
204800	1.69e-02	0.51	2.39e-02	0.51	2.39e-02	0.51	1.84e-02	0.52
819200	8.40e-03	0.50	1.19e-02	0.50	1.19e-02	0.50	9.16e-03	0.51
3276800	4.20e-03	0.50	5.94e-03	0.50	5.94e-03	0.50	4.57e-03	0.50

7.2. Numerical Example 2. In this example, we take an emblematical surface [18] which contains high curvature features. It can be represented as the zero level of the following level set function

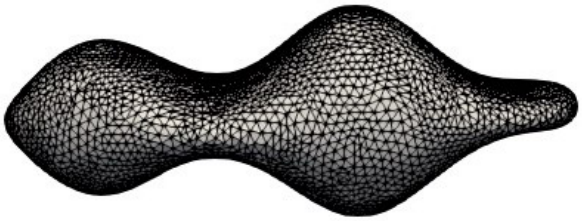
$$\Phi(x) = \frac{1}{4}x_1^2 + x_2^2 + \frac{4x_3^2}{(1+\frac{1}{2}\sin(\pi x_1))^2} - 1.$$

We consider the Laplace-Beltrami equation (2.5) with exact solution $u = x_1x_2$. The right-hand side function f is computed from u.

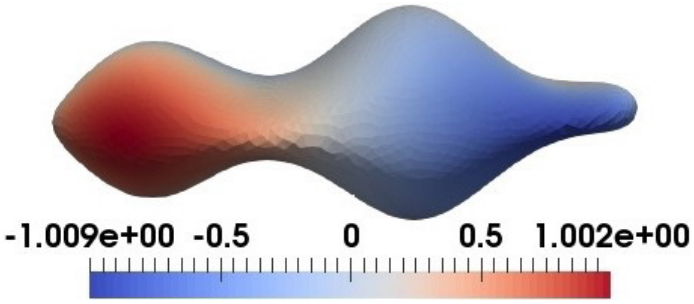
Figure 7.2b shows the finite element solution u_h on Delaunay mesh, see 7.2a, with 4606 Dofs. The numerical results is reported in Table 7.2. From the table, we clearly see that De converges at the optimal rate $\mathcal{O}(h)$ and De^I converges at a superconvergent rate $\mathcal{O}(h^2)$. As demonstrated in [7], some regions of the surface are with significant high curvature. Due to the existence of these area, only sub-superconvergence rate of order $\mathcal{O}(h^{1.8})$ is observed for PPR with approximated normal field and the other three methods in [30]. In contrast, the $\mathcal{O}(h^2)$ superconvergence rate can be observed in the PPR with exact normal field and in the PPPR method. In order to look more clearly into the relations between the recovery accuracy and the high curvature of a surface, we add another set of comparison in this example. In our numerical tests, we observed that the maximal recovery errors always happened at the area of the meshes generated from highest curvature surface regions. We plot a case example of the distribution of error function $|G_{h,r_2}u_h - \nabla_g u|$ in Figure 7.2c. Table 7.3 reports the maximal discrete errors of all the above six gradient recovery methods, in which PPPR method is the only one to achieve the superconvergence rate of $\mathcal{O}(h^2)$ asymptotically in the discrete maximal norm. This gives the evidence to our statement in Remark 4.3 that PPPR is relatively *curvature stable* in comparing with the other methods. At that point, we can draw the conclusion that PPPR is the best one with respect to arbitrary meshes and meshes generated by high curvature surfaces. Thus, in the following two examples, we shall only consider the PPPR method.

Table 7.2: Numerical Results for equation (5.1) on a general surface

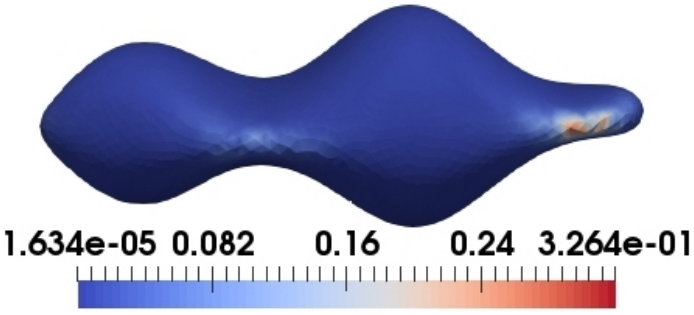
Dof	De	order	De^I	order	De^{r_1}	Order	De^{r_2}	order
1153	5.46e-01	–	2.78e-01	–	4.77e-01	–	3.34e-01	–
4606	2.85e-01	0.47	1.18e-01	0.62	2.01e-01	0.62	1.29e-01	0.69
18418	1.40e-01	0.51	3.45e-02	0.89	6.58e-02	0.81	4.38e-02	0.78
73666	6.97e-02	0.50	9.86e-03	0.90	1.97e-02	0.87	1.28e-02	0.89
294658	3.48e-02	0.50	2.58e-03	0.97	5.33e-03	0.95	3.40e-03	0.96
1178626	1.74e-02	0.50	6.57e-04	0.99	1.37e-03	0.98	8.68e-04	0.99
4714498	8.70e-03	0.50	1.66e-04	0.99	3.46e-04	0.99	2.18e-04	1.00
Dof	De^{r_3}	order	De^{SA}	order	De^{WA}	Order	De^{ZZ}	order
1153	4.71e-01	–	4.83e-01	–	4.86e-01	–	4.95e-01	–
4606	1.98e-01	0.62	2.26e-01	0.55	2.30e-01	0.54	2.18e-01	0.59
18418	6.63e-02	0.79	8.30e-02	0.72	8.59e-02	0.71	7.45e-02	0.78
73666	2.06e-02	0.84	2.69e-02	0.81	2.82e-02	0.80	2.33e-02	0.84
294658	5.87e-03	0.91	7.72e-03	0.90	8.27e-03	0.89	6.60e-03	0.91
1178626	1.64e-03	0.92	2.14e-03	0.93	2.36e-03	0.90	1.83e-03	0.93
4714498	4.70e-04	0.90	6.04e-04	0.91	6.97e-04	0.88	5.22e-04	0.90



(a) Mesh



(b) Solution



(c) Error distribution of recovered gradient

Fig. 7.2: Numerical Solution on general surface.

7.3. Numerical Example 3. In the example, we consider a benchmark problem for adaptive finite element method for Laplace-Beltrami equation on the sphere [7, 11, 12]. We choose the right hand side function f such that he exact solution in spherical coordinate is given by

$$u = \sin^{\lambda}(\theta)\sin(\psi).$$

In case of $\lambda < 1$, it easy to see that the solution u has two singularity points at north and south poles and the solution u is barely in $H^1(\mathcal{M})$. In fact, $u \in H^{1+\lambda}(\mathcal{M})$.

To obtain optimal convergence rate, adaptive finite element method (AFEM) is used. Different from existing methods in the literature, recovery-based *a posteriori* error estimator is adopted. We start with the initial mesh given as in Fig 7.3a. The mesh is adaptively refined using the Dorfler [15] marking strategy with parameter

Table 7.3: Comparison of discrete maximal norms of gradient recovery methods on a general surface

Dof	$De_0^{r_1}$	order	$De_0^{r_2}$	order	$De_0^{r_3}$	Order
1153	9.70e-01	–	7.93e-01	–	8.73e-01	–
4606	5.43e-01	0.42	3.26e-01	0.64	4.77e-01	0.44
18418	1.92e-01	0.75	1.09e-01	0.79	2.22e-01	0.55
73666	8.57e-02	0.58	5.18e-02	0.54	9.16e-02	0.64
294658	2.50e-02	0.89	1.40e-02	0.94	3.51e-02	0.69
1178626	7.80e-03	0.84	3.59e-03	0.98	1.59e-02	0.57
4714498	3.56e-03	0.57	9.03e-04	0.99	7.57e-03	0.53
Dof	De_0^{SA}	order	De_0^{WA}	order	De_0^{ZZ}	Order
1153	7.16e-01	–	7.09e-01	–	8.13e-01	–
4606	5.09e-01	0.25	5.36e-01	0.20	5.83e-01	0.24
18418	2.73e-01	0.45	3.05e-01	0.41	2.65e-01	0.57
73666	1.42e-01	0.47	1.47e-01	0.53	1.11e-01	0.63
294658	5.66e-02	0.66	6.08e-02	0.64	3.67e-02	0.80
1178626	2.39e-02	0.62	2.75e-02	0.57	1.62e-02	0.59
4714498	1.12e-02	0.55	1.31e-02	0.54	7.63e-03	0.54

equal to 0.3. Fig 7.3b plots the mesh after the 18 adaptive refinement steps. Clearly, the mesh successfully resolves the singularities. The numerical errors are displayed in Fig 7.4a. As expected, optimal convergence rate for H^1 error can be observed. In addition, we observe that the recovery is superconvergent to the exact gradient at a rate of $\mathcal{O}(h^2)$.

To test the performance of our new recovery-based *a posterior* error estimator for Laplace-Beltrami problem, the effectivity index κ is used to measure the quality of an error estimator [1, 3], which is defined by the ratio between the estimated error and the true error

$$\kappa = \frac{\left\|G_h u_h - T_h \nabla_g T_h^{-1} u_h\right\|_{0,\mathcal{M}_h}}{\left\|T_h \nabla_g (u - T_h^{-1} u_h)\right\|_{0,\mathcal{M}_h}} \tag{7.2}$$

The effectivity index is plotted in Fig 7.4b . We see that κ converges asymptotically to 1 which indicates the posteriori error estimator (6.1)or (6.2) is asymptotically exact.

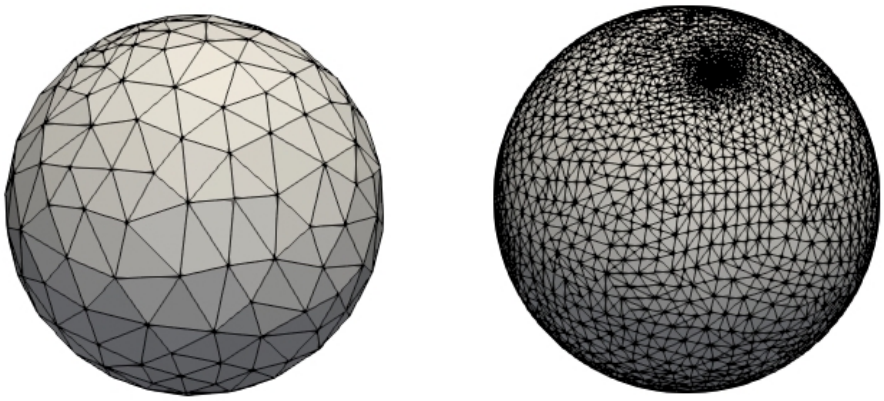
7.4. Numerical Example 4. In this example, we consider the following Laplace-Beltrami type equation on Dziuk surface as in [8]:

$$-\Delta_g + u = f, \quad \text{on } \Gamma,$$

where $\Gamma = \left\{x \in \mathbb{R}^3 : (x_1 - x_3^2)^2 + x_2^2 + x_3^2 = 1\right\}$. f is chosen to fit the exact solution

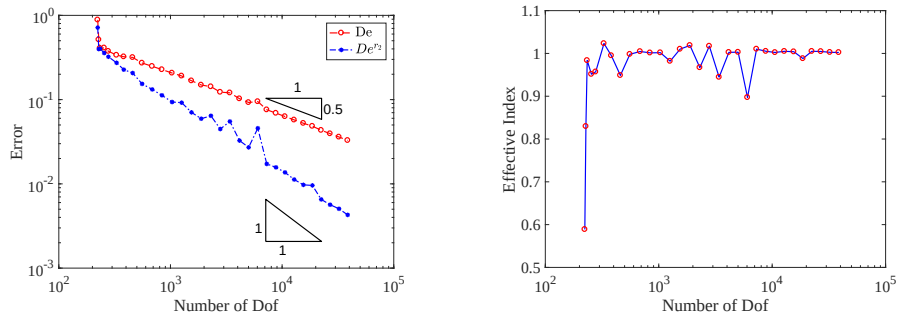
$$u(x, y, z) = e^{\frac{1}{1.85-(x-0.2)^2}} \sin(y).$$

Note that the solution has an exponential peak. To track this phenomena, we adopt AFEM with an initial mesh graphed in Fig 7.5a. Fig 7.5b shows the adaptive refined mesh. We would like to point out that the mesh is refined not only around the exponential peak but also at the high curvature areas.



(a) (b)

Fig. 7.3: Meshes for Example 3: (a) Initial mesh; (b) Adaptive refined mesh.

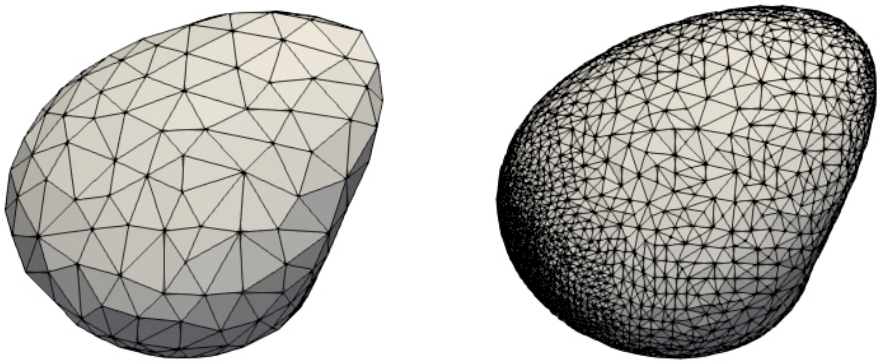


(a) (b)

Fig. 7.4: Numerical Result for Example 3: (a) Errors; (b) Effective index.

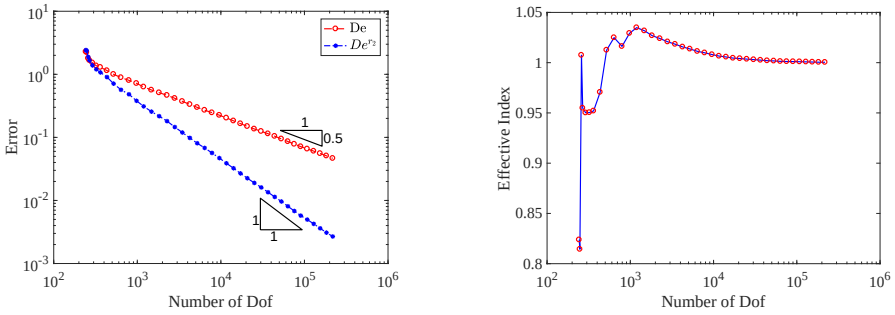
Fig 7.6a displays the numerical errors. It demonstrates the optimal convergence rate in H^1 norm and a superconvergence rate for the recovered gradient. The effective index is showed in Fig 7.6b, which converges to 1 quickly after the first few iterations. Again, it indicates the error estimator (6.1) (or (6.2)) is asymptotically exact.

8. Conclusion. In this paper, we have proposed a curvature stable gradient recovery method (PPPR) for data defined on manifolds. In comparing with existing methods for data on surfaces in the literature, cf. [16, 30], the proposed method has several improvements: The first highlight is that it does not require the exact surfaces, which makes it a realistic and robust method for practical problems; Second, it does not need the element patch to be $\mathcal{O}(h^2)$ symmetric to achieve superconvergence. Third, it is the most curvature stable methods in comparing with the existing methods. Aside from that, we have evolved the traditional PPR method (for planar problems) to function value preserving at the mean time. By testing with some benchmark examples, it is quite evident that the proposed method numerically performs better than the methods in the state of the art. We have also shown the capability of the



(a) (b)

Fig. 7.5: Meshes for Example 4: (a) Initial mesh; (b) Adaptive refined mesh.



(a) (b)

Fig. 7.6: Numerical Result for Example 4: (a) Errors; (b) Effective index.

recovery operator for constructing *a posterior* error estimator. Even though we only develop the methods for linear finite element methods on triangulated meshes, the idea should be applicable to higher order FEM on more accurate approximations of surfaces, e.g. piece-wise polynomial surfaces, B-splines or NURBUS. We leave this as a potential work for future.

Gradient recovery has other applications, like enhancing eigenvalues, pre-processing data in image science, simplifying higher order discretization of PDEs, or even designing new numerical methods for higher order PDEs, and so on. It would also be interesting to further investigate the full usage of the PPPR method for problems with solutions defined on a manifold domain.

Ackowledgement. The authors thank Dr. Pravin Madhavan, Dr. Bjorn Stinner and Dr. Andreas Dedner for their kind help and discussions on a numerical example. GD acknowledges support from the Austrian Science Fund (FWF): Geometry and Simulation, project S11704. HG acknowledges support from the Center for Scientific Computing from the CNSI, MRL: an NSF MRSEC (DMR-1121053) and NSF CNS-

0960316 for providing computing resources.

Appendix A. Proof of Lemma 2.1.

Proof. In general, there are infinitely many isomorphic parametrizations for a given patch $S \subset \mathcal{M}$. Let us pick arbitrarily two of them, which are denoted by

$$\mathbf{r} : \Omega \to S \quad \text{and } \mathbf{s} : \Omega_s \to S \,,$$

respectively, where Ω and Ω_s are planar parameter domains, then there exist

$$\mathbf{t} : \Omega \to \Omega_s$$

to be a bijective, differentiable mapping, such that $\mathbf{r} = \mathbf{s} \circ \mathbf{t}$. That means for an arbitrary but fixed position $x \in S$, we have $\xi \in \Omega$ and $\mathbf{t}(\xi) = \zeta$, such that

$$x = \mathbf{s}(\zeta) = \mathbf{s}(\mathbf{t}(\xi)) = \mathbf{r}(\xi).$$

Then we have

$$\partial \mathbf{r}(\xi) = \partial \mathbf{s}(\mathbf{t}(\xi)) \partial \mathbf{t}(\xi),$$

and consequently, for every function $v : S \to \mathbb{R}$,

$$v \circ \mathbf{r} : \Omega \to \mathbb{R} \quad \text{and} \quad v \circ \mathbf{s} : \Omega_s \to \mathbb{R},$$

we have

$$\nabla_g v(\mathbf{r}(\xi))\ \partial \mathbf{r}(\xi) = \nabla (v \circ \mathbf{r})(\xi) \text{ and } \nabla_g v(\mathbf{s}(\zeta))\ \partial \mathbf{s}(\zeta) = \nabla (v \circ \mathbf{s})(\zeta). \tag{A.1}$$

Using chain rule on both sides of the former equation of (A.1), then we get

$$\nabla_g v(\mathbf{s}(\zeta))\ \partial \mathbf{s}(\mathbf{t}(\xi)) \partial \mathbf{t}(\xi) = \partial (v \circ \mathbf{s}(\mathbf{t}(\xi))) \partial \mathbf{t}(\xi) \Rightarrow \nabla_g v(\mathbf{s}(\zeta))\ \partial \mathbf{s}(\mathbf{t}(\xi)) = \partial (v \circ \mathbf{s}(\mathbf{t}(\xi))),$$

which gives the latter equation in (A.1) since $\partial \mathbf{t}(\xi)$ is non-degenerate. Using the same process but consider $\mathbf{t}^{-1} : \Omega_s \to \Omega$, we can show the reverse implication. Thus, we have shown that any two arbitrary parameterizations $\mathbf{r}$ and $\mathbf{s}$ lead to the same gradient values at same positions. ∎

REFERENCES

[1] M. Ainsworth and J. T. Oden, *A posteriori error estimation in finite element analysis*, Pure and Applied Mathematics (New York), Wiley-Interscience [John Wiley & Sons], New York, 2000.

[2] T. Aubin, *Best constants in the Sobolev imbedding theorem: the Yamabe problem*, in Seminar on Differential Geometry, vol. 102 of Ann. of Math. Stud., Princeton Univ. Press, Princeton, N.J., 1982, pp. 173–184.

[3] I. Babuška and T. Strouboulis, *The finite element method and its reliability*, Numerical Mathematics and Scientific Computation, The Clarendon Press, Oxford University Press, New York, 2001.

[4] R. E. Bank and J. Xu, *Asymptotically exact a posteriori error estimators. I. Grids with superconvergence*, SIAM J. Numer. Anal., 41 (2003), pp. 2294–2312 (electronic).

[5] F. Camacho and A. Demlow, *L_2 and pointwise a posteriori error estimates for FEM for elliptic PDEs on surfaces*, IMA J. Numer. Anal., 35 (2015), pp. 1199–1227.

[6] L. Chen, *iFEM: an innovative finite element methods package in MATLAB*, University of California at Irvine, (2009).

[7] A. Y. Chernyshenko and M. A. Olshanskii, *An adaptive octree finite element method for PDEs posed on surfaces*, Comput. Methods Appl. Mech. Engrg., 291 (2015), pp. 146–172.

[8] A. Dedner and P. Madhavan, *Adaptive discontinuous Galerkin methods on surfaces*, Numer. Math., 132 (2016), pp. 369–398.
[9] A. Dedner, P. Madhavan, and B. Stinner, *Analysis of the discontinuous Galerkin method for elliptic problems on surfaces*, IMA J. Numer. Anal., 33 (2013), pp. 952–973.
[10] A. Demlow, *Higher-order finite element methods and pointwise error estimates for elliptic problems on surfaces*, SIAM J. Numer. Anal., 47 (2009), pp. 805–827.
[11] A. Demlow and G. Dziuk, *An adaptive finite element method for the Laplace-Beltrami operator on implicitly defined surfaces*, SIAM J. Numer. Anal., 45 (2007), pp. 421–442 (electronic).
[12] A. Demlow and M. A. Olshanskii, *An adaptive surface finite element method based on volume meshes*, SIAM J. Numer. Anal., 50 (2012), pp. 1624–1647.
[13] M. P. do Carmo, *Riemannian geometry*, Mathematics: Theory & Applications, Birkhäuser Boston, Inc., Boston, MA, 1992. Translated from the second Portuguese edition by Francis Flaherty.
[14] G. Dong, B. Jüttler, O. Scherzer, and T. Takacs, *Convergence of tikhonov regularization for solving ill–posed operator equations with solutions defined on surfaces*, Inverse Probl. Imaging, 11 (2017), pp. 221 – 246.
[15] W. Dörfler, *A convergent adaptive algorithm for Poisson's equation*, SIAM J. Numer. Anal., 33 (1996), pp. 1106–1124.
[16] Q. Du and L. Ju, *Finite volume methods on spheres and spherical centroidal Voronoi meshes*, SIAM J. Numer. Anal., 43 (2005), pp. 1673–1692 (electronic).
[17] G. Dziuk, *Finite elements for the Beltrami operator on arbitrary surfaces*, in Partial differential equations and calculus of variations, vol. 1357 of Lecture Notes in Math., Springer, Berlin, 1988, pp. 142–155.
[18] G. Dziuk and C. M. Elliott, *Finite element methods for surface PDEs*, Acta Numer., 22 (2013), pp. 289–396.
[19] J. Grande and A. Reusken, *A higher order finite element method for partial differential equations on surfaces*, SIAM J. Numer. Anal., 54 (2016), pp. 388–414.
[20] H. Guo and X. Yang, *Polynomial preserving recovery for high frequency wave propagation*, J. Sci. Comput., (2016), pp. 1–21.
[21] H. Guo and Z. Zhang, *Gradient recovery for the Crouzeix-Raviart element*, J. Sci. Comput., 64 (2015), pp. 456–476.
[22] H. Guo, Z. Zhang, and R. Zhao, *Hessian recovery for finite element methods*, Math. Comp., (2016), pp. 1–22.
[23] E. Hebey, *Nonlinear analysis on manifolds: Sobolev spaces and inequalities*, vol. 5 of Courant Lecture Notes in Mathematics, New York University, Courant Institute of Mathematical Sciences, New York; American Mathematical Society, Providence, RI, 1999.
[24] A. M. Lakhany, I. Marek, and J. R. Whiteman, *Superconvergence results on mildly structured triangulations*, Comput. Methods Appl. Mech. Engrg., 189 (2000), pp. 1–75.
[25] J. M. Lee, *Riemannian manifolds*, vol. 176 of Graduate Texts in Mathematics, Springer-Verlag, New York, 1997. An introduction to curvature.
[26] M. A. Olshanskii and A. Reusken, *A finite element method for surface PDEs: matrix properties*, Numer. Math., 114 (2010), pp. 491–520.
[27] M. A. Olshanskii, A. Reusken, and J. Grande, *A finite element method for elliptic equations on surfaces*, SIAM J. Numer. Anal., 47 (2009), pp. 3339–3358.
[28] M. A. Olshanskii and D. Safin, *A narrow-band unfitted finite element method for elliptic PDEs posed on surfaces*, Math. Comp., 85 (2016), pp. 1549–1570.
[29] L. Rineau and M. Yvinec, *3D surface mesh generation*, in CGAL User and Reference Manual, CGAL Editorial Board, 4.9 ed., 2016.
[30] H. Wei, L. Chen, and Y. Huang, *Superconvergence and gradient recovery of linear finite elements for the Laplace-Beltrami operator on general surfaces*, SIAM J. Numer. Anal., 48 (2010), pp. 1920–1943.
[31] J. Xu and Z. Zhang, *Analysis of recovery type a posteriori error estimators for mildly structured grids*, Math. Comp., 73 (2004), pp. 1139–1152 (electronic).
[32] Z. Zhang and A. Naga, *A new finite element gradient recovery method: superconvergence property*, SIAM J. Sci. Comput., 26 (2005), pp. 1192–1213 (electronic).
[33] O. C. Zienkiewicz and J. Z. Zhu, *The superconvergent patch recovery and a posteriori error estimates. I. The recovery technique*, Internat. J. Numer. Methods Engrg., 33 (1992), pp. 1331–1364.
[34] ———, *The superconvergent patch recovery and a posteriori error estimates. II. Error estimates and adaptivity*, Internat. J. Numer. Methods Engrg., 33 (1992), pp. 1365–1382.