SIAM J. SCI. COMPUT.
Vol. 42, No. 3, pp. A1885–A1912

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 require the tangent spaces of the exact manifolds which have been assumed for some significant gradient recovery methods in the literature. Another advantage of PPPR is that superconvergence is guaranteed without the symmetric condition which is required in the existing techniques. As an application, we show its capability of constructing an asymptotically exact a posteriori error estimator. Several numerical examples on two-dimensional surfaces are presented to support the theoretical results, and comparisons with existing methods are documented, showing that PPPR outperforms the other methods, in particular in the case of high curvature surfaces as well as mildly structured meshes.

Key words. gradient recovery, manifolds, superconvergence, parametric polynomial preserving, a posteriori error estimator

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

DOI. 10.1137/18M1191336

1. Introduction.

Numerical methods for approximating variational problems or partial differential equations (PDEs) with solutions defined on surfaces or manifolds have been of growing interest over the last decades. Finite element methods, as one of the most important methods for numerically solving PDEs, are well established for those problems. A starting point can be traced back to [21], which is the first work to investigate a finite element method for solving elliptic PDEs on surfaces. Since then, there have been many extensions in both numerical analysis and practical algorithms; see [13, 14, 15, 22, 38, 39, 40] and the references therein. In the literature, most works focus on the a priori error analysis of various surface finite element methods. Only a few works, to the best of our knowledge, take into account the a posteriori error analysis and superconvergence of finite element methods in a surface setting; see [6, 10, 12, 15, 16, 20, 42]. A recently proposed approach in [23] 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 vital role. Gradient recovery techniques, which are important in *postprocessing* numerical solutions to improve the accuracy of gradient approximations, have been widely studied and have found many applications in numerical analysis. For planar problems, the study of gradient recovery methods has reached a certain maturity stage, and there is a massive literature; see [1, 4, 24, 25, 31, 43, 45, 46, 47], to name a few. We point out some significant

*Submitted to the journal's Methods and Algorithms for Scientific Computing section May 30, 2018; accepted for publication (in revised form) March 25, 2020; published electronically June 30, 2020.

https://doi.org/10.1137/18M1191336

Funding: The work of the first author was partially supported by the German Research Foundation under Germany's Excellence Strategy: The Berlin Mathematics Research Center MATH+ (EXC-2046/1, project ID: 390685689). The work of the second author was partially supported by the Andrew Sisson Fund of the University of Melbourne.

†Institute for Mathematics, Humboldt University of Berlin, 10099 Berlin, Germany (guozhi.dong@hu-berlin.de) and Weierstrass Institute for Applied Analysis and Stochastics, 10117 Berlin, Germany (dong@wias-berlin.de).

‡Corresponding author. School of Mathematics and Statistics, The University of Melbourne, Parkville, VIC 3010, Australia (hailong.guo@unimelb.edu.au).

methods among them, like the classical Zienkiewicz–Zhu (ZZ) superconvergent patch recovery [46] and a later method called polynomial preserving recovery (PPR) [45]. These two approaches work under different philosophies methodologically. The former method first locates positions of superconvergent points in the neighborhood of each nodal point and then recovers the gradients themselves at the nodal point by fitting the selected neighbored superconvergent points to achieve higher order approximation accuracy; the latter method first fits a polynomial in the least-squares sense at each nodal point and then takes the gradient of the fitted polynomial to have the recovered gradient. Both methods can produce comparable superconvergence results, but the ZZ-scheme requires stronger conditions on the discretized meshes than the PPR method.

Gradient recovery methods for data defined on curved spaces have only recently been investigated. In [42], several gradient recovery methods have been extended to a general surface setting for linear finite element solutions which are defined on polyhedrons by triangulation. The surface in [42] is considered to be a zero level set of a smooth function embedded in an ambient space, and the gradient of functions on the surface is represented using the ambient gradient with tangential projection. 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 require knowledge of the exact surface: the vertices of the triangles are assumed to be located on the exact surface, and the exact normal vectors are given. However, this information is usually not available in reality, where we have only interpolation or approximations of surfaces—for instance, polyhedrons, splines, or polynomial surfaces. How to deal with gradient recovery in such cases and prove its superconvergence is an open question in [42]. On the other hand, the generalized ZZ-scheme for surface elements gives the most competitive results in [42], but its superconvergence, including several other methods, is proven with a condition that the mesh is $\mathcal{O}(h^2)$-symmetric. In the planar cases, this restrictive condition, however, is not necessary for the PPR method.

This motivates us to consider a generalization of the PPR method to the manifolds setting. A follow-up question would be "What are polynomials in the curved manifold domain?" Using the idea from the literature, e.g., [20], one could consider polynomials defined locally on the tangent spaces of the manifolds. Apparently, such a straightforward generalization based on tangent spaces will again fall into the awkward situation of the exact manifolds and their tangent spaces being unknown.

To overcome these difficulties, we go back to the original definition of manifolds. Every local patch of a manifold resembles a planar Euclidean domain; therefore a local parametrization for a patch of the manifold can always be established with respect to a parametric domain Ω that is not necessarily a tangent space. The idea is then to use polynomials to recover the unknown parametrization function of the discretized patch locally on the parametric domain Ω, as well to fit the corresponding local data or finite element solutions on Ω iso-parametrically, from which we are able to recovery the gradient using the intrinsic definition (see formulas (2.1) and (2.2)). Our proposed method is 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 is revealed that the idea of using the parametric domain is particularly useful to *address the issue of unavailable tangent spaces and vertices*, and thus it enables us to answer the open problem in [42]. Even though we only prove the superconvergence with exact vertices in this paper, as can also be observed from our numerical examples, the superconvergence does hold for nonexact vertices of the

triangulation. We can also observe from numerical examples that, compared with the methods proposed in [42], PPPR has much better performance for gradient recovery in the high curvature cases. Moreover, the original PPR method [45] does not preserve the function values at the nodal points in its prerecovery step. In this paper, we take care of this issue, so that the PPPR is able to not only preserve the *parametric polynomial* but also preserve the *surface sampling points* and the *function values* at the given points simultaneously. This property makes the given data invariant in using the PPPR method.

The rest of the paper is organized as follows: Section 2 gives a preliminary account of relevant differential geometry concepts and an example PDE problem. Section 3 introduces discretized function spaces and collects some geometric notations frequently used in this paper. Section 4 presents the new algorithms, particularly the proposed PPPR method. Section 5 exhibits the numerical analysis and some relevant properties of the PPPR method. Section 6 shows the recovery-based a posteriori estimator by using the PPPR operator. In section 7, we present numerical results verifying the theoretical analysis and make comparisons with existing methods. Finally, some conclusions are drawn in section 8. We defer the proof of a basic lemma until Appendix A.

2. Background. We only show some basic geometric 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 [17, 32]. In this paper, we shall consider $(\mathcal{M}, g)$ as an oriented, connected, C^3 smooth, regular, and compact Riemannian manifold without boundary, where g denotes the Riemann metric tensor. The idea we are going to work on should have 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 recall 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

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

where $\mathbf{v} = \gamma(\sigma)'|_{\sigma=0}$ is a tangential vector field.

Then, the gradient is an operator such that

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

where $T_x\mathcal{M}$ is the tangent space of $\mathcal{M}$ at x. We can think of the gradient as a tangent vector field on the manifold $\mathcal{M}$. Using 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 entry of the inverse of the metric tensor g, and ∂_i denotes the tangential basis. Fix $\mathbf{r} : \Omega \to S \subset \mathcal{M}$ to be a local geometric mapping; then we can rewrite (2.1) into a matrix form with this concrete local parametrization. That is,

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

In (2.2), $\bar{u} = u \circ \mathbf{r}$ is the pull-back of function u to the local planar parametric domain Ω, ∇ denotes the gradient operator on the planar domain Ω, $\partial\mathbf{r}$ is the Jacobian of $\mathbf{r}$, and

$$g \circ \mathbf{r} = \partial\mathbf{r}(\partial\mathbf{r})^{\top}.$$

Remark 2.1. The parametrization map $\mathbf{r}$ is not specified here. Note that $\mathbf{r}$ is nonunique. Typically it can be constructed as a function graph on some parametric domain, which will be used in our later algorithms. 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 generalized inverse of $\partial\mathbf{r}$. See [18, Appendix] for a detailed explanation.

We consider $\mathcal{M}$ a regular manifold in the paper. "Regular" means that the Jacobian of the parametrization $\partial\mathbf{r}$ and its inverse $(\partial\mathbf{r})^{\dagger}$ are functions with bounded norms in the space $W^{2,\infty}$ on Ω. We have the following lemma whose proof is given in Appendix A.

LEMMA 2.2. *The gradient calculated by using* (2.2) *is invariant for different regular bijective parametrization functions* $\mathbf{r}$.

Let $\omega = dvol$ be the volume form on $\mathcal{M}$ and ∂_j $(j = 1, \ldots, n)$ be the tangential bases, and let $T\mathcal{M} = \bigcup_{x\in\mathcal{M}} T_x\mathcal{M}$ be the tangent bundle which consists of all the tangent planes $T_x\mathcal{M}$ of $\mathcal{M}$. For every tangent vector field $\mathbf{v} : \mathcal{M} \to T\mathcal{M}$, $\mathbf{v} = v^i\partial_i$, we have an $(n-1)$ form defined by the interior product of $\mathbf{v}$ and the volume form ω in the following way:

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

where $k_1, \ldots, k_{n-1}$ are $(n-1)$ indexes with k taking out from $1, \ldots, n$. The divergence of the vector field $\mathbf{v}$ 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 explicitly write the volume form as

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

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

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

This implies that the divergence operator is actually the dual of the gradient operator. With the above preparation, we can now define the Laplace–Beltrami operator, which is denoted by Δ_g in our paper, as the divergence of the gradient, that is,

$$\Delta_g u = \mathrm{div}_g(\nabla_g u) = \frac{1}{\sqrt{|\det g|}}\partial_i(g^{ij}\sqrt{|\det g|}\partial_j u). \tag{2.5}$$

We would like to mention that if the manifold $\mathcal{M}$ is a hypersurface, that is, $\mathcal{M} \subset \mathbb{R}^{n+1}$ and it has codimension 1, then 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. That is,

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

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 ∇_e 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 $\mathcal{M}$. This type of definition has been applied in many references, e.g., [42], which consider problems in an ambient space setting.

With the definition of covariant derivatives on manifolds, many function spaces on Euclidean domains can be studied analogously in the setting of manifolds. Sobolev spaces on manifolds [30] are one of the most investigated spaces which provide a breeding ground to study PDEs. We are interested in numerically approximating PDEs whose solutions are defined on $\mathcal{M}$. Even though our methods are *problem independent*, in this paper the analysis will be mainly conducted for the Laplace–Beltrami operator (2.5) and its generated PDEs. For the purpose of both analysis and applications, we consider the *Laplace–Beltrami equation* as an exemplary problem [21]: For a given f satisfying $\int_{\mathcal{M}} f \, dvol = 0$, find u that solves the equation

$$-\Delta_g u = f \text{ on } \mathcal{M}, \quad \text{with} \int_{\mathcal{M}} u \, dvol = 0, \tag{2.6}$$

where *dvol* denotes the manifold volume measure.

3. Function spaces on discretized manifolds. The discretization of smooth manifolds has been widely studied in many settings, especially in terms of surfaces [22]. A discretized surface, in most cases, is a piecewise polynomial surface. One of the simplest cases is the polygonal approximation to a given smooth surface, especially with triangulations. Finite element methods for triangulated meshes on surfaces were first studied in [21] using the linear element. In [14], a generalization of [21] to high order finite element methods is proposed based on triangulated surfaces. In order to have an optimal convergence rate, it is shown that the geometric approximation error and the function approximation error have to be compatible with each other. In fact, the balance of the geometric approximation error and the function approximation error is also the key point in the development of our recovery algorithm.

For convenience, Table 1 collects notations frequently used in the paper.

Let $\mathcal{M}_h = \bigcup_{j \in J_h} \tau_{h,j}$ be a triangular mesh, and let $h = \max_{j \in J_h} \operatorname{diam}(\tau_{h,j})$ be the maximum diameter. To better present our main idea, we mostly stick to the simplest case, which is the linear finite elements on triangulated surfaces; thus the nodes consist of simply the vertices of $\mathcal{M}_h$, and we denote the set by $\mathcal{N}_h = \{x_i\}_{i \in I_h}$.

In the following, we define transform operators between the function spaces on $\mathcal{M}$ and on $\mathcal{M}_h$. Let $\mathcal{V}(\mathcal{M})$ and $\mathcal{V}(\mathcal{M}_h)$ be some ansatz function spaces. Then we define

$$\begin{aligned} T_h : \mathcal{V}(\mathcal{M}) &\to \mathcal{V}(\mathcal{M}_h), \\ v &\mapsto v \circ P_h, \end{aligned} \tag{3.1}$$

TABLE 1
Notations.

Notation	Remark
$(\mathcal{M}, g)$	a smooth, connected, oriented, and close manifold with metric g
$(\mathcal{M}_h, g_h)$	a triangular approximation of $\mathcal{M}$ with 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	a local domain on the tangent space at a position $x \in \mathcal{M}$
$(P_h)^{\pm 1}$	bijective maps between $\mathcal{M}_h$ and $\mathcal{M}$
$\mathcal{V}(\mathcal{M})/\mathcal{V}_h(\mathcal{M}_h)$	ansatz function spaces for functions on $\mathcal{M}/\ \mathcal{M}_h$
$(T_h)^{\pm 1}$	operators between function spaces on $\mathcal{M}$ and on $\mathcal{M}_h$
h	the diameter of the triangulation mesh in $\mathcal{M}_h$
Ω	a parametric domain for a patch on $\mathcal{M}/\mathcal{M}_h$
ζ	a position variable in the parameter domain Ω
$\mathbf{r}/\mathbf{r}_h$	a local parametrization map from Ω to a patch of $\mathcal{M}/\mathcal{M}_h$
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 $\mathcal{M}$
$\|\cdot\|_{k,p,\mathcal{M}}$	$W^{k,p}$ seminorm 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 $\mathcal{M}_h$
$\mathbb{P}_2(\Omega)$	the 2nd order polynomial space over a planar domain Ω
$a \circ b$	function a composed with function b
$\alpha \lesssim \beta$	denotes the inequality $\alpha \leq C\beta$ where C is a constant
$\mathcal{O}(\sigma)$	denotes the quantity satisfies $\lim_{\sigma\to 0} \frac{\mathcal{O}(\sigma)}{\sigma} = C$ for $\sigma > 0$

and its inverse

$$(T_h)^{-1} : \mathcal{V}(\mathcal{M}_h) \to \mathcal{V}(\mathcal{M}), \qquad v_h \mapsto v_h \circ P_h^{-1}, \tag{3.2}$$

where P_h is a continuous and bijective projection map from every element in $\{\tau_{h,j}\}_{j\in J_h}$ to every element in $\{\tau_j\}_{j\in J_h}$.

We will use the following definition to characterize the approximation quantity of $\mathcal{M}_h$ to $\mathcal{M}$. For the purpose of later analysis, we assume that both T_{x_i} and Ω_i are compact domains corresponding to selected compact patches on $\mathcal{M}_h$ or $\mathcal{M}$.

DEFINITION 3.1. *Let $\mathcal{M}_h = \bigcup_{j\in J_h} \tau_{h,j}$ be a triangular approximation of $\mathcal{M}$. Let $\mathcal{K}_i \subset \mathcal{M}_h$ be the triangle patches associated with the vertex $x_i \in \mathcal{N}_h$. Then there is a curved patch $\mathcal{M}_i \subset \mathcal{M}$. Let $\mathcal{K}_i$ and $\mathcal{M}_i$ be parametrizable by a common domain Ω_i with $\mathbf{r}_{h,i}$ and $\mathbf{r}_i$ being their parametrization functions, respectively. We call $\mathcal{M}_h$ a* regular approximation *of $\mathcal{M}$ if*

$$\lim_{h\to 0} \|\mathbf{r}_{h,i} - \mathbf{r}_i\|_{\infty;\Omega_i} = 0 \quad \textit{for all } i \in I_h, \tag{3.3}$$

if both $|\partial \mathbf{r}_{h,i}|$ and its inverse $|(\partial \mathbf{r}_{h,i})^\dagger|$ are uniformly bounded on Ω_i (edges are ignored) for all $i \in I_h$.

Based on our assumptions on $\mathcal{M}$, this indicates that $\mathbf{r}_i \in W^{3,\infty}(\Omega_i)$ for all $i \in I_h$. In particular, since we consider the vertices of $\mathcal{M}_h$ located on $\mathcal{M}$, then $\mathbf{r}_{h,i}$ is a linear interpolation of $\mathbf{r}_i$, and typically, $\mathcal{M}_h$ is a regular approximation of $\mathcal{M}$ in such a case, and it converges to $\mathcal{M}$ as $h \to 0$. In the following, we introduce conditions on the triangular meshes which are common conditions to guarantee the supercloseness (cf. [4, Definition 2.4], [34, Definition 1.2], or [42, Definition 3.2]).

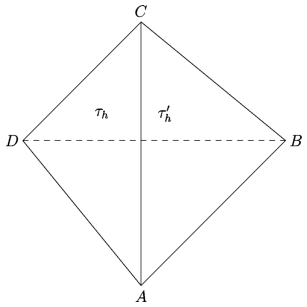


FIG. 1. *Illustration of two adjacent triangles.*

DEFINITION 3.2. *Suppose τ_h and τ_h' are two adjacent triangles in $\mathcal{T}_h$, as illustrated in Figure* 1. *They are said to form an $\mathcal{O}(h^2)$ parallelogram if*

$$|\overrightarrow{AB} - \overrightarrow{CD}| = \mathcal{O}(h^2) \quad \text{and} \quad |\overrightarrow{BC} - \overrightarrow{DA}| = \mathcal{O}(h^2).$$

DEFINITION 3.3. *A triangulation mesh $\mathcal{T}_h$ is said to satisfy the $\mathcal{O}(h^{2\sigma})$* irregular condition *if there exist a partition $\mathcal{T}_{h,1} \bigcup \mathcal{T}_{h,2}$ of $\mathcal{T}_h$ and a positive constant σ such that every two adjacent triangles in $\mathcal{T}_{h,1}$ form an $\mathcal{O}(h^2)$ parallelogram and*

$$\sum_{\tau_h \subset \mathcal{T}_{h,2}} |\tau_h| = \mathcal{O}(h^{2\sigma}).$$

Before going further, we make a general assumption for a nonadaptive triangulation $\mathcal{M}_h$.

ASSUMPTION 3.4. *Let $\mathcal{M}_h$ be a triangulation of $\mathcal{M}$ with all of the nodes located on $\mathcal{M}$. We assume it to be quasi-uniform and shape regular and to be a regular approximation of $\mathcal{M}$. Moreover, it satisfies the $\mathcal{O}(h^{2\sigma})$ irregular condition.*

We have the following lemma for the transform operators.

LEMMA 3.5. *Let $\mathcal{V}(\mathcal{M}) \hookrightarrow W^{k,p}(\mathcal{M})$ for a fixed $k \in \mathbb{N}$ and $p \geq 1$, and $\mathcal{M}_h$ satisfies Assumption* 3.4. *Then the predefined operators $(T_h)^{\pm 1}$ are uniformly bounded between the spaces $W^{k,p}(\mathcal{M})$ and $W^{k,p}(\mathcal{M}_h)$ in the sense that there exist positive constants $c \leq C$, and*

$$c \|T_h v\|_{k,p,\mathcal{M}_h} \leq \|v\|_{k,p,\mathcal{M}} \leq C \|T_h v\|_{k,p,\mathcal{M}_h} \text{ for all } v \in \mathcal{V}(\mathcal{M}).$$

Proof. Denote $\check{v}_h := T_h v$, and let $\tau_j \subset \mathcal{M}$ be the curved triangle corresponding to $\tau_{h,j} \subset \mathcal{M}_h$. If $p = \infty$, every function v and its derivatives are uniformly bounded on $\mathcal{M}$, and the same is true for function $\check{v}_h$ and its derivatives over $\mathcal{M}_h$. Then we can always find constants c_h^1 and C_h^1 satisfying

$$c_h^1 \|\check{v}_h\|_{k,\infty,\mathcal{M}_h} \leq \|v\|_{k,\infty,\mathcal{M}} \leq C_h^1 \|\check{v}_h\|_{k,\infty,\mathcal{M}_h}.$$

If $1 \leq p < \infty$, using the results in [14, page 811], there exist positive and bounded constants $c_{h,j}$ and $C_{h,j}$ for each pair of triangle faces $\tau_{h,j}$ and τ_j such that

$$c_{h,j}^2 \|\check{v}_h\|_{k,p,\tau_{h,j}}^p \leq \|v\|_{k,p,\tau_j}^p \leq C_{h,j}^2 \|\check{v}_h\|_{k,p,\tau_{h,j}}^p.$$

For both cases, due to the regular approximation condition in Assumption 3.4, we have $c_h^1 \to 1$, $C_h^1 \to 1$ when $h \to 0$ as well as $c_{h,j}^2 \to 1$ and $C_{h,j}^2 \to 1$ when $h \to 0$

for all $j \in J_h$. Thus, both $\{c^a_{h,j}\}$ and $\{C^a_{h,j}\}$ are uniformly bounded sequences with respect to the mesh size h and also the index j for $a = 1, 2$. Denote $c := \min_{a,h,j}\{c^a_{h,j}\}$ and $C := \max_{a,h,j}\{C^a_{h,j}\}$. Since $\|v\|^p_{k,p,\mathcal{M}} = \sum_{j\in J_h} \|v\|^p_{k,p,\tau_j}$ for $p \in [1,\infty)$, we have the estimates

$$c\,\|\check{v}_h\|^p_{k,p,\mathcal{M}_h} \le \|v\|^p_{k,p,\mathcal{M}} \le C\,\|\check{v}_h\|^p_{k,p,\mathcal{M}_h},$$

and the same holds for $p = \infty$. This gives the conclusion. ∎

4. Parametric polynomial preserving recovery on manifolds. Our developments are based on the PPR method proposed in [45] for planar problems. It is a robust and high accuracy approach for recovering gradient on mildly unstructured meshes. This idea has been used to develop a Hessian recovery technique in a recent paper [26]. In this paper, we show the possibility of generalizing the idea to problems on manifolds. To simplify the presentation, we shall restrict ourselves to the case of two-dimensional manifolds hereafter.

We will focus on the case where the data is a linear finite element solution on $\mathcal{M}_h$. Therefore, $\mathcal{V}_h(\mathcal{M}_h)$ is restricted to linear finite element spaces in what follows. At each node x_i, let h_i be the length of the longest edge attached to x_i. For any natural number k, let $B_{kh_i}(x_i)$ be the set of vertices in a discrete geodesic ball centered at x_i with discrete geodesic radius $k \times h_i$, i.e.,

$$B_{kh_i}(x_i) = \{x \in \mathcal{N}_h : |x - x_i| \le k \times h_i\}.$$

Then we define $B(x_i) = B_{k_i h_i}(x_i)$ with k_i being the smallest integer such that $B(x_i)$ satisfies the rank condition (see [45]) in the following sense.

DEFINITION 4.1. *A selected vertices set $B(x_i)$ is said to satisfy the rank condition of the PPR or PPPR if it admits a unique least-squares fitted polynomial p_i in* (4.3) *or s_i and p_i in* (4.5) *and* (4.6)*, respectively.*

For a discretized manifold, the main difficulty is that the vertices in $B(x_i)$ are in general not located on the same plane. Another challenge is that there is no trivial definition of polynomials in a manifold setting. An idea that appeared in the literature is to use the domain in tangent space T_{x_i} at every vertex x_i as a local parameter domain, project the neighbored vertices of x_i onto this common planar plane, and then define polynomials locally by the coordinates of the tangent space. This idea has been applied in [20] and also in [42] 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 very feasible in practice. This problem has been proposed as an open question in [42].

In our algorithms, we consider an alternative method for the polynomial reconstruction instead of the one which is initially proposed in [45] for the planar PPR. The method in [45] assumes that a second order polynomial has the form

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

and solves the linear system $A\mathbf{a} = \mathbf{b}$ for $\mathbf{a} = (a_0, a_1, \ldots, a_5)^T$, where

$$A = \begin{pmatrix} 1 & \zeta_{i_1,1} & \zeta_{i_1,2} & \zeta^2_{i_1,1} & \zeta_{i_1,1}\zeta_{i_1,2} & \zeta^2_{i_1,2} \\ 1 & \zeta_{i_2,1} & \zeta_{i_2,2} & \zeta^2_{i_2,1} & \zeta_{i_2,1}\zeta_{i_2,2} & \zeta^2_{i_2,2} \\ \vdots & \cdots & & & & \\ 1 & \zeta_{i_{|I_i|},1} & \zeta_{i_{|I_i|},2} & \zeta^2_{i_{|I_i|},1} & \zeta_{i_{|I_i|},1}\zeta_{i_{|I_i|},2} & \zeta^2_{i_{|I_i|},2} \end{pmatrix} \text{ and } \mathbf{b} = \begin{pmatrix} u_{h,i_1} \\ u_{h,i_2} \\ \vdots \\ u_{h,i_{|I_i|}} \end{pmatrix}. \tag{4.1}$$

The solution of the least-squares approximation in the algorithms is given by

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

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

Our observation is that there is one extra degree of freedom that can be removed in the reconstruction of the polynomials. Since the polynomial recovery procedure cannot improve the accuracy of the solution itself, it is unnecessary to adopt the solution in gradient recovery. We can fix this problem by using the following polynomial equation locally:

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

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 of the plane Ω_i; then the matrix and the vector in (4.1) can be simplified to

(4.2)

$$\tilde{A} = \begin{pmatrix} \zeta_{i_2,1} & \zeta_{i_2,2} & \zeta_{i_2,1}^2 & \zeta_{i_2,1}\zeta_{i_2,2} & \zeta_{i_2,2}^2 \\ \vdots & \dots & & & \\ \zeta_{i_{|I_i|},1} & \zeta_{i_{|I_i|},2} & \zeta_{i_{|I_i|},1}^2 & \zeta_{i_{|I_i|},1}\zeta_{i_{|I_i|},2} & \zeta_{i_{|I_i|},2}^2 \end{pmatrix} \text{ and } \tilde{\mathbf{b}} = \begin{pmatrix} u_{h,i_2} - u_{h,i_1} \\ \vdots \\ u_{h,i_{|I_i|}} - u_{h,i_1} \end{pmatrix}.$$

Solving the problem in the least-squares sense,

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

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

Using (4.2) instead of (4.1) preserves the original function values at the recovered nodal points. This idea can be applied to construct the polynomial functions in the second step of Algorithm 4.1, and also in both the second and third steps of Algorithm 4.2, which are going to be introduced next.

As a starting point, we first provide a 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 the same as the planar one. We sketch it in Algorithm 4.1 and still call it the PPR method. We describe the PPPR method in Algorithm 4.2. In both algorithms, $I_{h,i}$ denotes the set of the indexes of the selected vertices in $B(x_i)$ which satisfies the rank condition.

A straightforward remedy for missing exact normal fields is to find a way to approximate normal vectors at every vertex x_i. This can be done, for instance, by taking the simple average or weighted average of the normal vectors of each face adjacent to x_i. However, with such kinds of approximations, the recovery errors are very likely to be dominated by the errors of the approximation of the normal vector fields (see the numerical results in section 7). Therefore, a better estimation of the normal vectors is necessary in order to have higher recovery accuracy.

Algorithm 4.1 PPR Method (*with exact information of normal vectors*).

Let the discretized triangulation $\mathcal{M}_h$ and the data (FEM solutions) $(u_{h,i})_{i\in I_h}$ be given. Also, we have 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) For every x_i, select $B(x_i) \in \mathcal{M}_h$ including sufficient vertices, shift x_i to be the origin of T_{x_i}, choose an orthonormal basis $(\mathbf{t}_i^1, \mathbf{t}_i^2)$ of $T_{x_i^*}$, and then project the vertices $x_j^* \in B(x_i)$ to T_{x_i}, whose new coordinates read as ζ_{i_j}.

(2) Find a polynomial p_i over T_{x_i} by solving the least-squares problem

$$p_i = \arg\min_{p} \sum_{j\in I_{h,i}} |p(\zeta_{i_j}) - u_{h,j}|^2 \quad \text{for } p \in \mathbb{P}_2(T_{x_i}). \tag{4.3}$$

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

$$G_h^* u_h(x_i) = \partial_1 p_i(0)\mathbf{t}_i^1 + \partial_2 p_i(0)\mathbf{t}_i^2. \tag{4.4}$$

For the recovery of gradient $G_h^* u_h(x)$ when x is not a vertex of triangles, use linear Lagrange finite element basis to interpolate the values $\{G_h^* u_h(x_i)\}_{i\in I_h}$ at vertices of each triangle.

The success of Algorithm 4.1 relies on the availability of the exact normal vector or the exact tangential plane at each vertex. The requirement can be alleviated by using the intrinsic definition of surface gradient as in (2.2). The main idea of Algorithm 4.2 is to approximate the gradient on manifolds using this intrinsic definition. From such a point of view, it is sufficient to approximate the gradient via the Jacobian of a local geometric map and the gradient of the function defined on the local Euclidean space. In principle, the local geometric map can be constructed in terms of arbitrary Euclidean domain Ω_i, which is not necessarily restricted to the tangent spaces $\{T_{x_i}\}_{i\in I_h}$ in Algorithm 4.1. Therefore, the PPPR method (Algorithm 4.2) does not require knowledge of the tangent spaces of $\mathcal{M}$. Lemma 2.2 indicates that for every fixed x_i, taking arbitrary Ω_i, the gradient operator is analytically invariant. The crucial point in practice is that, numerically, the regular shape of the triangles must not be destroyed after projecting them to the domain Ω_i, and also for a superconvergence purpose, the $\mathcal{O}(h^{2\sigma})$ irregular condition should be properly preserved for the projected triangular mesh on Ω_i. Thus, we still have to construct a robust algorithm for this projection. Our suggestion is, at each vertex, to use the simple average or weighted average of the surrounding normal vectors, which can help us to locate and orient a suitable parameter domain Ω_i. This has been adopted in our numerical examples.

Note that if $\Omega_i = T_{x_i}$ for all $i \in I_h$, we shift x_i to be the origin of T_{x_i}, and let $\phi_i^1 = \mathbf{t}_i^1$, $\phi_i^2 = \mathbf{t}_i^2$, $\phi_i^3 = \mathbf{n}_i$; if $\partial_1 s_i(0) = \partial_2 s_i(0) \equiv 0$ for all $i \in I_h$, then the recovered gradient in (4.7) is equal to the one recovered in (4.4).

Let $\bar{G}_h$ be the PPR operator introduced in [34] for planar problems. The recovered gradient values at each vertex x_i by PPPR operator G_h can be represented by $\bar{G}_h$ in the sense of (4.8):

Algorithm 4.2 PPPR Method (*with no exact normal vectors*).

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) For every x_i, select $B(x_i) \in \mathcal{M}_h$ with sufficient vertices, using the simple (weighted) average of the out normal vectors of every triangle with vertices in $B(x_i)$, and normalizing the averaged vector to be ϕ_i^3, and then constructing a local parameter domain Ω_i orthogonal to ϕ_i^3. Shift x_i to be the origin of Ω_i, and choose (ϕ_i^1, ϕ_i^2) as the orthonormal basis of Ω_i, then project all selected vertices $x_j \in B(x_i)$ into the parameter domain Ω_i, and record the new coordinates as ζ_{i_j}.

(2) Reconstruct a second order polynomial surface S_i over Ω_i to approximate the local surface. Typically, it can be approximated locally as a function graph parametrized by Ω_i. That is $S_i = \tilde{\mathbf{r}}_{h,i}(\Omega_i) = \bigcup_{\zeta\in\Omega_i}(\zeta, s_i(\zeta))$, where s_i solves

$$s_i = \arg\min_s \sum_{j\in I_{h,i}} |s(\zeta_{i_j}) - \langle x_j, \phi_i^3\rangle|^2 \quad \text{for } s \in \mathbb{P}_2(\Omega_i). \tag{4.5}$$

(3) Find a second order polynomial p_i over the domain Ω_i by optimizing

$$p_i = \arg\min_p \sum_{j\in I_{h,i}} |p(\zeta_{i_j}) - u_{h,j}|^2 \quad \text{for } p \in \mathbb{P}_2(\Omega_i). \tag{4.6}$$

(4) Calculate the partial derivatives of both the polynomial approximated surface function in step (2) and the approximated polynomial function of FEM solution in step (3). Then approximate the gradient as (4.7) using the local coordinates:

$$G_h u_h(x_i) = \big(\partial_1 p_i(0), \partial_2 p_i(0)\big)\, J^\dagger(s_i)\, \big(\phi_i^1\ \phi_i^2\ \phi_i^3\big)^\top, \tag{4.7}$$

where $J^\dagger(s_i) = (J^\top(s_i)J(s_i))^{-1}J^\top(s_i)$ and $J^\top(s_i) = \begin{pmatrix} 1 & 0 & \partial_1 s_i(0) \\ 0 & 1 & \partial_2 s_i(0)\end{pmatrix}$. Equation (4.7) is derived from (2.3) in Remark 2.1 for calculating (2.2). We multiply with the orthonormal basis $\{\phi_i^1, \phi_i^2, \phi_i^3\}$ because we have to unify the coordinates from local ones to a global one.

For the recovery of the gradient $G_h u_h(x)$ when x is not a vertex of triangles, use linear Lagrange finite element basis to interpolate the values $\{G_h u_h(x_i)\}_{i\in I_h}$ at vertices of each triangle.

$$G_h u_h(x_i) = \bar{G}_h \bar{u}_h(\zeta_i)(\bar{G}_h \mathbf{r}_{h,i}(\zeta_i))^\dagger, \quad \zeta_i \in \Omega_i \text{ is the projection from } x_i. \tag{4.8}$$

Once the gradient at every vertex is recovered, the gradient $G_h u_h$ on the whole domain of $\mathcal{M}_h$ is obtained using interpolation as

$$G_h u_h(x) = \sum_{i\in I_h} \psi_i(x) G_h u_h(x_i), \tag{4.9}$$

where $\{\psi_i\}_{i\in I_h}$ are the nodal basis functions of the finite element space $\mathcal{V}_h(\mathcal{M}_h)$.

Our numerical results will show that choosing the approximations of normal vectors by either simple average or weighted average has very little influence on the

recovery accuracy of the gradient by Algorithm 4.2. This is different from the case in Algorithm 4.1 where the recovery accuracy highly relies on the error of the approximated normal vectors. The relation (4.8) indicates that the analysis of the PPR which has been developed for planar problems can be applied to Algorithm 4.2 to some extent. Moreover, the idea of approximating (2.2) by generalizing the ZZ-scheme seems feasible. One could similarly reconstruct the two level gradient recovery of the surface parametrization function $\mathbf{r}$ and the function $\bar{u}$ iso-parametrically. That is, one could replace the recovery operator $\bar{G}_h$ in (4.8) by using planar ZZ recovery. However, in order to achieve the superconvergence property, this generalization is still subject to the constraint that the meshes are $\mathcal{O}(h^2)$-symmetric.

Remark 4.2. Later we will report an experimental observation that the PPPR is able to give the most competitive results for the recovery of the 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 for unstructured meshes compared to the other methods, especially in that it does not require the $\mathcal{O}(h^2)$-symmetric condition. For a surface with complicated curvature, a well-structured triangulation after projecting to the parametric domains or tangent spaces, it is possible that the good structure is not preserved—for instance, the symmetric condition. 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. As we will see in Numerical Example 2 in section 7 on high curvature surfaces, the PPPR method gives the most competitive results. Quantitative analysis of this property might be worth exploring further as a future topic.

5. Superconvergence analysis. We prove the superconvergence property of the algorithms proposed in the previous section. Although our algorithms are problem independent, to make the discussion simple, we will take (2.6) as our model problem and focus on its approximation using the linear finite element method on triangulated surfaces. The variational formulation of problem (2.6) is given as follows: Find $u \in H^1(\mathcal{M})$ such that

$$\int_{\mathcal{M}} \nabla_g u \cdot \nabla_g v \, dvol = \int_{\mathcal{M}} fv \, dvol \quad \text{for all} \;\; v \in H^1(\mathcal{M}). \tag{5.1}$$

The regularity of the solutions has been proved in [2, Chapter 4]. In the finite element methods, the surface $\mathcal{M}$ is approximated by the triangulation $\mathcal{M}_h$ which satisfies Assumption 3.4, and the finite element space $\mathcal{V}_h(\mathcal{M}_h)$ is the piecewise linear function spaces defined over $\mathcal{M}_h$. The finite element solution is to find $u_h \in \mathcal{V}_h(\mathcal{M}_h)$ such that

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

Since there are quite a few local geometric notations involved, we summarize them in Table 2.

We prepare the proof of the superconvergence of the recovered gradient by first establishing the boundedness of the proposed gradient recovery operators.

TABLE 2
Notations on local geometry.

Notation	Remark
Ω_i	a local parametric domain for patches around vertex x_i
$\mathcal{K}_i$	local triangle patches of $\mathcal{M}_h$ around vertex x_i
$\tau_{h,j}$	$\tau_{h,j} \subset \mathcal{M}_h$ is the jth triangle face
τ_j	$\tau_j \subset \mathcal{M}$ is the jth curved triangle w.r.t. $\tau_{h,j} \subset \mathcal{M}_h$
$\tilde{\mathbf{r}}_{\tau_{h,j}}$	geometric mapping from $\tau_{h,j}$ to τ_j
$\mathbf{r}_{h,i}$	geometric mapping from patches in Ω_i to $\mathcal{K}_i$
$\mathbf{r}_i$	geometric mapping from patches in Ω_i to $\mathcal{M}_i$

LEMMA 5.1. *Choose an arbitrary but fixed vertex x_i, and let $\tau_{h,j}$ be one of the triangles connected to x_i, $\mathcal{K}_i$ be the selected triangle patches, and $\tau_{h,j} \subset \mathcal{K}_i \subset \mathcal{M}_h$. Let $h < 1$ be sufficiently small. Then G_h is a bounded linear operator in the sense that*

$$\|G_h v_h\|_{L^2(\tau_{h,j})} \leq C_{\mathcal{M}} \|\nabla_{g_h} v_h\|_{L^2(\mathcal{K}_i)} \quad \text{for all } v_h \in \mathcal{V}_h(\mathcal{M}_h), \tag{5.3}$$

where $C_{\mathcal{M}}$ is some constant dependent on the geometry of $\mathcal{M}$ and the shape of the triangulation $\mathcal{M}_h$ but independent of h.

Proof. Let us denote $\bar{v}_h = v_h \circ \mathbf{r}_{h,i}$, and recall (2.2). Then we have that on every parametric domain Ω_i,

$$(\nabla_{g_h} v_h) \circ \mathbf{r}_{h,i} = \nabla \bar{v}_h (\partial \mathbf{r}_{h,i})^\dagger \iff \nabla \bar{v}_h = (\nabla_{g_h} v_h) \circ \mathbf{r}_{h,i} \partial \mathbf{r}_{h,i}, \tag{5.4}$$

where $\partial \mathbf{r}_{h,i}$ and $(\partial \mathbf{r}_{h,i})^\dagger$ are piecewise constant functions. We take into account the assumptions that $\mathcal{M}$ is regular and C^3 smooth, and $\mathcal{M}_h$ is a regular approximation as specified in Definition 3.1. Then there exist positive constants c_r and C_r, such that

$$\frac{n}{C_r} \leq |\partial \mathbf{r}_{h,i}| \leq \frac{n}{c_r} \quad \text{and} \quad c_r \leq |(\partial \mathbf{r}_{h,i})^\dagger| \leq C_r \quad \text{for all } h \text{ and } i, \tag{5.5}$$

where n is the dimension number of $\mathcal{M}$. Correspondingly there are similar bounds for $|(\partial \mathbf{r}_i)^\dagger|$, and we denote them by c_r^* and C_r^*. Because $G_h u_h(x)$ is the linear interpolation of recovered gradient at three vertices of each triangle, and because of the fact that every $\tau_{h,j}$ is uniformly bounded, we get

$$|G_h v_h|(x) \leq \sum_{i \in V_j} |G_h v_h(x_i)| \leq C \max_{i \in V_j} |G_h v_h(x_i)| \quad \text{for all } x \in \tau_{h,j}, \tag{5.6}$$

where V_j denotes the index set of the vertices on $\tau_{h,j}$. Using the boundedness result of the planar PPR recovery operator [34, Theorem 3.2], we have for every $\tau_{h,j} \subset \mathcal{K}_i$

$$\left\|\bar{G}_h \bar{v}_h\right\|_{L^\infty(\tau_{h,j})} \leq C \|\nabla \bar{v}_h\|_{L^\infty(\Omega_i)}.$$

Now we are going to show that $\left\|(\bar{G}_h \mathbf{r}_{h,i})^\dagger\right\|_{L^\infty(\tau_{h,j})}$ is uniformly bounded for all $i \in I_h$ and $j \in J_h$. By noticing that $\mathbf{r}_{h,i}$ is a linear interpolation of $\mathbf{r}_i$ on Ω_i, the polynomial preserving property of the planar PPR operator implies

$$|\bar{G}_h \mathbf{r}_{h,i} - \partial \mathbf{r}_i| = \|\mathbf{r}_i\|_{3,\infty} \mathcal{O}(h^2). \tag{5.7}$$

The hidden constant is independent of both $\mathbf{r}_i$ and h. Taking into account that $\partial \mathbf{r}_i$ is uniformly bounded from below and above and $\mathbf{r}_i$ belongs to $W^{3,\infty}(\Omega_i)$, we can deduce

from (5.7) that there exists a constant c (dependent on $\|\mathbf{r}_i\|_{3,\infty}$) for sufficiently small $h^2 \le h_0^2$ (some fixed $h_0 \in \mathbb{R}^+$) such that

$$|\bar{G}_h \mathbf{r}_{h,i}| \ge \frac{n}{C_r^*} - ch_0^2 \;\Rightarrow\; \left\|(\bar{G}_h \mathbf{r}_{h,i})^\dagger\right\|_{L^\infty(\tau_{h,j_i})} \le \hat{C}, \tag{5.8}$$

where $\hat{C} := \frac{n}{\frac{n}{C_r^*} - ch_0^2}$. For every $\tau_{h,j} \subset \mathcal{K}_i$, we conclude that

$$\max_{i \in V_j} |G_h v_h(x_i)| \le \left\|\bar{G}_h \bar{v}_h\right\|_{L^\infty(\tau_{h,j})} \left\|(\bar{G}_h \mathbf{r}_{h,i})^\dagger\right\|_{L^\infty(\tau_{h,j})} \le \hat{C} \left\|\nabla \bar{v}_h\right\|_{L^\infty(\Omega_i)},$$

which together with (5.6) gives

$$\|G_h v_h\|_{L^\infty(\tau_{h,j})} \le C\hat{C} \left\|\nabla \bar{v}_h\right\|_{L^\infty(\Omega_i)}.$$

Using the formula on the right-hand side of (5.4) and the bounds on $|\partial \mathbf{r}_{h,i}|$ in (5.5), we get the boundedness result for G_h:

$$\|G_h v_h\|_{L^\infty(\tau_{h,j})} \le \frac{nC\hat{C}}{c_r} \left\|\nabla_{g_h} v_h\right\|_{L^\infty(\mathcal{K}_i)}.$$

All the constants here merely depend on $\partial \mathbf{r}$ (note that $|\partial \mathbf{r}_{h,i}|$ is also bounded by $|\partial \mathbf{r}_i|$) independent of h for all $h \le h_0$. Since v_h is a piecewise linear polynomial on $\mathcal{M}_h$, the inverse estimate implies

$$\left\|\nabla_{g_h} v_h\right\|_{L^\infty(\mathcal{K}_i)} \le \frac{C_{in}}{\sqrt{|\mathcal{K}_i|}} \left\|\nabla_{g_h} v_h\right\|_{L^2(\mathcal{K}_i)}$$

for some constant C_{in} independent of h. Here $|\mathcal{K}_i|$ denotes the area of $\mathcal{K}_i$. Finally, we have

$$\begin{aligned}\|G_h v_h\|_{L^2(\tau_{h,j})} &\le \sqrt{|\tau_{h,j}|}\, \|G_h v_h\|_{L^\infty(\tau_{h,j})} \le \sqrt{|\tau_{h,j}|} \frac{nC\hat{C}}{c_r} \left\|\nabla_{g_h} v_h\right\|_{L^\infty(\mathcal{K}_i)} \\ &\le \frac{nC\hat{C}C_{in}\sqrt{|\tau_{h,j}|}}{c_r\sqrt{|\mathcal{K}_i|}} \left\|\nabla_{g_h} v_h\right\|_{L^2(\mathcal{K}_i)}.\end{aligned}$$

The fact $\frac{\sqrt{|\tau_{h,j}|}}{\sqrt{|\mathcal{K}_i|}} \le 1$ indicates that we can set $C_{\mathcal{M}} := \frac{nC\hat{C}C_{in}}{c_r}$ and then

$$\|G_h v_h\|_{L^2(\tau_{h,j})} \le C_{\mathcal{M}} \left\|\nabla_{g_h} v_h\right\|_{L^2(\mathcal{K}_i)},$$

which completes the proof. ∎

The boundedness of the operator G_h^* in Algorithm 4.1 is a trivial case implicated by Lemma 5.1. Next we show the consistency of the PPPR gradient recovery operator by establishing the following lemma.

LEMMA 5.2. *Let $u \in W^{3,\infty}(\mathcal{M})$, and let u_I be the linear interpolation of the function u at every vertex of $\mathcal{M}_h$; also the assumption of Lemma* 5.1 *holds. Then we have the estimate*

$$\left\|\nabla_g u - (T_h)^{-1} G_h u_I\right\|_{0,\mathcal{M}} \le h^2 \sqrt{\mathcal{A}(\mathcal{M})} D(g, g^{-1}) \|u\|_{3,\infty,\mathcal{M}}, \tag{5.9}$$

where $D(g, g^{-1})$ is a constant determined by the metric tensor g and its inverse.

Proof. We start from a single triangle $\tau_{h,j} \subset \mathcal{M}_h$ of index j and then go through all $j \in J_h$. In particular, we consider the gradient formulation (2.2) on each triangle. Let $\tau_j \subset \mathcal{M}$ be the curved triangle corresponding to $\tau_{h,j} \subset \mathcal{M}_h$. Then we have

$$\left\|\nabla_g u - (T_h)^{-1} G_h u_I\right\|_{0,\tau_j}^2 = \int_{\tau_{h,j}} |\nabla \bar{u}_{\tau_{h,j}} (\partial \tilde{\mathbf{r}}_{\tau_{h,j}})^\dagger - G_h(u_I)|_{\tilde{\mathbf{r}}_{\tau_{h,j}}}|^2 \det(g \circ \tilde{\mathbf{r}}_{\tau_{h,j}}),$$

where $\bar{u}_{\tau_{h,j}} = u \circ \tilde{\mathbf{r}}_{\tau_{h,j}}$ and $\tilde{\mathbf{r}}_{\tau_{h,j}}$ is the geometrical mapping from $\tau_{h,j}$ to τ_j, that is, $\tau_j = \tilde{\mathbf{r}}_{\tau_{h,j}}(\tau_{h,j})$. $G_h(u_I)|_{\tilde{\mathbf{r}}_{\tau_{h,j}}}$ is the recovered gradient over the triangle $\tau_{h,j}$ by interpolating gradient values recovered at the vertices of $\tau_{h,j}$. Therefore, in the local coordinates of $\tau_{h,j}$, they are first order polynomials.

On the other hand, at every vertex x_i, let $\bar{u}_i(\zeta_i) = u \circ \mathbf{r}_{h,i}(\zeta_i) = u \circ \mathbf{r}_i(\zeta_i)$. The consistency of the PPR operator $\bar{G}_h$ on the planar domain [35, Theorem 2.2] implies that

$$|\nabla \bar{u}_i(\zeta_i) - \bar{G}_h \bar{u}_I(\zeta_i)| \le C h^2 \|\bar{u}\|_{3,\infty,\Omega_i}, \tag{5.10}$$

where ζ_i is the local coordinates for x_i. Let θ_i be coordinates for x_i on $\tau_{h,j}$. Then we have

$$\nabla_g u(x_i) = \nabla \bar{u}_i(\zeta_i)(\partial \mathbf{r}_i(\zeta_i))^\dagger = \nabla \bar{u}_{\tau_{h,j}}(\theta_i)(\partial \tilde{\mathbf{r}}_{\tau_{h,j}}(\theta_i))^\dagger \tag{5.11}$$

and $G_h u_I(x_i) = \bar{G}_h \bar{u}_I(\zeta_i)(\bar{G}_h \mathbf{r}_{h,i}(\zeta_i))^\dagger$. Note that because both $\partial \mathbf{r}_i$ and $\bar{G}_h \mathbf{r}_{h,i}$ (see (5.8)) are uniformly bounded from below, using the consistency error estimation (5.7), we derive

$$|(\partial \mathbf{r}_i(\zeta_i))^\dagger - (\bar{G}_h \mathbf{r}_{h,i}(\zeta_i))^\dagger| \lesssim h^2 \|\mathbf{r}_i\|_{3,\infty,\Omega_i}. \tag{5.12}$$

By the triangle inequality and the estimates (5.10) and (5.12), we obtain

$$\begin{aligned}
&|\nabla \bar{u}_i(\zeta_i)(\partial \mathbf{r}_i(\zeta_i))^\dagger - G_h u_I(x_i)| \\
\le& |\nabla \bar{u}_i(\zeta_i)(\partial \mathbf{r}_i(\zeta_i))^\dagger - \bar{G}_h \bar{u}_I(\zeta_i)(\partial \mathbf{r}_i(\zeta_i))^\dagger| \\
&+ |\bar{G}_h \bar{u}_I(\zeta_i)(\partial \mathbf{r}_i(\zeta_i))^\dagger - \bar{G}_h \bar{u}_I(\zeta_i)(\bar{G}_h \mathbf{r}_{h,i}(\zeta_i))^\dagger| \\
\lesssim& h^2 \|\bar{u}\|_{3,\infty,\Omega_i} |(\partial \mathbf{r}_i(\zeta_i))^\dagger| + h^2 \|\mathbf{r}_i\|_{3,\infty,\Omega_i} |\bar{G}_h \bar{u}_I(\zeta_i)| \\
\lesssim& h^2 \left(\|\bar{u}\|_{3,\infty,\Omega_i} \left\|(\partial \mathbf{r}_i)^\dagger\right\|_{0,\infty,\Omega_i} + \|\mathbf{r}_i\|_{3,\infty,\Omega_i} \|\bar{u}_I\|_{1,\infty,\Omega_i} \right).
\end{aligned} \tag{5.13}$$

Since $u \in W^{3,\infty}(\mathcal{M})$, each component of $\nabla \bar{u}_{\tau_{h,j}}(\partial \tilde{\mathbf{r}}_{\tau_{h,j}})^\dagger$, the vector-valued function, belongs to $W^{2,\infty}(\tau_{h,j})$. Let $\left(\nabla \bar{u}_{\tau_{h,j}}(\partial \tilde{\mathbf{r}}_{\tau_{h,j}})^\dagger\right)_I$ be the componentwise linear interpolation of the vector-valued function $\nabla \bar{u}_{\tau_{h,j}}(\partial \tilde{\mathbf{r}}_{\tau_{h,j}})^\dagger$ on $\tau_{h,j}$. Using the triangle inequality, we obtain

$$\begin{aligned}
&\left\|\nabla \bar{u}_{\tau_{h,j}}(\partial \tilde{\mathbf{r}}_{\tau_{h,j}})^\dagger - G_h u_I\right\|_{0,\tau_{h,j}} \\
\le& \left\|\nabla \bar{u}_{\tau_{h,j}}(\partial \tilde{\mathbf{r}}_{\tau_{h,j}})^\dagger - \left(\nabla \bar{u}_{\tau_{h,j}}(\partial \tilde{\mathbf{r}}_{\tau_{h,j}})^\dagger\right)_I\right\|_{0,\tau_{h,j}} \\
&+ \left\|\left(\nabla \bar{u}_{\tau_{h,j}}(\partial \tilde{\mathbf{r}}_{\tau_{h,j}})^\dagger\right)_I - G_h u_I\right\|_{0,\tau_{h,j}} \\
:=& F_1 + F_2.
\end{aligned} \tag{5.14}$$

By the standard error estimate of linear interpolation [5, 11], we can bound F_1 as

$$\begin{aligned}
F_1 \lesssim& h^2 |\nabla \bar{u}_{\tau_{h,j}}(\partial \tilde{\mathbf{r}}_{\tau_{h,j}})^\dagger|_{2,\tau_{h,j}} = h^2 |\nabla \bar{u}(\partial \mathbf{r}_i)^\dagger|_{2,\tau_{h,j}} \\
\lesssim& \|\bar{u}\|_{3,\infty,\Omega_i} \left\|(\partial \mathbf{r}_i)^\dagger\right\|_{2,\infty,\Omega_i} \sqrt{\mathcal{A}(\tau_{h,j})}.
\end{aligned} \tag{5.15}$$

To estimate F_2, we use the relationship (5.11) and (5.13) which deduce

$$
\begin{aligned}
F_2 &\lesssim \sum_{i\in V_j} |\nabla \bar{u}_{\tau_{h,j}}(\theta_i)(\partial \tilde{\mathbf{r}}_{\tau_{h,j}}(\theta_i))^\dagger - G_h u_I(x_i)| \sqrt{\mathcal{A}(\tau_{h,j})} \\
&= \sum_{i\in V_j} |\nabla \bar{u}_i(\zeta_i)(\partial \mathbf{r}_i(\zeta_i))^\dagger - G_h u_I(x_i)| \sqrt{\mathcal{A}(\tau_{h,j})} \\
&\lesssim h^2 \sum_{i\in V_j} \left(\|\bar{u}\|_{3,\infty,\Omega_i} \left\|(\partial \mathbf{r}_i)^\dagger\right\|_{2,\infty,\Omega_i} + \|\mathbf{r}_i\|_{3,\infty,\Omega_i} \|\bar{u}\|_{1,\infty,\Omega_i} \right) \sqrt{\mathcal{A}(\tau_{h,j})}.
\end{aligned} \tag{5.16}
$$

On each local parametric domain Ω_i, because of the regular property of $\mathcal{M}$, we have the following facts: (a) $g \circ \mathbf{r}_i = \partial \mathbf{r}_i (\partial \mathbf{r}_i)^T$; then $\|\partial \mathbf{r}_i\|_{k,\infty,\Omega_i}$ and $\left\|(\partial \mathbf{r}_i)^\dagger\right\|_{k,\infty,\Omega_i}$ for $k \in \{0,1,2\}$ on all Ω_i can be estimated by $\sqrt{\|g\|_{k,\infty}}$ and $\sqrt{\|g^{-1}\|_{k,\infty}}$, respectively. (b) Since $\mathcal{M}$ is C^3 smooth and regular with bounded curvature, both g and g^{-1} and their derivatives up to second order are uniformly bounded from below and above. On the other hand, we can estimate the norms

$$
\|\bar{u}\|_{k+1,\infty,\Omega_i} \le \left\|(\det g)^{-1}\right\|_{0,\infty} \sqrt{\|g\|_{k,\infty}} \|u\|_{k+1,\infty,\mathcal{M}} \quad \text{for } k \in \{0,1,2\}.
$$

Combining the estimates in (5.15) and (5.16) gives us

$$
\begin{aligned}
&\left(\|\bar{u}\|_{3,\infty,\Omega_i} \left\|(\partial \mathbf{r}_i)^\dagger\right\|_{2,\infty,\Omega_i} + \|\mathbf{r}_i\|_{3,\infty,\Omega_i} \|\bar{u}\|_{1,\infty,\Omega_i} \right) \\
\le& \sqrt{\|(\det g)^{-1}\|_{0,\infty}} \left(\|u\|_{3,\infty,\mathcal{M}} \sqrt{\|g\|_{2,\infty}} \sqrt{\|g^{-1}\|_{2,\infty}} \right. \\
&\left. + \sqrt{\|g\|_{2,\infty}} \sqrt{\|g\|_{0,\infty}} \|u\|_{1,\infty,\mathcal{M}} \right) \\
\le& C(g, g^{-1}) \|u\|_{3,\infty,\mathcal{M}},
\end{aligned} \tag{5.17}
$$

where $C(g, g^{-1})$ are constants determined by the geometry of $\mathcal{M}$, and they are uniformly bounded whenever Assumption 3.4 is satisfied. Using the local estimate (5.14) and (5.17), we can further derive that

$$
\begin{aligned}
&\int_{\tau_{h,j}} |\nabla \bar{u}_{\tau_{h,j}}(\partial \tilde{\mathbf{r}}_{\tau_{h,j}})^\dagger - G_h(u_I)|^2 \det(g \circ \tilde{\mathbf{r}}_{\tau_{h,j}}) \\
\le& \|\det g\|_{0,\infty} \left\|\nabla \bar{u}_{\tau_{h,j}}(\partial \tilde{\mathbf{r}}_{\tau_{h,j}})^\dagger - G_h(u_I)\right\|^2_{0,\tau_{h,j}} \\
\le& h^4 |V_j|^2 \|\det g\|_{0,\infty} (C(g, g^{-1}) \|u\|_{3,\infty,\mathcal{M}})^2 \mathcal{A}(\tau_{h,j}).
\end{aligned} \tag{5.18}
$$

Note that $|V_j| \equiv 3$ in our case. Summing over both sides of (5.18) for all indexes $j \in J_h$ and taking the square root, we get the final conclusion. The constant

$$
D(g, g^{-1}) = |V_j| \sqrt{\|\det g\|_{0,\infty}} C(g, g^{-1}),
$$

where $C(g, g^{-1})$ is as given in (5.17). Note that here the summation is bounded as we consider $\mathcal{M}$ to be compact and the fact that $\mathcal{A}(\mathcal{M}_h) \le \mathcal{A}(\mathcal{M})$; thus it does not reduce the order of h. ∎

Now we are ready to show the superconvergence of the recovered gradient on $\mathcal{M}_h$.

THEOREM 5.3. *Let the conditions in Lemma* 5.1 *and Assumption* 3.4 *hold, and let* $u \in W^{3,\infty}(\mathcal{M})$ *be the solution of* (5.1) *and* u_h *be the solution of* (5.2). *Then*

$$\begin{aligned}\left\|\nabla_g u - T_h^{-1} G_h u_h\right\|_{0,\mathcal{M}} \leq \quad & h^2 \left(\sqrt{\mathcal{A}(\mathcal{M})} D(g, g^{-1}) \|u\|_{3,\infty,\mathcal{M}} + \|f\|_{0,\mathcal{M}}\right) \\ & + C h^{1+\min\{1,\sigma\}} \left(\|u\|_{3,\mathcal{M}} + \|u\|_{2,\infty,\mathcal{M}}\right),\end{aligned} \tag{5.19}$$

where $D(g, g^{-1})$ *is the same constant as in Lemma* 5.2.

Proof. This is readily shown by considering the triangle inequality

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

The first term is bounded by Lemma 5.2. For the second term, since both $(T_h)^{-1}$ and G_h are bounded operators (Lemma 3.5 and Lemma 5.2), we have

$$\left\|(T_h)^{-1} G_h (u_I - u_h)\right\|_{0,\mathcal{M}} \leq C \left\|\nabla_{g_h} (u_I - u_h)\right\|_{0,\mathcal{M}_h}.$$

Then we use the result[1] of [42, Theorem 3.5] to estimate $\|\nabla_{g_h}(u_I - u_h)\|_{0,\mathcal{M}_h}$. These lead to the final estimate. ∎

Due to Lemma 3.5, we have the following result immediately, which is verified in the numerical part of the paper (section 7).

COROLLARY 5.4. *Given the same assumptions as in Theorem* 5.3, *we have*

$$\begin{aligned}\left\|T_h \nabla_g u - G_h u_h\right\|_{0,\mathcal{M}_h} \leq \quad & h^2 \left(\sqrt{\mathcal{A}(\mathcal{M})} D(g, g^{-1}) \|u\|_{3,\infty,\mathcal{M}} + \|f\|_{0,\mathcal{M}}\right) \\ & + C h^{1+\min\{1,\sigma\}} \left(\|u\|_{3,\mathcal{M}} + \|u\|_{2,\infty,\mathcal{M}}\right).\end{aligned} \tag{5.20}$$

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 $\tau_{h,j}$ as

$$\eta_{h,\tau_{h,j}} = \|G_h u_h - \nabla_{g_h} u_h\|_{0,\tau_{h,j}} \tag{6.1}$$

and the corresponding global error estimator as

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

With the previous superconvergence result, we can show the asymptotic exactness of error estimators based on the recovery operator G_h.

[1]The $\mathcal{O}(h^{2\sigma})$ condition is needed for the projected triangle meshes on each Ω_i in order to show the supercloseness, while what we have assumed is in fact on the meshes before projection as in [42]. We argue that for general smooth surfaces with uniformly bounded curvature, using the methods described in our algorithms, the projected shape of meshes will not be significantly changed as in [42]; therefore, the $\mathcal{O}(h^{2\sigma})$ condition can be guaranteed, although this might be not the case for the meshes located at the high curvature areas. Once a surface is highly curved, one may have to take into account the ratio of the high curvature areas; thus the $\mathcal{O}(h^{2\sigma})$ condition may be adapted to a new index $\bar{\sigma}$ according to the ratio of the high curvature areas. But in this paper, we skip the quantitative discussion on this point.

COROLLARY 6.1. *Assume the same conditions in Theorem* 5.3*, and let* u_h *be the finite element solution of discrete variational problem* (5.2)*. Further assume that there is a constant* $C(u) > 0$ *such that*

$$\|T_h\nabla_g u - \nabla_{g_h} u_h\|_{0,\mathcal{M}_h} \geq C(u)h. \tag{6.3}$$

Then it holds that

$$\left| \frac{\eta_h}{\|T_h\nabla_g u - \nabla_{g_h} u_h\|_{0,\mathcal{M}_h}} - 1 \right| \lesssim h^{\min\{1,\sigma\}}. \tag{6.4}$$

Proof. By the triangle inequality, we have

$$\eta_h \leq \|G_h u_h - T_h\nabla_g u\|_{0,\mathcal{M}_h} + \|T_h\nabla_g u - \nabla_{g_h} u_h\|_{0,\mathcal{M}_h}$$

and hence

$$\left| \frac{\eta_h}{\|T_h\nabla_g u - \nabla_{g_h} u_h\|_{0,\mathcal{M}_h}} - 1 \right| \leq \frac{\|G_h u_h - \nabla_{g_h} u_h\|_{0,\mathcal{M}_h}}{\|T_h\nabla_g u - \nabla_{g_h} u_h\|_{0,\mathcal{M}_h}} \lesssim h^{\min\{1,\sigma\}},$$

where we use the superconvergence result (5.20) and the assumption (6.3) in the last inequality. ∎

Remark 6.2. It is common to assume (6.3) for the asymptotical exactness of recovery-based a posteriori error estimators as in [1, 34, 45]. This is reasonable since the finite element solution error is not better than the interpolation error, which is bounded from below by $\mathcal{O}(h)$(except for some trivial cases).

Remark 6.3. Corollary 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 several numerical examples to demonstrate the superconvergence property of the proposed gradient recovery operators and make comparisons with existing gradient recovery operators. The first example shows the superconvergence results of the proposed gradient recovery operators even though the element patch is not $\mathcal{O}(h^2)$-symmetric. In this example, the vertices are located exactly on the torus. The second one compares the results on a more complicated surface and demonstrates the superiority of the PPPR method for surfaces with high curvature; this is an example in which the vertices are not located on the exact surfaces. The last two examples show the asymptotic exactness of the recovery-based a posteriori error estimator introduced in section 6. Some of our numerical tests are conducted based on the MATLAB package *i*FEM [9]. The initial meshes of the second, third, and fourth examples are generated using the three-dimensional surface mesh generation module of the Computational Geometry Algorithms Library [41]. To get meshes in other levels, we first perform either the uniform refinement or the newest bisection [8]. Then we project the newest vertices onto the $\mathcal{M}$. In the general case, there is no explicit project map available. Hence we adopt the first order approximation of projection map as given in [15]. Thus, the vertices of the meshes are not on the exact surface $\mathcal{M}$ but in an h^2 neighborhood for the second and fourth examples. We notice that in such cases the superconvergence results can still be observed.

Let G_h^{SA}, G_h^{WA}, and G_h^{ZZ} be recovery operators by simple averaging, weighted averaging, and the ZZ-scheme on tangent planes [42], respectively. Note that we use

the exact normal vectors for G_h^{ZZ} in the numerical examples. We denote G_h^*, G_h, and G_h^a to be the recovery operators given by Algorithm 4.1, Algorithm 4.2, and Algorithm 4.1 with approximations of normal vectors, respectively. The approximating normal vectors are computed by weighted averaging for the tests with G_h^a in our examples and are also used to implement Algorithm 4.2 to construct the local parametric domains Ω_i. We also remark that we use the function value preserving skill for the PPPR G_h, but not for G_h^*. For purposes of making comparisons, we define

$$De = \|T_h\nabla_g u - \nabla_{g_h} u_h\|_{0,\mathcal{M}_h}, \qquad De^I = \|\nabla_{g_h} u_I - \nabla_{g_h} u_h\|_{0,\mathcal{M}_h},$$
$$De^{r_1} = \|T_h\nabla_g u - G_h^* u_h\|_{0,\mathcal{M}_h}, \qquad De^{r_2} = \|T_h\nabla_g u - G_h u_h\|_{0,\mathcal{M}_h},$$
$$De^{r_3} = \|T_h\nabla_g u - G_h^a u_h\|_{0,\mathcal{M}_h}, \qquad De^{SA} = \left\|T_h\nabla_g u - G_h^{SA} u_h\right\|_{0,\mathcal{M}_h},$$
$$De^{WA} = \left\|T_h\nabla_g u - G_h^{WA} u_h\right\|_{0,\mathcal{M}_h}, \qquad De^{ZZ} = \left\|T_h\nabla_g u - G_h^{ZZ} u_h\right\|_{0,\mathcal{M}_h},$$

where u_h is the finite element solution, u is the analytical solution, and u_I is the linear finite element interpolation of u.

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

$$De_0^{r_1} = \|T_h\nabla_g u - G_h^* u_h\|_{0,\infty,\mathcal{M}_h}, \qquad De_0^{r_2} = \|T_h\nabla_g u - G_h u_h\|_{0,\infty,\mathcal{M}_h},$$
$$De_0^{r_3} = \|T_h\nabla_g u - G_h^a u_h\|_{0,\infty,\mathcal{M}_h}, \qquad De_0^{SA} = \left\|T_h\nabla_g u - G_h^{SA} u_h\right\|_{0,\infty,\mathcal{M}_h},$$
$$De_0^{WA} = \left\|T_h\nabla_g u - G_h^{WA} u_h\right\|_{0,\infty,\mathcal{M}_h}, \qquad De_0^{ZZ} = \left\|T_h\nabla_g u - G_h^{ZZ} u_h\right\|_{0,\infty,\mathcal{M}_h},$$

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

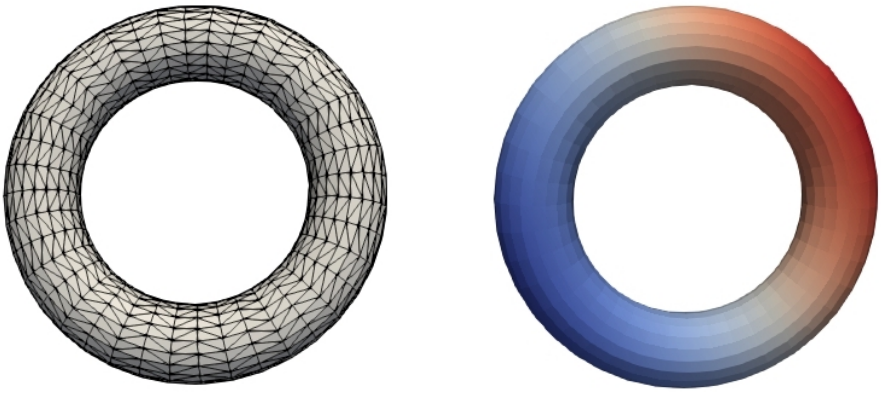
In the following tables, all convergence rates are listed in terms of the square root of the degree of freedom (Dof). Noticing that Dof $\approx h^{-2}$, the corresponding convergence rates in term of the mesh size h are the same as those we present in the tables.

7.1. Numerical Example 1. Our first example considers the 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 the torus surface is

$$\Phi(x) = \sqrt{(\sqrt{x_1^2 + x_2^2} - 4)^2 + x_3^2} - 1. \tag{7.1}$$

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

Table 3 lists the numerical results. As expected, H^1 error of the finite element solution is of $\mathcal{O}(h)$. Since the generated uniform meshes satisfy the $\mathcal{O}(h^{2\sigma})$ condition, $\mathcal{O}(h^2)$ 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 have a superconvergence rate of order $\mathcal{O}(h^2)$, while the recovered gradient using PPR with approximated normal field and the other three methods in [42] only converge at the optimal rate $\mathcal{O}(h)$.



(a) (b)

FIG. 2. *Numerical solution on torus surface:* (a) *Mesh.* (b) *Solution.*

TABLE 3
Numerical results for (5.1) *on torus surface.*

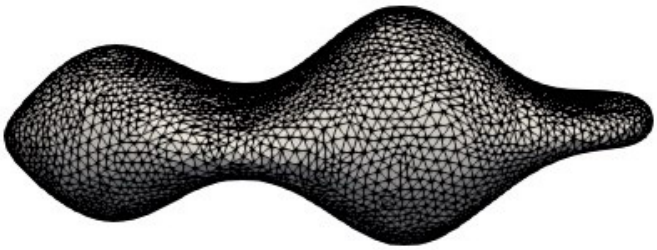
Dof	De	Order	De^I	Order	De^{r_1}	Order	De^{r_2}	Order
200	2.52e+00	–	9.43e-01	–	1.50e+00	–	1.59e+00	–
800	1.26e+00	1.00	2.65e-01	1.83	4.12e-01	1.86	4.37e-01	1.87
3200	6.29e-01	1.00	6.92e-02	1.94	1.06e-01	1.96	1.13e-01	1.96
12800	3.14e-01	1.00	1.75e-02	1.98	2.67e-02	1.99	2.84e-02	1.99
51200	1.57e-01	1.00	4.40e-03	1.99	6.70e-03	2.00	7.12e-03	2.00
204800	7.86e-02	1.00	1.10e-03	2.00	1.67e-03	2.00	1.78e-03	2.00
819200	3.93e-02	1.00	2.75e-04	2.00	4.19e-04	2.00	4.45e-04	2.00
3276800	1.97e-02	1.00	6.88e-05	2.00	1.05e-04	2.00	1.11e-04	2.00
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	1.68	7.22e-01	1.65	7.25e-01	1.65	6.91e-01	1.71
3200	1.68e-01	1.50	2.48e-01	1.54	2.49e-01	1.54	2.19e-01	1.66
12800	7.18e-02	1.22	1.03e-01	1.27	1.03e-01	1.27	8.39e-02	1.38
51200	3.42e-02	1.07	4.86e-02	1.09	4.86e-02	1.09	3.80e-02	1.14
204800	1.69e-02	1.02	2.39e-02	1.02	2.39e-02	1.02	1.84e-02	1.04
819200	8.40e-03	1.00	1.19e-02	1.01	1.19e-02	1.01	9.16e-03	1.01
3276800	4.20e-03	1.00	5.94e-03	1.00	5.94e-03	1.00	4.57e-03	1.00

7.2. Numerical Example 2. In this example, we take a surface [22] which contains high curvature features. It can be represented as the zero level of the following level set function:

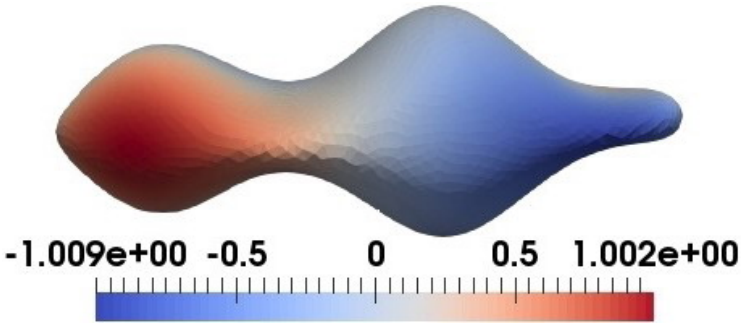
$$\Phi(x) = \frac{1}{4}x_1^2 + x_2^2 + \frac{4x_3^2}{(1+\frac{1}{2}\sin(\pi x_1))^2} - 1.$$

We consider the Laplace–Beltrami equation (2.6) with exact solution $u = x_1x_2$. The right-hand side function f can be computed from u.

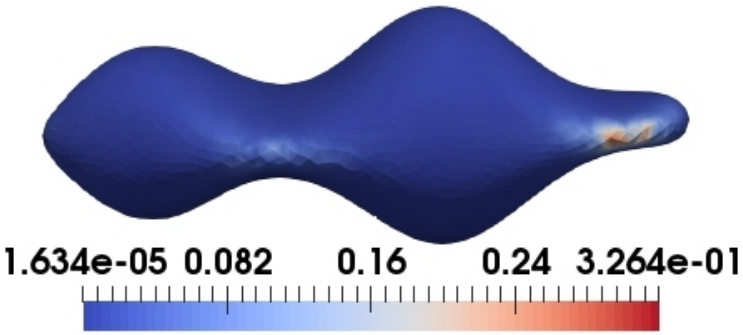
Figure 3b shows the finite element solution u_h on a Delaunay mesh (see Figure 3a) with 4606 Dofs. The numerical results are reported in Table 4. From the table, we clearly see that De converges at the optimal rate $\mathcal{O}(h)$ and De^I converges at a superconvergent rate $\mathcal{O}(h^2)$. As demonstrated in [10], some regions of the surface show significant high curvature. Due to the existence of these areas, only a sub-



(a) Mesh



(b) Solution



(c) Error distribution of recovered gradient

FIG. 3. *Numerical solution on a general surface.*

superconvergence rate of order $\mathcal{O}(h^{1.8})$ is observed for PPR with approximated normal field and the other three methods in [42]. 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 carefully into the recovery accuracy at the high curvature area, we add another set of comparisons in this example. In our numerical tests, we have observed that the maximal recovery errors always happened in the region where the meshes are generated from surface area with highest curvature. We plot a case example of the distribution of the error function $|G_h u_h - T_h \nabla_g u|$ in Figure 3c. Table 5 reports the maximal discrete errors of all of the above six gradient recovery methods, in which PPPR methods give the best rates for the recovery. This gives the evidence to our statement in Remark 4.2. At that point, we can say that PPPR is the best

method for arbitrary meshes and meshes generated by high curvature surfaces. Thus, in the following two examples, we shall only consider the PPPR method.

TABLE 4
Numerical results for (5.1) *on a general surface.*

Dof	De	Order	De^I	Order	De^{r_1}	Order	De^{r_2}	Order
1153	5.46e-01	–	2.75e-01	–	4.76e-01	–	4.62e-01	–
4606	2.85e-01	0.94	1.18e-01	1.23	2.01e-01	1.24	1.95e-01	1.24
18418	1.40e-01	1.03	3.45e-02	1.77	6.58e-02	1.61	6.45e-02	1.60
73666	6.97e-02	1.00	9.86e-03	1.80	1.97e-02	1.74	1.96e-02	1.72
294658	3.48e-02	1.00	2.58e-03	1.94	5.33e-03	1.89	5.32e-03	1.88
1178626	1.74e-02	1.00	6.57e-04	1.97	1.37e-03	1.96	1.37e-03	1.96
Dof	De^{r_3}	Order	De^{SA}	Order	De^{WA}	Order	De^{ZZ}	Order
1153	4.08e-01	–	4.82e-01	–	4.85e-01	–	4.90e-01	–
4606	1.77e-01	1.21	2.26e-01	1.10	2.30e-01	1.08	2.16e-01	1.18
18418	5.54e-02	1.67	8.30e-02	1.44	8.59e-02	1.42	7.34e-02	1.56
73666	1.61e-02	1.78	2.69e-02	1.63	2.82e-02	1.61	2.28e-02	1.69
294658	4.49e-03	1.85	7.72e-03	1.80	8.27e-03	1.77	6.40e-03	1.83
1178626	1.28e-03	1.81	2.14e-03	1.85	2.36e-03	1.81	1.74e-03	1.88

TABLE 5
Comparison of discrete maximal norms of gradient recovery methods on a general surface.

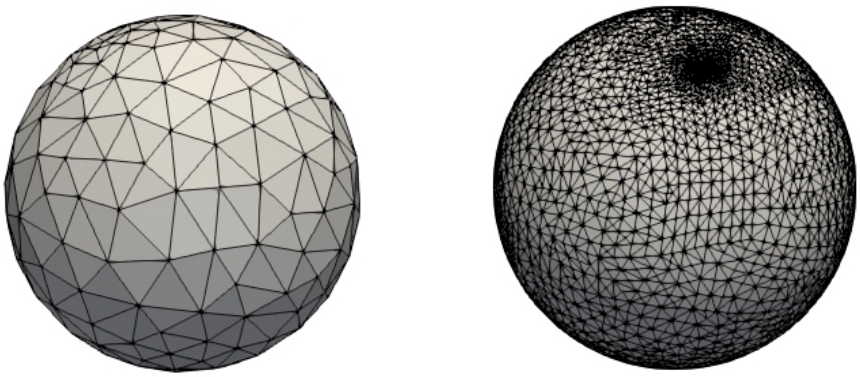
Dof	$De_0^{r_1}$	Order	$De_0^{r_2}$	Order	$De_0^{r_3}$	Order
1153	9.70e-01	–	8.67e-01	–	7.28e-01	–
4606	5.43e-01	0.84	4.66e-01	0.90	5.36e-01	0.44
18418	1.92e-01	1.50	1.87e-01	1.32	2.61e-01	1.04
73666	8.57e-02	1.16	8.25e-02	1.18	1.04e-01	1.33
294658	2.50e-02	1.78	2.54e-02	1.70	4.54e-02	1.19
1178626	7.80e-03	1.68	7.67e-03	1.73	2.10e-02	1.11
Dof	De_0^{SA}	Order	De_0^{WA}	Order	De_0^{ZZ}	Order
1153	7.16e-01	–	7.08e-01	–	8.19e-01	–
4606	5.09e-01	0.49	5.36e-01	0.40	5.69e-01	0.53
18418	2.73e-01	0.90	3.05e-01	0.82	2.43e-01	1.22
73666	1.42e-01	0.95	1.47e-01	1.05	1.08e-01	1.17
294658	5.66e-02	1.32	6.08e-02	1.27	3.46e-02	1.64
1178626	2.39e-02	1.25	2.75e-02	1.14	1.31e-02	1.40

Note that in Table 5 the maximal error of the Algorithm 4.1 (De^{r_1}), which uses the exact normal vectors, is larger than the maximal error of Algorithm 4.2 (De^{r_2}). This is not surprising, as we have reported that in this example, the vertices of the discrete mesh are no longer located on the exact analytical surface. Particularly at the high curvature area, there might be a large deviation between the exact normal vector and the practical normal vector. Therefore, even with exact normal vectors, it brings unavoidable errors to the computations. This also shows an advantage of the PPPR method (Algorithm 4.2).

7.3. Numerical Example 3. In this example, we consider a benchmark problem for the adaptive finite element method for the Laplace–Beltrami equation on the sphere [10, 15, 16]. We choose the right-hand side function f such that the exact solution in spherical coordinates is given by

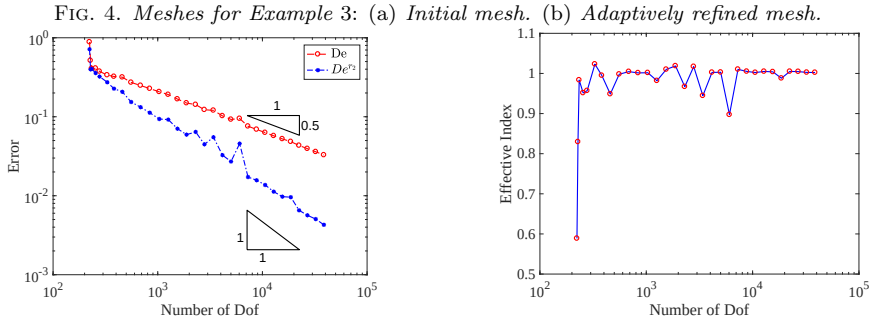
$$u = \sin^{\lambda}(\theta)\sin(\psi).$$

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, $u \in H^{1+\lambda}(\mathcal{M})$.



(a) (b)

FIG. 4. *Meshes for Example* 3: (a) *Initial mesh.* (b) *Adaptively refined mesh.*



(a) (b)

FIG. 5. *Numerical result for Example* 3: (a) *Errors.* (b) *Effective index.*

To obtain the optimal convergence rate, we use the adaptive finite element method (AFEM). Differently from the existing methods in the literature, the recovery-based a posteriori error estimator is adopted. We start with the initial mesh given as in Figure 4a. The mesh is adaptively refined using the Dörfler [19] marking strategy with parameter equal to 0.3. Figure 4b plots the mesh after the 18 adaptive refinement steps. The mesh successfully resolves the singularities. The numerical errors are displayed in Figure 5a. As expected, an optimal convergence rate for H^1 error can be observed. Also, we observe that the recovered gradient is superconvergent to the exact gradient at a rate of $\mathcal{O}(h^2)$.

To test the performance of our new recovery-based a posteriori error estimator for the 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 exact error:

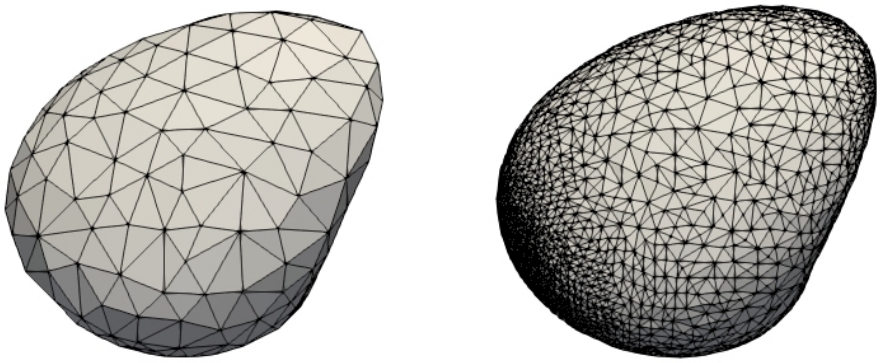
$$\kappa = \frac{\|G_h u_h - \nabla_{g_h} u_h\|_{0,\mathcal{M}_h}}{\|T_h \nabla_g u - \nabla_{g_h} u_h)\|_{0,\mathcal{M}_h}}. \tag{7.2}$$

The effectivity index is plotted in Figure 5b . We see that κ converges asymptotically

to 1, which indicates the posteriori error estimator (6.1) (or (6.2)) is asymptotically exact.

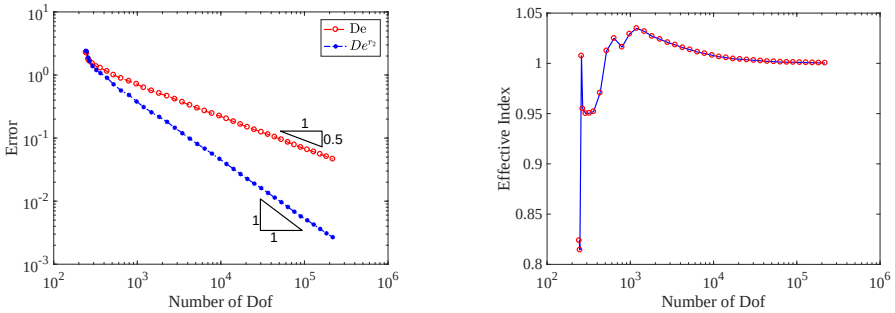
7.4. Numerical Example 4. In this example, we consider a Laplace–Beltrami-type equation on a Dziuk surface as in [12]:

$$-\Delta_g u + u = f \quad \text{on} \quad \mathcal{M}.$$



(a) (b)

FIG. 6. *Meshes for Example* 4: (a) *Initial mesh.* (b) *Adaptively refined mesh.*



(a) (b)

FIG. 7. *Numerical result for Example* 4: (a) *Errors.* (b) *Effective index.*

Here $\mathcal{M}$ is given by $\{x \in \mathbb{R}^3 : (x_1 - x_3^2)^2 + x_2^2 + x_3^2 = 1\}$. 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 phenomenon, we adopt AFEM with an initial mesh graphed in Figure 6a. Figure 6b shows the adaptively 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. Figure 7a displays the numerical errors. It demonstrates the optimal convergence rate in the H^1 norm and a superconvergence rate for the recovered gradient. The effective index is shown

in Figure 7b, 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, a novel gradient recovery method for data defined on manifolds has been proposed. It uses parametric polynomials to fit both manifolds and also finite element method solutions defined on the manifolds, which are shown to be able to recover a better gradient for the finite element method solutions. Compared with existing methods for data on surfaces in the literature (cf. [20, 42]), the proposed method has several improvements: The first highlight is that it does not require the normal vectors of the exact manifold, 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, all of our numerical tests show evidence that PPPR recovers gradients with the best accuracy at high curvature areas compared with the existing methods. We have shown the capability of the recovery operator for constructing a posteriori error estimators. In fact, the superconvergence does hold for $\mathcal{M}_h$ with no exact vertices, but the theoretical proof is postponed to a follow-up paper. Even though we have only discussed linear finite element methods on triangulated meshes, the idea should be applicable to higher order finite element methods on more accurate approximations of surfaces, e.g., piecewise polynomial surfaces, B-splines, or NURBS. However, these are nontrivial works, and we leave them as future topics.

Gradient recovery has other applications, like enhancing eigenvalues [27, 36, 37], simplifying higher order discretization of PDEs [23], and designing new numerical methods for PDEs [7, 28, 29, 33, 44]. Moreover, it may help for the vector field regularization in the context of [18], where the geometric approximation accuracy is needed to be one more order higher than the order of function approximation accuracy needed to optimally regularize vector fields on manifolds. The superconvergence property of the recovery scheme might be able to reduce the additional higher order requirement in manifold approximation for vector fields and achieve optimal convergence rates as in the case of scalar valued functions. It would be interesting to investigate further the full usage of the PPPR method for problems with solutions defined on manifolds.

Appendix A. Proof of Lemma 2.2.

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

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

respectively, where Ω and Ω_s are planar parameter domains. Then there exists

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

a bijective, differentiable mapping, such that $\mathbf{r} = \mathbf{s} \circ \mathbf{t}$. That means that 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 the chain rule on both sides of the former equation of (A.1), 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 nondegenerate. Using the same process but considering $\mathbf{t}^{-1}:\Omega_s\to\Omega$, we can show the reverse implication. Thus, we have shown that any two arbitrary parametrizations $\mathbf{r}$ and $\mathbf{s}$ lead to the same gradient values at the same positions. ∎

Acknowledgments. The authors thank Dr. Pravin Madhavan, Dr. Bjorn Stinner, and Dr. Andreas Dedner for their kind help and discussions on a numerical example. The authors also thank the editor and referees for their efforts, valuable comments, and suggestions, which have significantly improved the paper. Part of the work of GD was done while he was employed and working at Computational Science Center, Faculty of Mathematics, University of Vienna in Austria.

REFERENCES

[1] M. Ainsworth and J. T. Oden, *A Posteriori Error Estimation in Finite Element Analysis*, Pure Appl. Math. (New York), Wiley-Interscience, New York, 2000.

[2] T. Aubin, *Best constants in the Sobolev imbedding theorem: The Yamabe problem*, in Seminar on Differential Geometry, Ann. of Math. Stud. 102, Princeton University Press, Princeton, NJ, 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, part* I. *Grids with superconvergence*, SIAM J. Numer. Anal., 41 (2003), pp. 2294–2312, https://doi.org/10.1137/S003614290139874X.

[5] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, 3rd ed., Texts Appl. Math. 15, Springer, New York, 2008.

[6] 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.

[7] H. Chen, H. Guo, Z. Zhang, and Q. Zou, *A C^0 linear finite element method for two fourth-order eigenvalue problems*, IMA J. Numer. Anal., 37 (2017), pp. 2120–2138.

[8] L. Chen, *Short implementation of bisection in MATLAB*, in Recent Advances in Computational Sciences, World Scientific, Hackensack, NJ, 2008, pp. 318–332.

[9] L. Chen, *iFEM: An Innovative Finite Element Methods Package in MATLAB*, Tech. rep., University of California at Irvine, Irvine, CA, 2009.

[10] 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.

[11] P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, Classics Appl. Math. 40, SIAM, Philadelphia, 2002, https://doi.org/10.1137/1.9780898719208.

[12] A. Dedner and P. Madhavan, *Adaptive discontinuous Galerkin methods on surfaces*, Numer. Math., 132 (2016), pp. 369–398.

[13] 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.

[14] 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, https://doi.org/10.1137/070708135.

[15] 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, https://doi.org/10.1137/050642873.

[16] 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, https://doi.org/10.1137/110842235.

[17] M. P. do Carmo, *Riemannian Geometry*, Mathematics: Theory & Applications, Birkhäuser Boston, Boston, 1992; translated from the second Portuguese edition by Francis Flaherty.
[18] 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.
[19] W. Dörfler, *A convergent adaptive algorithm for Poisson's equation*, SIAM J. Numer. Anal., 33 (1996), pp. 1106–1124, https://doi.org/10.1137/0733054.
[20] Q. Du and L. Ju, *Finite volume methods on spheres and spherical centroidal Voronoi meshes*, SIAM J. Numer. Anal., 43 (2005), pp. 1673–1692, https://doi.org/10.1137/S0036142903425410.
[21] G. Dziuk, *Finite elements for the Beltrami operator on arbitrary surfaces*, in Partial Differential Equations and Calculus of Variations, Lecture Notes in Math. 1357, Springer, Berlin, 1988, pp. 142–155.
[22] G. Dziuk and C. M. Elliott, *Finite element methods for surface PDEs*, Acta Numer., 22 (2013), pp. 289–396.
[23] 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, https://doi.org/10.1137/14097820X.
[24] H. Guo, C. Xie, and R. Zhao, *Superconvergent gradient recovery for virtual element methods*, Math. Models Methods Appl. Sci., 29 (2019), pp. 2007–2031.
[25] H. Guo and Z. Zhang, *Gradient recovery for the Crouzeix-Raviart element*, J. Sci. Comput., 64 (2015), pp. 456–476.
[26] H. Guo, Z. Zhang, and R. Zhao, *Hessian recovery for finite element methods*, Math. Comp., 86 (2017), pp. 1671–1692.
[27] H. Guo, Z. Zhang, and R. Zhao, *Superconvergent two-grid methods for elliptic eigenvalue problems*, J. Sci. Comput., 70 (2017), pp. 125–148.
[28] H. Guo, Z. Zhang, and Q. Zou, *A C^0 linear finite element method for biharmonic problems*, J. Sci. Comput., 74 (2018), pp. 1397–1422.
[29] H. Guo, Z. Zhang, and Q. Zou, *A C^0 Linear Finite Element Method for Sixth Order Elliptic Equations*, preprint, https://arxiv.org/abs/1804.03793, 2018.
[30] E. Hebey, *Nonlinear Analysis on Manifolds: Sobolev Spaces and Inequalities*, Courant Lecture Notes in Mathematics 5, Courant Institute of Mathematical Sciences, New York University, New York; AMS, Providence, RI, 1999.
[31] 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.
[32] J. M. Lee, *Riemannian Manifolds: An Introduction to Curvature*, Grad. Texts in Math. 176, Springer-Verlag, New York, 1997.
[33] J. Liang and H. Zhao, *Solving partial differential equations on point clouds*, SIAM J. Sci. Comput., 35 (2013), pp. A1461–A1486, https://doi.org/10.1137/120869730.
[34] A. Naga and Z. Zhang, *A posteriori error estimates based on the polynomial preserving recovery*, SIAM J. Numer. Anal., 42 (2004), pp. 1780–1800, https://doi.org/10.1137/S0036142903413002.
[35] A. Naga and Z. Zhang, *The polynomial-preserving recovery for higher order finite element methods in $2D$ and $3D$*, Discrete Contin. Dyn. Syst. Ser. B, 5 (2005), pp. 769–798.
[36] A. Naga and Z. Zhang, *Function value recovery and its application in eigenvalue problems*, SIAM J. Numer. Anal., 50 (2012), pp. 272–286, https://doi.org/10.1137/100797709.
[37] A. Naga, Z. Zhang, and A. Zhou, *Enhancing eigenvalue approximation by gradient recovery*, SIAM J. Sci. Comput., 28 (2006), pp. 1289–1300, https://doi.org/10.1137/050640588.
[38] M. A. Olshanskii and A. Reusken, *A finite element method for surface PDEs: Matrix properties*, Numer. Math., 114 (2010), pp. 491–520.
[39] 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, https://doi.org/10.1137/080717602.
[40] 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.
[41] L. Rineau and M. Yvinec, *$3D$ surface mesh generation*, in CGAL User and Reference Manual, CGAL Editorial Board, 4.9 ed., 2016, http://doc.cgal.org/4.9/Manual/packages.html#PkgSurfaceMesher3Summary.
[42] 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, https://doi.org/10.1137/100785016.
[43] J. Xu and Z. Zhang, *Analysis of recovery type a posteriori error estimators for mildly struc-*

tured grids, Math. Comp., 73 (2004), pp. 1139–1152.
[44] M. Xu, H. Guo, and Q. Zou, *Hessian recovery based finite element methods for the Cahn-Hilliard equation*, J. Comput. Phys., 386 (2019), pp. 524–540.
[45] Z. Zhang and A. Naga, *A new finite element gradient recovery method: Superconvergence property*, SIAM J. Sci. Comput., 26 (2005), pp. 1192–1213, https://doi.org/10.1137/S1064827503402837.
[46] 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.
[47] O. C. Zienkiewicz and J. Z. Zhu, *The superconvergent patch recovery and a posteriori error estimates.* II. *Error estimates and adaptivity*, Internat. J. Numer. Methods Engrg., 33 (1992), pp. 1365–1382.