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 is that superconvergence is guaranteed for PPPR without the symmetric condition which has been asked in the existing techniques. There is also numerical evidence that the superconvergence by PPPR is high curvature stable, which distinguishes itself from the other methods. As an application, we show that 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 make comparisons with state of the art 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 most important methods for numerically solving PDEs, are well established for those problems. A starting point can be traced back to [19], 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 in both analysis and algorithms, see [11–13, 20, 33–35] and the references therein. In the literature, most of the works focus on the a priori error analysis of various surface finite element methods. 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, 9, 10, 13, 14, 18, 37]. Recently, there is an approach proposed in [21] 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 vital role. Gradient recovery techniques, which are important in post-processing solutions or data to improve 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 a stage of maturity, and there is a massive of works in the literature, to name a few [1, 4, 23, 28, 38–41]. We point out some significant methods among them, like the classical Zienkiewicz-Zhu (ZZ) superconvergent patch recovery [40], and a later method called polynomial preserving recovery (PPR) [39]. Those two approaches work under different philosophies in methodology. The former method first locates positions of superconvergent points for the gradient of the finite element solutions 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 fitting in a local patch at each nodal points and then
Gradient recovery methods for data defined on curved spaces have only recently been investigated. In [37], 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 [37] is considered 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 require 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 is 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 with surface elements gives the most competitive results in [37], including several other methods, but their superconvergence is proved with the assumption that the local patch is \(O(h^2)\)–symmetric on the discretized surfaces, just as the planar cases, which is restrictive in applications. However, the condition is not necessary for the PPR method.

This triggers us to generalize the PPR method to the 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., [18], one could consider polynomials locally on the tangent spaces of the manifolds. Apparently, 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 which consists of patches locally parametrized by Euclidean planar domains, but not necessarily sticking their tangent spaces. On these local planar domains, one can use polynomials to fit the unknown surfaces from the given sampling points patch-wisely, as well to approximate the data or finite element solutions iso-parametrically. Our proposed method is thus called parametric polynomial preserving recovery (PPPR) which does not rely on the \(O(h^2)\)–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 problems in [37]. Another advantage of the PPPR method, which will be observed later in a numerical example, is that it is relatively high curvature stable in comparing with the methods proposed in [37]. This is verified by all of our numerical tests on the high curvature surfaces, but a quantitative analysis will be open in the paper. Moreover, the original PPR method [39] does not preserve the function values at the nodal points in its pre-recovery step. In this paper, we take care of this issue, so that 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 exemplary PDE problem. Section 3 introduces discretized function spaces and collects some geometric notati-
ous frequently used in this paper. Section 4 presents the new algorithms especially the PPPR for gradient recovery on manifolds. We also make remarks on the comparison of algorithms and the idea of preserving function values, and provide an argument for its high curvature stable property in recovery. Section 5 gives a brief analysis of the superconvergence properties of the proposed method. Section 6 shows the recovery-based a posteriori estimator by using the PPPR operator. Finally, we present some numerical results and the comparisons with existing methods in Section 7. We postpone 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 [15, 29]. In this paper, we shall consider \((M, g)\) 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 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 \(u: M \to \mathbb{R}\) 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 \(\gamma\) on the manifold

\[
D_u u = \frac{du(\gamma(t))}{dt}
\]

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

The gradient then is an operator such that

\[
(\nabla_g u(x), v(x))_g = D_u u, \quad \text{for all } v(x) \in T_x M \text{ and all } x \in M,
\]

where \(T_x M\) is the tangent space of \(M\) at \(x\). We can think of the gradient as a tangent vector field on the manifold \(M\). In a local coordinate, the gradient has the form

\[
\nabla_g u = \sum_{i,j} g^{ij} \partial_j u \partial_i,
\]

(2.1)

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

\[
(\nabla_g u) \circ r = \nabla \bar{u}(g \circ r)^{-1} \partial r.
\]

In (2.2), \(\bar{u} = u \circ r\) is the pull back of function \(u\) to the local planar parameter domain \(\Omega\), \(\nabla\) denotes the gradient on the planar domain \(\Omega\), \(\partial r\) is the Jacobian of \(r\), and

\[
g \circ r = \partial r(\partial r)^T.
\]

**Remark 2.1.** \(r\) is not specified here, and we will make it clear when it becomes necessary later. We actually have a relation that

\[
(\partial r)^\dagger = (g \circ r)^{-1} \partial r,
\]

(2.3)

where \((\partial r)^\dagger\) denotes the Moore-Penrose inverse of \(\partial r\). See [16, Appendix] for a detailed explanation.
A regular manifold $M$ will be characterized in the paper in the sense that there exist parametrizations $\partial r$ and their inverse $(\partial r)^\dagger$ are functions with bounded norm in the space $W^{3,\infty}$ on every $\Omega$.

Note that the parametrization map $r$ is not unique. Typical ones can be constructed through function graphs which will be used in our later algorithms. We have the following lemma whose proof is given in Appendix A.

**Lemma 2.1.** The gradient (2.2) is invariant under different chosen of regular isomorphic parametrization functions $r$.

Let $\omega = \text{dvol}$ be the volume form on $M$, and $\partial_j$ ($j = 1, \cdots, n$) be the tangential bases. $TM = \bigcup_{x \in M} T_x M$ be the tangent bundle which consists of all the tangent planes $T_x M$ of $M$. For every tangent vector field $v : M \rightarrow TM$, $v = v^i \partial_i$, we have a $(n - 1)$ form defined by the interior product of $v$ and the volume form $\omega$ through the following way

$$i_v \omega = \sum_k \omega(v, \partial_{k1}, \cdots, \partial_{kn-1}),$$

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 $v$ satisfies

$$d(i_v \omega) = \text{div}_g(v) \omega, \quad (2.4)$$

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

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

By equation (2.4), the divergence of the vector field $v$ can be computed by

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

It 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 $\Delta_g$ in our paper, as the divergence of the gradient, that is

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

We would like to mention that if the manifold $M$ is a hyper-surface, that is $M \subset \mathbb{R}^{n+1}$ which has co-dimension 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 $M$ respectively. That is

$$\nabla_g u = (P_T \nabla_e) u_e \text{ and } \text{div}_g v = (P_T \nabla_e) \cdot v_e,$$

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

$$P_T = \text{Id} - \mathbf{n} \otimes \mathbf{n},$$
and \( n \) is a unit normal vector field of \( M \). Such type of definitions has been applied in many references e.g. [37] which consider problems in an ambient space setting.

With the generalized notions of the differentiation on manifolds, the function spaces on manifold domains can be studied analogously to Euclidean domains. Sobolev spaces on manifolds [27] are one of the mostly investigated spaces, which provide a breeding ground to study PDEs. We are interested in numerically approximating PDEs whose solutions are defined on \( M \). Even though our methods are problem independent, in this paper, the analysis will be mainly implemented 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 [19]:

For a given \( f \) satisfying \( \int_M f \, d\text{vol} = 0 \), it is asked to solve the equation

\[
-\Delta_g u = f \quad \text{on} \quad M,
\]

with \( \int_M u \, d\text{vol} = 0 \),

(2.6)

where \( d\text{vol} \) denotes the manifold volume measure.

3. Function Spaces on Discretized Manifolds. The discretization of a smooth manifold \( M \) has been widely studied in many settings, especially in terms of surfaces [20]. 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 [19] by using the linear element. In [12], a generalization of [19] to high order finite element method 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 has 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 3.1 collects some notations frequently referred in the paper.

We consider \( M_h = \bigcup_{j \in J_h} \tau_{h,j} \) a triangular mesh, where \( T_h = \{ \tau_{h,j} \}_{j \in J_h} \) is the set of triangles, and \( h = \max_{j \in J_h} \text{diam}(\tau_{h,j}) \) is the maximum diameter. To better present our main idea, we mostly stick to the simplest case which is the linear finite elements with triangulated surfaces, thus the nodes consist of simply the vertices of \( M_h \), which we denote by \( N_h = \{ x_i \} \in I_h \).

In the following, we define transform operators between the function spaces on \( M \) and on \( M_h \). Let \( V(M) \) and \( V_h(M_h) \) be some ansatz function spaces, define

\[
T_h : V(M) \to V_h(M_h);
\]

\( v \mapsto v \circ P_h \),

(3.1)

and its inverse

\[
(T_h)^{-1} : V_h(M_h) \to V(M);
\]

\( v_h \mapsto v_h \circ P_h^{-1} \),

(3.2)

where \( P_h \) is a continuous and bijective projection map from \( M_h \) to \( M \).

We will use the following definition to characterize the approximation quantity of \( M_h \) to \( M \).

**Definition 3.1.** Let \( M_h = \bigcup_{j \in J_h} \tau_{h,j} \) be a triangular approximation of \( M \). For every \( \tau_{h,j} \), there is a curved triangle face on \( M \), denoted by \( \tau_j \), satisfies \( \bigcup_{j \in I_h} \tau_j = M \).
Table 3.1: Notations

| Notation | Remark |
|----------|--------|
| \((M,g)\) | a smooth, connected, oriented and close manifold with metric \(g\) |
| \((M_h, g_h)\) | a polyhedral approximation of \(M\) with piece-wise smooth metric \(g_h\) |
| \(\mathbf{n}\) | a unit normal vector field on \(M\) |
| \(\nabla_g\) | gradient operator with respect to the metric \(g\) |
| \(\Delta_g\) | Laplace-Beltrami operator with respect to the metric \(g\) |
| \(T_x\) | the tangent space at a position \(x \in M\) |
| \((P_h)^{\pm 1}\) | bijective maps between \(M_h\) and \(M\) |
| \(\mathcal{V}(M)/\mathcal{V}_h(M_h)\) | ansatz function spaces for functions on \(M/M_h\) |
| \((T_h)^{\pm 1}\) | operators map between function spaces on \(M\) and on \(M_h\) |
| \(\Omega\) | a planar domain which locally parametrize a patch of \(M\) |
| \(\zeta\) | a position variable in the parameter domain \(\Omega\) |
| \(\rho(\text{or } r_h)\) | a local parametrization map from \(\Omega\) to a patch of \(M\) (or \(M_h\)) |
| \(\text{vol(}\text{or } \text{vol}_h)\) | the volume (area) measure of \(M\) (or \(M_h\)) |
| \(\|\cdot\|_{k,p,\mathcal{M}}\) | \(W^{k,p}\) norm of functions defined on \(M\) |
| \(\|\cdot\|_{k,p,\mathcal{M}}\) | \(W^{k,p}\) semi-norm of functions defined on \(M\) |
| \(\|\cdot\|_{k,\mathcal{M}}\) | \(H^k\) norm of functions defined on \(M\) |
| \(I_h\) | the total number of the nodal points (vertices) of \(M_h\) |
| \(J_h\) | the total number of the triangles on \(M_h\) |
| \(P^2(\Omega)\) | the \(2\)nd order polynomial space over a planar domain \(\Omega\) |
| \(a \circ b\) | function \(a\) composed with function \(b\) |
| \(\alpha \preceq \beta\) | denotes the inequality \(\alpha \leq C\beta\) where \(C\) is a constant |

Let \(\tau_{h,j}\) and \(\tau_j\) be parametrizable by a common domain \(\Gamma_j\) with \(r_{j,h}\) and \(r_j\) be their parametrization functions respectively. We call \(M_h\) is a regular approximation of \(M\) if

\[
\lim_{h \to 0} \|r_{j,h} - r_j\|_{k,\infty,\Gamma_j} = 0 \quad \text{for all } j \in J_h. \tag{3.3}
\]

for a fixed number \(k \in \mathbb{N}\), and both \(|\partial r_{j,h}|\) and its inverse \(|(\partial r_{j,h})^{-1}|\) are bounded uniformly for all \(j \in J_h\).

If \(M_h\) is a regular approximation of \(M\), then it converges to \(M\) as \(h \to 0\). Here we introduce conditions on the triangle meshes which are common conditions to guarantee the supercloseness (cf. [4, Definition 2.4], [30, Definition 1.2] or [37, Definition 3.2]).

**Definition 3.2.** Suppose \(\tau_h\) and \(\tau'_h\) are two adjacent triangles in \(T_h\), as illustrated in Figure 3.1. They are said to form an \(O(h^2)\) parallelogram if

\[
|AB - CD| = O(h^2), \quad \text{and} \quad |BC - DA| = O(h^2).
\]

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

\[
\sum_{\tau_h \in T_h,2} |\tau_h| = O(h^{2\sigma}).
\]
Before going further, we make a general assumption for a non-adaptive triangulation $M_h$.

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

We have the following lemma for the transform operators.

**Lemma 3.5.** Let $V(M) \hookrightarrow W^{k,p}(M)$ for a fixed $k \in \mathbb{N}$ and $p \geq 1$. Then the transform operators $(T_h)^{\pm 1}$ are uniformly bounded between the spaces $V(M)$ and $V_h(M_h)$ as long as the space $V_h(M_h)$ is compatible with the regularity of $M_h$.

**Proof.** The compatible of the regularity of $M_h$ with $W^{k,p}(M_h)$ makes sure that $V_h(M_h) \hookrightarrow W^{k,p}(M_h)$. For every $v \in V(M)$, denote $\tilde{v}_h := T_h v$. Each triangle $\tau_{h,j}$ in $T_h$ is corresponding to a curved triangle $\tau_j$. If $p = \infty$, every function $v$ and its derivatives are uniformly bounded on $M$, as well for function $\tilde{v}_h$ and its derivatives over $M_h$. Then we can always find constants $c^1_h$ and $C^1_h$ such that

$$c^1_h \| \tilde{v}_h \|_{k,\infty,M_h} \leq \| v \|_{k,\infty,M} \leq C^1_h \| \tilde{v}_h \|_{k,\infty,M_h}.$$ 

If $1 \leq p < \infty$, using the results in [12, page 811], we have the equivalence of $\| v \|_{k,p,\tau_j}$ and $\| \tilde{v}_h \|_{k,p,\tau_{h,j}}$. That is there exists positive and bounded constants $c_{h,j}$ and $C_{h,j}$, such that

$$c^2_{h,j} \| \tilde{v}_h \|_{k,p,\tau_{h,j}} \leq \| v \|_{k,p,\tau_j} \leq C^2_{h,j} \| \tilde{v}_h \|_{k,p,\tau_{h,j}},$$

holds on each pair of the triangular faces. For both the two cases, due to the regular approximation condition in Assumption 3.4, we have $c^1_h \to 1$, $C^1_h \to 1$ when $h \to 0$ as well as $c^2_{h,j} \to 1$ and $C^2_{h,j} \to 1$ when $h \to 0$ for all $j \in J_h$. Thus, both $\{c_{h,j} \}$ and $\{C_{h,j} \}$ are uniformly bounded sequences with respect to the mesh size $h$ and also the index $j$ for $a = 1, 2$. Denote $c := \min_{a,h,j} \{c_{h,j} \}$ and $C := \max_{a,h,j} \{C_{h,j} \}$. Since
\[ \|v\|_{k,p,M}^p = \sum_{j \in J_h} \|v\|_{k,p,\tau_j}^p \] for \( p \in [1, \infty) \), we have the estimates

\[ c \|\tilde{v}_h\|_{k,p,M_h}^p \leq \|v\|_{k,p,M}^p \leq C \|\tilde{v}_h\|_{k,p,M_h}^p , \]

and also

\[ c \|\tilde{v}_h\|_{k,\infty,M_h} \leq \|v\|_{k,\infty,M} \leq \|\tilde{v}_h\|_{k,\infty,M_h} \]

which completes the conclusion. \( \square \)

4. Parametric Polynomial Preserving Recovery on Manifolds. Our developments are based on the PPR method proposed in [39] 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 [24]. 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 here and after.

We will focus on the case where the data is a linear finite element solution on \( M_h \), therefore \( V_h(M_h) \) is restrict to 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 geodesic ball centered at \( x_i \) with geodesic radius \( k \times h_i \), i.e.,

\[ B_{kh_i}(x_i) = \{ x \in N_{h_i} : |x - x_i| \leq k \times h_i \}. \]

Then we define \( B(x_i) = B_{kh_i}(x_i) \) with \( k_i \) be the smallest integer such \( B(x_i) \) satisfies the rank condition (see [39]) 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 (4.1) or \( s_i \) and \( p_i \) in (4.3) and (4.4) 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. Some idea appeared in the literature is to use the tangent space \( T_{x_i} \) at every vertex \( x_i \) as a local parameter domain, and project the neighbouring vertices of \( x_i \) onto this common planar plane, then define polynomials locally by the coordinates of the tangent space. This idea has been applied in [18] and also in [37] to generalize the ZZ method and several other methods. However, the exact manifold \( M \) is usually not given in real problems. Therefore, the tangent spaces \( (T_{x_i})_{x_i \in h} of M \) are blind to users, which makes the idea not much reliable in practice. This problem has also been claimed as an open issue in [37].

As a starting point, we first provide a direct generalization of the PPR method based on given tangent spaces of the exact manifold \( M \). In this case, the algorithm is pretty much the same as the planar one. We sketch it in Algorithm 1 and still name it as the PPR method. We describe the PPPR method in Algorithm 2. In both algorithms, \( I_i \) denotes 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 \( x_i \), for instance, by simple average or weighted average of the normal vectors of each faces adjacent to \( x_i \). However, with such kinds of approximations, the recovery errors will most likely be dominated by
Algorithm 1 PPR Method (with Information of Exact Normal Vectors)

Let the discretized triangular surface $M_h$ and the data (FEM solutions) $(u_{h,i})_{i \in I_h}$ be given. Also, we have the the normal vector $(n_i)_{i \in I_h}$ of $M$ at each vertex $x_i$. Then repeat steps (1) – (3) for all $i \in I_h$:

1. For every $x_i$, select $B(x_i) \in M_h$ including sufficient vertices, and shift $x_i$ to be the origin of $T_{x_i}$, and choose an orthonormal basis $(\tau_1, \tau_2)$ of $T_{x_i}$, then project the vertices $x_j \in B(x_i)$ to $T_{x_i}$ whose new coordinates read as $\zeta_j$.

2. Find a polynomial $p_i$ over $T_{x_i}$ by solving the least squares problem

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

(4.1)

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

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

(4.2)

For the recovery of the gradient $G_{1,h}u_h$ on the whole $M_h$, 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.

the errors of the approximation of the normal vector fields. See also the numerical results in Section 7.

In the following, we shall present another algorithm which is neither relying on the information of the tangent spaces, nor the exact vertices of the surface. Our idea goes to the gradient formulation (2.2), where we can calculate the gradient from an arbitrary local parametrization. It is, in fact, taking an intrinsic point of view on manifolds, which are locally parametrizable by some Euclidean domains $\Omega_i$ but not restrict to $T_{x_i}$ in Algorithm 1. Lemma 2.1 indicates that for every fixed $x_i$, taking arbitrary $\Omega_i$, the gradient operator is analytically invariant. The crucial point is that, numerically, the shape of the triangles must not be destroyed after projecting them to the domain $\Omega_i$, and also for a superconvergence purpose, the $O(h^{2q})$ irregular condition should be properly preserved for the projected triangular mesh on $\Omega_i$. Thus, we still have to find a good way for this projection. In practice, at each vertex, we can use the simple average or weighted average of the surrounding normal vectors to help us to locate and orient a suitable parameter domain $\Omega_i$, and this is what we have adopted in our numerical examples.

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 $\phi_1 = \tau_1, \phi_2 = \tau_2, \phi_3 = n_i$. It is easy to find that the recovered gradient in (4.5) is equal to the one recovered in (4.2). That means Algorithm 2 actually generalizes Algorithm 1. Thus, later on we may do not distinguish the notation $G_h$ for the recovery operator given either by Algorithm 1 or by Algorithm 2. However, we would like to emphasize that the two algorithms are not equivalent if the exact normal vector fields are unknown, in this case, we will use $G_{1,h}$ for the operator in Algorithm 1.

Let $\bar{G}_h$ be the PPR operator introduced in [30] for planar problems, then on every local patch $\Omega$, the PPPR operator $\bar{G}_h$ can be represented by $\bar{G}_h$ in the following sense:

$$(G_h u_h \circ r_{j,h}) = G_h \bar{u}_h ((G_h r_{j,h})^\dagger).$$

(4.6)

Our numerical results will show that the chosen of the approximations of normal
Algorithm 2 PPPR Method (without Asking for Exact Normal Vectors)

Let the discretized triangular surface $M_h$ 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 $B(x_i) \in M_h$ with sufficient vertices, using simple (weighted) average of the out normal vectors of every triangles with vertices in $B(x_i)$, and normalizing the averaged vector to be $\phi^0_i$, and then constructing a local parameter domain $\Omega_i$ orthogonal to $\phi^0_i$. Shift $x_i$ to be the origin of $\Omega_i$, and choose $(\phi^1_i, \phi^2_i)$ the orthonormal basis of $\Omega_i$, then project all selected vertices $x_j \in B(x_i)$ into the parameter domain $\Omega_i$, and record the new coordinates as $\zeta_j$.

2. Reconstruct a 2nd order polynomial surface $S_i$ over $\Omega_i$ to approximate the local surface. Typically, we can approximate it locally as a function graph parametrized by $\Omega_i$. That is $S_i = r_{h,i}(\Omega_i) = \bigcup_{\zeta \in \Omega_i} (\zeta, s_i(\zeta))$, where $s_i$ solves
\[
s_i = \arg \min_s \sum_{j \in I_i} |s(\zeta_j) - \langle x_j, \phi^0_i \rangle|^2 \text{ for } s \in P^2(\Omega_i).
\] (4.3)

3. Find a 2nd order polynomial $p_i$ over the domain $\Omega_i$ by optimizing
\[
p_i = \arg \min_p \sum_{j \in I_i} |p(\zeta_j) - u_{h,j}|^2 \text{ for } p \in P^2(\Omega_i).
\] (4.4)

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 we can approximate the gradient which is given in (2.2). In the local coordinates,
\[
G_{2,h}u(x_i) = (\partial_1 p_i(0,0), \partial_2 p_i(0,0)) \left( \begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right) \begin{pmatrix} \partial_1 s_i(0,0) \\ \partial_2 s_i(0,0) \end{pmatrix}^T \begin{pmatrix} \phi^1_i \\ \phi^2_i \\ \phi^3_i \end{pmatrix}^T.
\] (4.5)

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 $M_h$, 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.

Vectors by either simple average or weighted average has very little influence on the recovery accuracy of the gradient by Algorithm 2, which contrast to the case for Algorithm 1 where the recovery accuracy highly relies on the error of the approximated normal vectors. The nature of Algorithm 2 allows us to apply the analysis of the PPR which has been developed for planar problems. Moreover, the idea of approximating (2.2) by generalizing ZZ scheme seems feasible. One could similarly reconstruct the two levels gradient recovery of the surfaces parametrization function $r$ and the function $\hat{u}$ iso-parametrically. That is to replace the recovery operator $\hat{G}_h$ in (4.6) by using planar ZZ recovery. However, in order to achieve the superconvergence property, this adapted method can never skip the constraint that the meshes should be $O(h^2)$ symmetric.

Remark 4.2. In fact, for both Algorithm 1 and Algorithm 2, we have an al-
ternative way for the polynomial reconstruction instead of the one which is initially proposed in [39]. The method in [39] assumes that a second order polynomial has a form

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

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

\[
A = \begin{pmatrix}
1 & \zeta_{i,1} & \zeta_{i,2} & \zeta_{i,1}^2 & \zeta_{i,1} \zeta_{i,2} & \zeta_{i,2}^2 \\
1 & \zeta_{i,2} & \zeta_{i,2}^2 & \zeta_{i,1} \zeta_{i,2} & \zeta_{i,2}^2 & \zeta_{i,2}^2 \\
\vdots & \ldots & \ldots & \ldots & \ldots & \ldots \\
1 & \zeta_{i,1} \zeta_{i,2} & \zeta_{i,1} \zeta_{i,1}^2 & \zeta_{i,1} \zeta_{i,2}^2 & \zeta_{i,2}^2 & \zeta_{i,2}^2 \\
\end{pmatrix} \text{ and } \mathbf{b} = \begin{pmatrix} u_{h,i_1} \\ u_{h,i_2} \\ \vdots \\ u_{h,i_1} \\ u_{h,i_2} \end{pmatrix}. \tag{4.7}
\]

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

Our observation is that there is some extra freedom can be reduced in the reconstruction of the polynomials. Since the polynomial recovery 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

\[ \tilde{p}(y) = u_{h,i_1} + a_1 y_1 + a_2 y_2 + a_3 y_1^2 + a_4 y_1 y_2 + a_5 y_2^2, \text{ for } y = (y_1, y_2) \in \Omega_i \]

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 \( \Omega_i \), then the matrix and the vector in (4.7) can be simplified to

\[
\tilde{A} = \begin{pmatrix}
\zeta_{i,1} & \zeta_{i,2} & \zeta_{i,1}^2 & \zeta_{i,1} \zeta_{i,2} & \zeta_{i,2}^2 \\
\vdots & \ldots & \ldots & \ldots & \ldots \\
\zeta_{i,1} \zeta_{i,2} & \zeta_{i,1} \zeta_{i,1}^2 & \zeta_{i,1} \zeta_{i,2}^2 & \zeta_{i,2}^2 \\
\end{pmatrix} \text{ and } \tilde{b} = \begin{pmatrix} u_{h,i} - u_{h,i_1} \\ \vdots \\ u_{h,i} - u_{h,i_1} \end{pmatrix}. \tag{4.8}
\]

Solving the problem in the least squares sense

\[ \tilde{a} = (\tilde{A}^T \tilde{A})^{-1} \tilde{A}^T \tilde{b}, \]

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

Using (4.8) instead of (4.7), 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 2nd step of Algorithm 1, and also to replace the polynomial reconstruction in both the 2nd and the 3rd step of Algorithm 2.

Remark 4.3. An experimental observation will be reported later that the PPPR (also the PPR with the exact normal field) seems 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, 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 is very likely not keeping the good structure any more, 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. All of our numerical tests support this hypothesis, as the one showed in Numerical Example 2 in Section 7. However, a quantitative analysis of this property is open for future.

5. Superconvergence Analysis. In the following, we shall show the superconvergence property of the proposed algorithms. Although our algorithms are problem independent, to make the discussion simple, we will take the equation (2.6) as our example, and discuss its approximation by the linear finite element method on triangulated surfaces. The variational formulation of problem (2.6) is given as follows: Find $u \in H^1(M)$ such that

$$\int_M \nabla g_u \cdot \nabla g_v \, dvol = \int_M f v \, dvol \text{ for all } v \in H^1(M). \quad (5.1)$$

The regularity of the solutions has been proved in [2, Chapter 4]. In the finite element methods, the surface $M$ is approximated by the triangulation $M_h$ which satisfy Assumption 3.4, and the solution is simulated in the piecewise linear function spaces $V_h$ defined over $M_h$.

$$\int_{M_h} \nabla g_{v_h} \cdot \nabla g_{v_h} \, dvol_h = \int_{M_h} f_h v_h \, dvol_h \text{ for all } v_h \in V_h(M_h). \quad (5.2)$$

**Lemma 5.1.** Let $G_h$ be the gradient recovery operator by Algorithm 1 or 2, then we have:

1. $G_h$ is a bounded linear operator in the sense that, for every $x_i \in \tau_{h,j} \subset M_h$

$$\|G_h v_h\|_{L^2(\tau_{h,j})} \lesssim \|\nabla g_{v_h}\|_{L^2(\kappa_i)} \text{ for all } i \in I, \; j \in J, \text{ and } v_h \in V_h(M_h),$$

where $\kappa_i \subset M_h$ is the selected triangle patches connected to $x_i$.

2. $G_h$ preserves polynomial in every parametric domain $\Omega_i$ for all $i \in I_h$. If $u \in H^3(M) \cap W^{2,\infty}(M)$, then we have the estimate

$$\|\nabla g u - T_h^{-1} G_h (T_h u)\|_{0, M} \lesssim h^2 (\|u|_{3, M} + \|\sqrt{|g|}\|_{2, \infty}). \quad (5.4)$$

**Proof.** For the first statement, let us denote $\bar{v}_h = v_h \circ r_{j,h}$, and recall (2.2). Then we have that on every triangle,

$$(\nabla_{g_{x_h}} v_h) \circ r_{j,h} = \nabla \bar{v}_h (\partial r_{j,h}) \iff \nabla \bar{v}_h = (\nabla_{g_{x_h}} v_h) \circ r_{j,h} \partial r_{j,h}, \quad (5.5)$$

where $\partial r_{j,h}$ and $(\partial r_{j,h})^\dagger$ are piece-wise constant functions. We take into account the assumptions that $M$ is regular and $C^3$ smooth, and $M_h$ is a regular approximation as specified in Definition 3.1. Then there exist positive constants $c_r$ and $C_r$, such that

$$c_r \leq |(\partial r_{j,h})^\dagger| \leq C_r \text{ and } \frac{1}{C_r} \leq |\partial r_{j,h}| \leq \frac{1}{c_r} \text{ for all } h \text{ and } j. \quad (5.6)$$

Considering the formulation in (4.6) and the boundedness result of the PPR operator on planar domain [30], we have then the following estimate for all triangles,

$$\|G_h v_h\|_{L^2(\tau_{h,j})} \leq C_r \tilde{C} \|\nabla \bar{v}_h\|_{L^2(\Omega_i)}.$$

12
Using the formula on the right of (5.5), and the bounds on $|\partial r_{j,h}|$ in (5.6), we get the boundedness result for $G_h$ in the first statement

$$\|G_h v_h\|_{L^2(\tau_{r,h})} \leq C_{r,v} \|\nabla g_h v_h\|_{L^2(K_h)}.$$ 

For the second statement, the polygonal preserving property on the local patch $\Omega_i$ is a trivial application of [39, Theorem 2.1]. One could directly get the statement of polygonal preserving for the recovery of both the Jacobian of the local planar mapping function $r$ and the gradient of the pulled back function $u \circ r$ on each $\Omega_i$ (One can also refer to [22] for a proof of sharp estimate for PPR). For the estimation in (5.4), we still consider the formulation (2.2) on every local patch. Let $\mathcal{M}_1 \subset \mathcal{M}$ be the area corresponding to $K_i \subset \mathcal{M}_h$.

$$\|\nabla u - T_h^{-1} G_h (T_h u_h)\|_{0,\mathcal{M}_1}^2 = \int_{\Omega_i} |\nabla \bar{u}(\partial r)^t - \nabla \bar{u}_G (\partial r_G)^t|^2 \det(g \circ r)| \quad \text{for all } i \in I_h$$

where $\bar{u}_G$ and $r_G$ are the interpolated recovery functions and surface patches in the algorithms. We have the following inequality

$$\|\nabla u - T_h^{-1} G_h (T_h u_h)\|_{0,\mathcal{M}_1}^2 \leq \int_{\Omega_i} |\nabla \bar{u}(\partial r)^t - \nabla \bar{u}_G (\partial r_G)^t|^2 \det(g \circ r)|$$

$$\leq 2 \int_{\Omega_i} \left( |\nabla \bar{u} - \nabla \bar{u}_G|^2 (|\partial r|^t)^2 + |\nabla \bar{u}_G|^2 (|\partial r_G|^t)^2 + |\det(g \circ r)| \right)$$

$$\leq 2h^4 \left( C_g \|\bar{u}\|_{0,\Omega_i}^2 + C_{u,g} \left\|\sqrt{|g \circ r|}\right\|^2 \right) (5.7)$$

where $C_g$ is a constant depends on $u$, and $C_{u,g}$ is a constant depend on the functions $u$ and $g$. The last inequality above we use the polynomial preserving results for planar functions. We also use the fact that $g \circ r = \partial r(\partial r)^t$, therefore $\left\|\sqrt{|g \circ r|}\right\|_{0,\Omega_i}$ is equivalent to $\|\partial r\|_{0,\Omega_i}$. Since we have assumed $\mathcal{M}$ is $C^3$ smooth and regular, then both $|\partial r|$ and $\det g$ and their derivatives up to second order are uniformly bounded from below and from above. This then allows us to estimate

$$\|\bar{u}\|_{3,\Omega_i} \lesssim \|u\|_{3,\mathcal{M}_1} \quad \text{and} \quad \left\|\sqrt{|g \circ r|}\right\|_{0,\Omega_i} \lesssim \left\|\sqrt{|g|}\right\|_{0,\Omega_i}. (5.8)$$

Combining (5.7) and (5.8) with the compactness of $\mathcal{M}$, we get the final conclusion. □

**Theorem 5.2.** Let Assumption 3.4 hold for $M_h$, and $u \in H^3(\mathcal{M}) \cap W^{2,\infty}(\mathcal{M})$ be the solution of (5.1), and $u_h$ be the solution of (5.2). Then

$$\|\nabla g u - T_h^{-1} G_h u_h\|_{0,\mathcal{M}} \lesssim h^{1+\min\{1,\alpha\}} (\|u\|_{3,\mathcal{M}_1} + \|u\|_{2,\infty,\mathcal{M}_1} + \left\|\sqrt{|g|}\right\|_{0,\Omega_i}) + h^2 f_0. (5.9)$$

**Proof.** Define $u_I$ to be the interpolated function of the real solution $u$ which defined on $\mathcal{M}$. Then $u_I(x_i) \equiv u(x_i)$ at all the vertices of the mesh, and therefore we have

$$G_h(T_h u_I) \equiv G_h(T_h u). (5.10)$$

Using the triangle inequality, we arrive the following estimate

$$\|\nabla g u - T_h^{-1} G_h u_h\|_{0,\mathcal{M}} \leq \|\nabla g u - (T_h)^{-1} G_h (T_h u_I)\|_{0,\mathcal{M}} + \|(T_h)^{-1} G_h (T_h u_I - u_h)\|_{0,\mathcal{M}}.$$
Combining with (5.10), the first term on the right hand side is estimated by (5.4) in Lemma 5.1. For the second term, since both \((T_h)^{-1}\) and \(G_h\) are bounded operators (Lemma 3.5 and Lemma 5.1), we have

\[
\|(T_h)^{-1}G_h(T_hu_I - u_h)\|_{0,M_h} \leq C \|\nabla g_h(T_hu_I - u_h)\|_{0,M_h}
\]

then using the result \(^1\) of [37, Theorem 3.5] to estimate \(\|\nabla g_h(T_hu_I - u_h)\|_{0,M_h}\). These lead to the final estimate. \(\Box\)

Due to Lemma 3.5, we have the following result immediately.

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

\[
\|T_h\nabla g u - G_h u_h\|_{0,M_h} \lesssim h^{1+\min\{1,\sigma\}} (\|u\|_{3,M} + \|u\|_{2,\infty,M} + \|\sqrt{|g|}\|_{2,\infty}) + h^2 \|f\|_{0,M}.
\]

(5.11)

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

\[
\eta_{h,\tau_{h,j}} = \|G_h u_h - \nabla g_h u_h\|_{0,\tau_{h,j}},
\]

and the corresponding global error estimator as

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

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

**Corollary 6.1.** Assume the same conditions in Theorem 5.2 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

\[
\|T_h\nabla g u - \nabla g_h u_h\|_{0,\Omega} \geq C(u)h.
\]

(6.3)

Then it holds that

\[
\left|\frac{\eta_h}{\|T_h\nabla g u - \nabla g_h u_h\|_{0,\Omega}} - 1\right| \lesssim h^{\min\{1,\sigma\}}.
\]

(6.4)

**Proof.** By the triangle inequality, we have

\[
\eta_h \leq \|G_h u_h - T_h\nabla g u\|_{0,\Omega} + \|T_h\nabla g u - \nabla g_h u_h\|_{0,\Omega}.
\]

\(^1\)The \(O(h^{2\sigma})\) condition is asked for the projected triangle meshes on each \(\Omega_h\) in order to show the superconvergence, while what we have assumed in is fact on the meshes before projection as [37]. This gap is bounded by the order \(O(h)\), see also [37]. We argue that for general smooth surfaces with uniformly bounded curvature, using the ways described in our algorithm, the projected shape of meshes will not be significantly changed as [37], therefore the \(O(h^{2\sigma})\) condition can be guaranteed, although this might not be 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 \(O(h^{2\sigma})\) condition may have to be adapted to new index \(\sigma\) according to ratio of the high curvature areas. But in this paper, we skip the quantitative discussion on this point.

14
and hence
\[
\eta_h \left\| T_h \nabla_g u - \nabla_{g_h} u_h \right\|_{0,M_h} - 1 \leq \frac{\|G_h u_h - \nabla_{g_h} u_h\|_{0,M_h}}{\|T_h \nabla_g u - \nabla_{g_h} u_h\|_{0,M_h}} \lesssim h^{\min\{1,\sigma\}}.
\]
where we use the superconvergence result (5.9) and the assumption (6.3) in the last inequality. □

**Remark 6.1.** The assumption (6.3) is common assumption to show the asymptotical exactness of recovery-based a posteriori error estimators as [1, 30, 39]. It is reasonable since that the finite element solution error is not better than the interpolation error which is bounded from below by \(O(h^2)\) (except some trivial cases).

**Remark 6.2.** 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 is to show the superconvergence results of the proposed gradient recovery operators even though the element patch is not \(O(h^2)\)-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 posteriori error estimator introduced in Section 6. Some of our numerical tests are conducted based on MATLAB package iFEM [8]. 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 [36]. To get meshes in other levels, we first perform either the uniform refinement or the newest bisection [7]. Then we project the newest vertices onto the \(M\). In general case, there is no explicit project map available. Hence we adopt the first order approximation of projection map as given in [13]. Thus, the vertices of the meshes are not on the exact surface \(M\) in our test except for the first numerical example.

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 [37], respectively. Note that we use the exact normal vectors for \(G_h^{ZZ}\) in the numerical examples. We denote \(G_{1,h}, G_{2,h}\), and \(G_{1,h}^c\) 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 \(G_{1,h}^c\) in our examples, which are also used to implement Algorithm 2 to construct the local parametric domains \(\Omega_i\). 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 comparisons, we define:

\[
\begin{align*}
D_e &= \|T_h \nabla_g u - \nabla_{g_h} u_h\|_{0,M_h}, \\
D_e^{r_2} &= \|T_h \nabla_g u - G_{1,h} u_h\|_{0,M_h}, \\
D_e^{r_3} &= \|T_h \nabla_g u - G_{1,h}^c u_h\|_{0,M_h}, \\
D_e^{WA} &= \|T_h \nabla_g u - G_h^{WA} u_h\|_{0,M_h}, \\
D_e^{ZZ} &= \|T_h \nabla_g u - G_h^{ZZ} u_h\|_{0,M_h}, \\
D_e^{l} &= \|\nabla_{g_h} u - \nabla_{g_h} u_h\|_{0,M_h}, \\
D_e^{r_2} &= \|T_h \nabla g u - G_{2,h} u_h\|_{0,M_h}, \\
D_e^{r_3} &= \|T_h \nabla g u - G_{2,h}^c u_h\|_{0,M_h}, \\
D_e^{SA} &= \|T_h \nabla g u - G_h^{SA} u_h\|_{0,M_h}, \\
D_e^{ZZ} &= \|T_h \nabla g u - G_h^{ZZ} u_h\|_{0,M_h},
\end{align*}
\]

where \(u_h\) is the finite element solution, \(u\) is the analytical solution and \(u^l\) 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

\[
\begin{align*}
D e_0^{1} &= \|T_h \nabla g u - G_{1,h} u_h\|_{0,\infty, M_h}, \\
D e_0^{2} &= \|T_h \nabla g u - G_{2,h} u_h\|_{0,\infty, M_h}, \\
D e_0^{A} &= \|T_h \nabla g u - G_{h}^{A} u_h\|_{0,\infty, M_h}, \\
D e_0^{WA} &= \|T_h \nabla g u - G_{h}^{WA} u_h\|_{0,\infty, M_h}, \\
D e_0^{ZZ} &= \|T_h \nabla g u - G_{h}^{ZZ} u_h\|_{0,\infty, M_h},
\end{align*}
\]

where \(\|\cdot\|_{0,\infty, M_h}\) means the maximum absolute value at all vertices.

In the following tables, all convergence rates are listed in term of the degree of freedom (DOF). Noticing Dof \(\approx h^{-2}\), the corresponding convergence rates in term of 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

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

To construct a series meshes on torus without \(O(h^2)\) 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 \(O(h)\). Since the generated uniform meshes satisfy the \(O(h^{2\alpha})\) condition, \(O(h^2)\) supercloseness for \(D e_0^{1}\) 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 \(O(h^2)\); while the recovered gradient using PPR with approximated normal field and the other three methods in [37] only converge at the optimal rate \(O(h)\).
Table 7.1: Numerical Results for equation (5.1) on torus surface.

| Dof  | De order | De′ order  | De″ order | De′′ order | De′′′ order |
|------|----------|------------|-----------|------------|-------------|
| 200  | 2.52e+00 | 9.43e-01   | 1.50e+00  | 1.59e+00   | 9.93        |
| 800  | 1.26e+00 | 0.50       | 2.65e-01  | 0.92       | 0.93        |
| 12800| 3.14e-01 | 0.50       | 1.75e-02  | 0.99       | 0.99        |
| 819200| 1.97e-02| 0.50       | 4.40e-03  | 0.97       | 0.97        |

7.2. Numerical Example 2. In this example, we take an emblematical surface [20] which contains high curvature features. It can be represented as the zero level of the following level set function

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

We consider the Laplace-Beltrami equation (2.6) with exact solution \( u = x_1x_2 \). The right-hand side function \( f \) 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 \( D e \) converges at the optimal rate \( O(h) \) and \( D e' \) converges at a superconvergent rate \( O(h^2) \). As demonstrated in [9], some regions of the surface are with significant high curvature. Due to the existence of these areas, only sub-supercorvergence rate of order \( O(h^{1.8}) \) is observed for PPR with approximated normal field and the other three methods in [37]. In contrast, the \( O(h^2) \) superconvergence rate can be observed in the PPR with exact normal field and in the PPPR method. 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 the error function \( |G_{h,2}u_h - T_h\nabla g \cdot u| \) 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 \( O(h^2) \) asymptotically in the discrete maximal norm. This gives the evidence to our statement in Remark 4.3 that PPPR is relatively curvature stable compared to the other methods. At that point, we can say that PPPR is the best one for 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  | $Dc^1$ order | $Dc^2$ order | $Dc^3$ order | $Dc^1$ I order | $Dc^2$ r1 Order | $Dc^3$ r2 order |
|------|---------------|---------------|---------------|----------------|-----------------|----------------|
| 1153 | 5.46e-01      | 2.78e-01      | 4.77e-01      | 2.01e-01       | 1.20e-01        | 3.34e-01       |
| 4606 | 2.85e-01      | 0.47          | 1.16e-01      | 0.62           | 0.62            | 0.69           |
| 18418| 1.40e-01      | 0.51          | 3.45e-02      | 0.89           | 6.58e-02        | 0.81           |
| 73666| 6.97e-02      | 0.50          | 9.86e-03      | 0.90           | 1.97e-02        | 0.87           |
| 1178626| 1.74e-02    | 0.50          | 6.57e-04      | 0.99           | 1.37e-03        | 0.98           |
| 4714498| 8.70e-03     | 0.50          | 1.06e-04      | 0.99           | 3.46e-04        | 0.99           |

| Dof  | $Dc^3$ order | $Dc^3 A$ order | $Dc^3 W$ order | $Dc^3 Z$ order |
|------|--------------|----------------|----------------|----------------|
| 1153 | 4.71e-01      | 4.83e-01       | 4.86e-01       | 4.95e-01       |
| 4606 | 1.98e-01      | 2.26e-01       | 2.30e-01       | 2.18e-01       |
| 18418| 6.63e-02      | 8.30e-02       | 8.59e-02       | 8.70e-02       |
| 73666| 2.06e-02      | 2.69e-02       | 2.82e-02       | 2.33e-02       |
| 1178626| 1.64e-03    | 2.14e-03       | 2.36e-03       | 1.83e-03       |
| 4714498| 4.70e-04     | 6.04e-04       | 6.97e-04       | 5.22e-04       |

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

| Dof  | $Dc^1_0$ order | $Dc^2_0$ order | $Dc^3_0$ order | $Dc^1_0$ I order | $Dc^2_0$ r1 Order | $Dc^3_0$ r2 order |
|------|----------------|---------------|---------------|-----------------|-----------------|----------------|
| 1153 | 9.70e-01       | 7.93e-01      | 8.73e-01      | 4.77e-01        | 4.95e-01        |
| 4606 | 5.43e-01       | 3.26e-01      | 2.50e-01      | 2.18e-01        | 2.33e-01        |
| 18418| 1.92e-01       | 1.09e-01      | 1.06e-01      | 1.83e-01        | 2.33e-01        |
| 73666| 8.57e-02       | 5.18e-02      | 9.16e-02      | 2.33e-01        | 2.33e-01        |
| 1178626| 2.50e-02    | 1.40e-02       | 3.59e-02       | 1.83e-01        |
| 4714498| 1.42e-03     | 9.03e-03       | 7.57e-03       | 1.59e-02        |

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 [9, 13, 14]. We choose the right-hand side function $f$ such that the exact solution in spherical coordinate is given by

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

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

18
To obtain optimal convergence rate, we use adaptive finite element method (AFEM). 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 Dörfler [17] marking strategy with parameter equal to 0.3. Fig 7.3b plots the mesh after the 18 adaptive refinement steps. 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. Also, we observe that the recovery is superconvergent to the exact gradient at a rate of $O(h^2)$.

To test the performance of our new recovery-based a posteriori error estimator for Laplace-Beltrami problem, the effectivity index $\kappa$ is used to measure the quality of an error estimator [1,3], which is defined by the ratio between the estimated error and
Fig. 7.3: Meshes for Example 3: (a) Initial mesh; (b) Adaptively refined mesh.

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

the exact error

\[ \kappa = \frac{\|G_h u_h - \nabla g_h u_h\|_0, M_h}{\|T_h \nabla g u - \nabla g_h u_h\|_0, M_h} \] (7.2)

The effectivity index is plotted in Fig 7.4b. We see that \( \kappa \) 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 [10]:

\[ -\Delta_g u = f, \quad \text{on } \Gamma, \]

where \( \Gamma = \{ x \in \mathbb{R}^3 : (x_1 - x_1^{(0)})^2 + x_2^2 + x_3^2 = 1 \} \). \( f \) is chosen to fit the exact solution

\[ u(x, y, z) = e^{1.85 - (x - 0.2)^2} \sin(y). \]

Note that the solution has an exponential peak. To track this phenomenon, we adopt AFEM with an initial mesh graphed in 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.

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 shown 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.

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

8. Conclusion. In this paper, we have proposed a gradient recovery method which preserves the parametric polynomials for data defined on manifolds. In comparing with existing methods for data on surfaces in the literature, cf. [18, 37], the proposed method has several improvements: The first highlight is that it does not require exact normal vectors of the surfaces, which makes it a realistic and robust method for practical problems; Second, it does not need the element patch to be $O(h^2)$ symmetric to achieve superconvergence. Third, all of our numerical tests show evidences that it is a curvature stable method in comparing with the existing methods. We have evolved the traditional PPR method (for planar problems) to function value preserving at the meantime, and shown the capability of the recovery operator for constructing a posteriori 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 NURBS. However, these are not trivial works, and we leave them for future. Aside from that, the superconvergence result for nodes which do not located on exact manifolds can be numerically observed, but it remains to be theoretically investigated.

Gradient recovery has other applications, like enhancing eigenvalues [25, 31, 32], simplifying higher order discretization of PDEs [21], designing new numerical methods for higher order PDEs [6, 26]. Moreover, it may help for the vector field regularization in the context of [16], where the geometric approximation accuracy is asked to be 1 order higher then the function approximation accuracy for regularizing vector fields on surfaces. The superconvergence property might be able to reduce the additional higher order accuracy on surfaces to have optimal convergence rates. It would be interesting to investigate further the full usage of the PPPR method for problems
with solutions defined on manifolds.

Acknowledgement. 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. The authors thank the referees for their comments and suggestions which significantly improve the paper.

Appendix A. Proof of Lemma 2.1.

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

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

respectively, where $\Omega$ and $\Omega_s$ are planar parameter domains, then there exist

$$ t : \Omega \rightarrow \Omega_s $$

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

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

Then we have

$$ \partial r(\xi) = \partial s(t(\xi)) \partial t(\xi), $$

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

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

we have

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

Fig. 7.6: Numerical Result for Example 4: (a) Errors; (b) Effective index.
Using chain rule on both sides of the former equation of (A.1), then we get
\[ \nabla g \nabla (s(\xi)) \partial s(t(\xi)) \partial t(\xi) = \nabla g \nabla (s(\xi)) \partial s(t(\xi)) = \partial (v \circ s(t(\xi))), \]
which gives the latter equation in (A.1) since \( \partial t(\xi) \) is non-degenerate. Using the same process but consider \( t^{-1} : \Omega \rightarrow \Omega \), we can show the reverse implication. Thus, we have shown that any two arbitrary parameterizations \( \tau \) and \( s \) 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 Yamab e 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] 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.
[7] L. Chen, Short implementation of bisection in MATLAB, in Recent advances in computational sciences, World Sci. Publ., Hackensack, NJ, 2008, pp. 318-332.
[8] ——, iFEM: an innovative finite element methods package in MATLAB, University of California at Irvine, (2009).
[9] A. Y. Chenykh 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.
[10] A. Dedner and P. Madhavan, Adaptive discontinuous Galerkin methods on surfaces, Numer. Math., 132 (2016), pp. 369-398.
[11] A. Dedner, P. Madhavan, and B. Stin n, Analysis of the discontinuous Galerkin method for elliptic problems on surfaces, IMA J. Numer. Anal., 33 (2013), pp. 952-973.
[12] 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.
[13] 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).
[14] 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.
[15] M. P. do Carmo, Riemannian geometry, Mathematics: Theory & Applications, Birkhauser Boston, Inc., Boston, MA, 1992. Translated from the second Portuguese edition by Francis Flaherty.
[16] G. Dong, B. J"{u}ttler, 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.
[17] W. D"{o}rfler, A convergent adaptive algorithm for Poisson’s equation, SIAM J. Numer. Anal., 33 (1996), pp. 1106-1124.
[18] Q. Du and L. Ju, Finite volume methods on spheres and spherical centroidal Voronoi meshes, SIAM J. Numer. Anal., 43 (2005), pp. 1675-1692 (electronic).
[19] 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.
[20] G. Dziuk and C. M. Elliott, Finite element methods for surface PDEs, Acta Numer., 22 (2013), pp. 289-396.
[21] 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.
[22] H. Guo and X. Yang, Polynomial preserving recovery for high frequency wave propagation, J. Sci. Comput., 71 (2017), pp. 594–614.
[23] H. Guo and Z. Zhang, Gradient recovery for the Crouzeix-Raviart element, J. Sci. Comput., 64 (2015), pp. 456–476.
[24] H. Guo, Z. Zhang, and R. Zhao, Hessian recovery for finite element methods, Math. Comp., 86 (2017), pp. 1671–1692.
[25] ———, Superconvergent two-grid methods for elliptic eigenvalue problems, J. Sci. Comput., 70 (2017), pp. 125–148.
[26] 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.
[27] 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.
[28] 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.
[29] J. M. Lee, Riemannian manifolds, vol. 176 of Graduate Texts in Mathematics, Springer-Verlag, New York, 1997. An introduction to curvature.
[30] A. Naga and Z. Zhang, A posteriori error estimates based on the polynomial preserving recovery, SIAM J. Numer. Anal., 42 (2004), pp. 1780–1800 (electronic).
[31] ———, Function value recovery and its application in eigenvalue problems, SIAM J. Numer. Anal., 50 (2012), pp. 272–286.
[32] A. Naga, Z. Zhang, and A. Zhou, Enhancing eigenvalue approximation by gradient recovery, SIAM J. Sci. Comput., 28 (2006), pp. 1289–1300.
[33] M. A. Olshanskii and A. Reusken, A finite element method for surface PDEs: matrix properties, Numer. Math., 114 (2010), pp. 491–520.
[34] 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.
[35] M. A. Olshanskii and D. Sarin, A narrow-band unfitted finite element method for elliptic PDEs posed on surfaces, Math. Comp., 85 (2016), pp. 1549–1570.
[36] L. Rineau and M. Yvinec, 3D surface mesh generation, in CGAL User and Reference Manual, CGAL Editorial Board, 4.9 ed., 2016.
[37] 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.
[38] 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).
[39] Z. Zhang and A. Naga, A new finite element gradient recovery method: superconvergence property, SIAM J. Sci. Comput., 26 (2005), pp. 1192–1213 (electronic).
[40] O. C. Zienkiewicz and J. Z. Zhu, The superconvergent patch recovery and a posteriori error estimates. I. The recovery technique, Internat. J. Numer. Methods Engng., 33 (1992), pp. 1331–1364.
[41] ———, The superconvergent patch recovery and a posteriori error estimates. II. Error estimates and adaptivity, Internat. J. Numer. Methods Engng., 33 (1992), pp. 1365–1382.