# Isogeometric Analysis for Surface PDEs with Extended Loop Subdivision

Qing Pan[a],   Timon Rabczuk[b],   Gang Xu[c],   Chong Chen[d,*]

[a]Key Laboratory of High Performance Computing and Stochastic Information Processing
(Ministry of Education of China), Hunan Normal University, Changsha, China
[b]Institute of Structural Mechanics, Bauhaus Universität-Weimar, Weimar, Germany
[c]Department of Computer Science, Hangzhou Dianzi University, Hangzhou, China
[d]LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of
Sciences, Beijing, China

**Abstract**

We investigate the isogeometric analysis for surface PDEs based on the extended Loop subdivision approach. The basis functions consisting of quartic box-splines corresponding to each subdivided control mesh are utilized to represent the geometry exactly, and construct the solution space for dependent variables as well, which is consistent with the concept of isogeometric analysis. The subdivision process is equivalent to the $h$-refinement of NURBS-based isogeometric analysis. The performance of the proposed method is evaluated by solving various surface PDEs, such as surface Laplace-Beltrami harmonic/biharmonic/triharmonic equations, which are defined on different limit surfaces of the extended Loop subdivision for different initial control meshes. Numerical experiments demonstrate that the proposed method has desirable performance in terms of the accuracy, convergence and computational cost for solving the above surface PDEs defined on both open and closed surfaces. The proposed approach is proved to be second-order accuracy in the sense of $L^2$-norm by theoretical and/or numerical results, which is also outperformed over the standard linear finite element by several numerical comparisons.

**Keywords:** Isogeometric Analysis, Extended Loop Subdivision, Surface PDEs

## 1   Introduction

The isogeometric analysis (IGA) was introduced by Hughes et al. [1, 2] to replace the traditional finite elements by Non-Uniform Rational B-Splines (NURBS) [3, 4] or T-splines [5, 6, 7, 8]. The concept of IGA shows great potential in developing the seamless integration between computer-aided design (CAD) and finite element method (FEM), which possesses higher accuracy than

---

*Corresponding author. E-mail address: chench@lsec.cc.ac.cn

traditional FEM. The $h$-refinement, $p$-refinement and even $k$-refinement can be easily implemented by knot insertion and/or order elevation to improve the simulation accuracy without changing the geometry, which vanish the necessity of subsequent communication with CAD system. From the analytical point of view, hierarchical bases are preferred. Several constructions of such spaces have been proposed, the most common of which are (Truncated) Hierarchical B-Splines (THB) [9], Locally Refined (LR)-Splines [10, 11] and Polynomial/Rational Splines over Hierarchical T-Meshes (PHT/RHT splines) [12, 13, 14, 15, 16]. A review including an efficient IGA code can be found in [17]. A computation framework is reused in IGA on a set of three-dimensional models with similar semantic features in [18].

On the other hand, surface PDEs have been attained extensive attentions in a variety of subjects, such as structural mechanics, biology, fluid dynamics, and image processing, etc. [19, 20, 21, 22, 23, 24, 25, 26, 27]. A comprehensive review about this topic can be founded in [32]. Although the FEMs were developed to solve such kind of problems (see, for instance, [28, 29, 30, 31]), which cannot possess the ability to represent the geometry exactly [2]. To overcome this deficiency, the NURBS-based IGA was proposed to solve the high-order surface PDEs in [32]. In this article, we will further study the subdivision-based IGA, and use this method to solve surface PDEs.

Subdivision is a powerful technique in surface modeling, which provides a simple and efficient method to generate smooth surfaces with arbitrary topology structures. Moreover, it is capable of recovering sharp features of surfaces with creases and corners. Subdivision surfaces play a key role in computer graphics and numerical analysis. A class of piecewise-smooth surface representations was introduced based on subdivision to reconstruct smooth surface from scattered data [33]. Thin-shell finite element analysis was used to describe both the geometry and associated displacement fields [34]. The limit representation of Loop subdivision for triangular meshes was combined with the diffusion model to arrive at a discretized version of the diffusion problem [35]. Mixed finite element methods based on subdivision technology were used to construct high-order smooth surfaces with specified boundary conditions [36]. Truncated hierarchical Catmull-Clark subdivision was developed recently to support local refinement and generalize truncated hierarchical B-splines to arbitrary topology [37].

As a compatible technique of NURBS, subdivision surfaces are capable of the refinability of B-spline techniques. There recently have been a few works on the application of the subdivision-based IGA approach. Volumetric IGA based on Catmull-Clark solids was investigated in [38]. For complex physical domain, Powell-Sabin splines were used as IGA tools for advection-diffusion-reaction problems [39]. The bivariate splines in the rational Bernstein-Bézier form over the triangulation were applied in [40]. A reproducing kernel triangular B-spline-based FEM was proposed to solve PDEs [41]. Collocated isogeometric boundary element methods and unstructured analysis-suitable T-spline surfaces were coupled for linear elastostatic problems in [42]. A new generalized surface and IGA elements having the vertices of the irregular quad mesh through complementing bicubic splines and biquartic splines near irregularities in the mesh layout was presented in [43]. A framework for geometric design and IGA on unstructured quadrilateral meshes was proposed in [44]. A new type of Hermite bases for bicubic spline defined over a rectangular mesh with arbitrary topology was investigated in the framework of

IGA [45]. Second-order PDEs on surfaces with the IGA was considered in [46].

**Contributions**    In the present paper, we introduce an alternative numerical method, namely, the isogeometric analysis based on the extended Loop subdivision approach (IGA-Loop), for solving the PDEs on subdivision surfaces. The finite elements consisting of quartic box-splines corresponding to each subdivided control mesh are utilized to represent the geometry of interest, and construct the solution space for dependent variables as well. The subdivision process is equivalent to the $h$-refinement of NURBS-based isogeometric analysis. The basis functions induced by the extended Loop subdivision possess the ability to represent the geometry exactly which is consistent with the concept of isogeometric analysis. The performance of the proposed method is evaluated by solving various surface PDEs, such as surface Laplace-Beltrami harmonic/biharmonic/triharmonic equations, which are defined on various limit surfaces of the extended Loop subdivision for different initial control meshes. Numerical experiments demonstrate that the proposed method has desirable performance in terms of the accuracy, convergence and computational cost for solving the above classical surface PDEs defined on both open and closed surfaces. The proposed approach is proved to be second-order accuracy in the sense of $L^2$-norm by theoretical and numerical results for the surface Laplace-Beltrami harmonic equation, which is outperformed over the standard linear finite element by several numerical comparisons. Furthermore, the $L^2$-norm error with the second-order convergence rate can be observed for the surface Laplace-Beltrami biharmonic/triharmonic equations.

**Outline**    Let us outline the content of the paper. The required mathematical preliminaries are introduced in Section 2. The isogeometric analysis based on the extended Loop subdivision is investigated in Section 3. We apply the proposed method to solve second/fourth/sixth-order surface PDEs in Section 4. The Section 5 presents several numerical experiments with comparison to the linear finite element method. Finally, the Section 6 concludes the paper.

## 2    Mathematical Preliminaries

Here we introduce some mathematical preliminaries which are requisite for treating the PDEs on surface.

### 2.1    Surface Parameterization

Let $\mathcal{S} = \{\boldsymbol{x}(u,v) \in \mathbb{R}^3 : [u,v]^{\mathrm{T}} \in \Omega \subset \mathbb{R}^2\}$ be a piece of regular smooth parametric surface. The tangent plane and unit normal vector at $\boldsymbol{x}$ of $\mathcal{S}$ are defined as

$$T_{\boldsymbol{x}}\mathcal{S} = \mathrm{span}\{\boldsymbol{x}_u, \boldsymbol{x}_v\} \quad \text{and} \quad \boldsymbol{n} = \frac{\boldsymbol{x}_u \times \boldsymbol{x}_v}{\|\boldsymbol{x}_u \times \boldsymbol{x}_v\|},$$

respectively, where $\boldsymbol{x}_u$ and $\boldsymbol{x}_v$ are the coordinate tangent vectors. The *first fundamental form* of $\mathcal{S}$ is

$$I = \langle \mathrm{d}\boldsymbol{x}, \mathrm{d}\boldsymbol{x} \rangle = g_{11}\mathrm{d}u\mathrm{d}v + 2g_{12}\mathrm{d}u\mathrm{d}v + g_{22}\mathrm{d}u\mathrm{d}v,$$

where the coefficients $g_{\alpha\beta} = \langle \boldsymbol{x}_{u^\alpha}, \boldsymbol{x}_{u^\beta} \rangle$, $\alpha, \beta = 1, 2$ with $(u^1, u^2) = (u, v)$. The *second fundamental form* of $\mathcal{S}$ is

$$II = -\langle \mathrm{d}\boldsymbol{x}, \mathrm{d}\boldsymbol{n} \rangle = b_{11}\mathrm{d}u\mathrm{d}u + 2b_{12}\mathrm{d}u\mathrm{d}v + b_{22}\mathrm{d}v\mathrm{d}v$$

with the coefficients $b_{\alpha\beta} = \langle \boldsymbol{x}_{u^\alpha u^\beta}, \boldsymbol{n} \rangle = -\langle \boldsymbol{x}_{u^\alpha}, \boldsymbol{n}_{u^\beta} \rangle$. Set

$$g = \det[g_{\alpha\beta}], \quad [g^{\alpha\beta}] = [g_{\alpha\beta}]^{-1} \quad \text{and} \quad b = \det[b_{\alpha\beta}] \quad \text{for} \quad \alpha, \beta = 1, 2.$$

**Area**   The positive value

$$\int_{\mathcal{S}} \mathrm{d}A = \iint_{\Omega} |\boldsymbol{x}_u \wedge \boldsymbol{x}_v| \mathrm{d}u\mathrm{d}v \tag{2.1}$$

is called the *area* of surface $\mathcal{S}$.

**Mean Curvature**   First we introduce Weingarten transformation $\mathcal{W} : T_{\boldsymbol{x}}\mathcal{S} \to T_{\boldsymbol{x}}\mathcal{S}$ which is a linear mapping, and satisfies

$$\mathcal{W}(\boldsymbol{x}_u) = -\boldsymbol{n}_u \quad \text{and} \quad \mathcal{W}(\boldsymbol{x}_v) = -\boldsymbol{n}_v.$$

The linear mapping can be represented by a $2 \times 2$ matrix $S = [b_{\alpha\beta}][g^{\alpha\beta}]$. The eigenvalues $k_1$ and $k_2$ of $S$ are the principal curvatures of $\mathcal{S}$. Their arithmetic average is the *mean curvature*:

$$H := \frac{k_1 + k_2}{2} = \frac{\mathrm{tr}(S)}{2} = \frac{b_{11}g_{22} - 2b_{12}g_{12} + b_{22}g_{11}}{2g}.$$

## 2.2   Surface Differential Operators

Let $C^1(\mathcal{S})$ denote the set of function $f : \mathcal{S} \to \mathbb{R}$ with continuous derivatives at any $\boldsymbol{x} \in \mathcal{S}$. Analogously, we can define the sets $C^2(\mathcal{S})$, $C^3(\mathcal{S})$, …... For the sufficiently differentiable $f(\boldsymbol{x}) = f(\boldsymbol{x}(u, v))$ defined on surface $\mathcal{S}$, we denote

$$f_u = \frac{\partial f(\boldsymbol{x}(u, v))}{\partial u} \quad \text{and} \quad f_v = \frac{\partial f(\boldsymbol{x}(u, v))}{\partial v}.$$

For simplicity, we set $f_1 := f_u$, $f_2 := f_v$, $f_{11} := f_{uu}$, $f_{12} := f_{uv}$ and $f_{22} := f_{vv}$.

**Tangential Gradient Operator**   Let $f \in C^1(\mathcal{S})$, then the *tangential gradient operator* $\nabla_{\mathcal{S}}$ acting on $f$ at $\boldsymbol{x} \in \mathcal{S}$ is defined as

$$\nabla_{\mathcal{S}} f(\boldsymbol{x}) = [\boldsymbol{x}_u, \boldsymbol{x}_v][g^{\alpha\beta}][f_u, f_v]^{\mathrm{T}} \in \mathbb{R}^3. \tag{2.2}$$

By the definition of $g^{\alpha\beta}$ we have

$$\nabla f(\boldsymbol{x}) = [f_u, f_v]^{\mathrm{T}} = [\boldsymbol{x}_u, \boldsymbol{x}_v]^{\mathrm{T}} \nabla_{\mathcal{S}} f. \tag{2.3}$$

Obviously, $\nabla_{\mathcal{S}} f \in T_{\boldsymbol{x}}\mathcal{S}$ and hence $\langle \nabla_{\mathcal{S}} f(\boldsymbol{x}), \boldsymbol{n} \rangle = 0$.

**Divergence Operator**  Let $\boldsymbol{v}$ be a $C^1$ smooth vector field on surface $\mathcal{S}$. Then the *divergence operator* $\mathrm{div}_{\mathcal{S}}$ acting on $\boldsymbol{v}$ is defined as

$$\mathrm{div}_{\mathcal{S}}(\boldsymbol{v}) = \frac{1}{\sqrt{g}} \left[ \frac{\partial}{\partial u}, \frac{\partial}{\partial v} \right] \left[ \sqrt{g}[g^{\alpha\beta}][\boldsymbol{x}_u, \boldsymbol{x}_v]^{\mathrm{T}} \boldsymbol{v} \right].$$

**Laplace-Beltrami Operator**  Let $f \in C^2(\mathcal{S})$, then $\nabla_{\mathcal{S}} f$ is a smooth vector field on surface $\mathcal{S}$. The *Laplace-Beltrami operator* (LBO) $\Delta_s$ acting on $f$ is defined as

$$\Delta_{\mathcal{S}} f = \mathrm{div}_{\mathcal{S}}(\nabla_{\mathcal{S}} f).$$

With the definitions of $\Delta_{\mathcal{S}}$ and $\mathrm{div}_{\mathcal{S}}$, we derive

$$\Delta_{\mathcal{S}} f = \frac{1}{g}(g_{22} f_{11} + g_{11} f_{22} - 2g_{12} f_{12}).$$

## 2.3  Global Properties of Surface

Here we introduce the concept of the global surface, and derive some global properties, for example *Green formula*. There will be quite useful for the numerical analysis of surface PDEs.

**Definition 2.1**  *The $\mathcal{S} = \bigcup\limits_{\alpha \in \Lambda} \mathcal{S}_{\alpha}$ is called a global orientable surface in $\mathbb{R}^3$ if it satisfies:*

*(1) Each $\mathcal{S}_{\alpha} = \{\boldsymbol{x}_{\alpha}(u,v) \in \mathbb{R}^3 : [u,v]^{\mathrm{T}} \in \Omega_{\alpha} \subset \mathbb{R}^2\}$ is a regular surface in $\mathbb{R}^3$.*

*(2) For any $\alpha, \beta \in \Lambda$, if $\mathcal{S}_{\alpha} \cap \mathcal{S}_{\beta} \neq \varnothing$, $\mathcal{S}_{\alpha} \cap \mathcal{S}_{\beta}$ is still a piece of surface and the composite mapping*

$$\boldsymbol{x}_{\beta}^{-1} \circ \boldsymbol{x}_{\alpha} : \Omega_{\alpha\beta} = \boldsymbol{x}_{\alpha}^{-1}(\Omega_{\alpha} \cap \Omega_{\beta}) \subset \Omega_{\alpha} \longrightarrow \Omega_{\beta\alpha} = \boldsymbol{x}_{\beta}^{-1}(\Omega_{\alpha} \cap \Omega_{\beta}) \subset \Omega_{\beta}$$

*is differentiable.*

*(3) For every $\mathcal{S}_{\alpha}$, there is an orientation $\boldsymbol{n}_{\alpha}$. Moreover $\boldsymbol{n}_{\alpha} = \boldsymbol{n}_{\beta}$ for $\boldsymbol{x} \in \mathcal{S}_{\alpha} \cap \mathcal{S}_{\beta}$.*

**Theorem 2.1 (Green formula for LBO)**  *Let $\mathcal{S}$ be an orientable surface, and $\Omega$ be a subregion of $\mathcal{S}$ with a piecewise smooth boundary $\partial\Omega$. Let $\boldsymbol{n}_c$ be the outward unit normal along the boundary $\partial\Omega$. Then for a given smooth vector field $\boldsymbol{v} \in C^1(\Omega)$, we have*

$$\int_{\Omega} \langle \boldsymbol{v}, \nabla_s f \rangle + f \, \mathrm{div}_{\mathcal{S}}(\boldsymbol{v}) \mathrm{d}A = \int_{\partial\Omega} f \langle \boldsymbol{v}, \boldsymbol{n}_c \rangle \mathrm{d}s. \tag{2.4}$$

## 2.4 Sobolev Spaces on Surface

The FEM is applied to solve PDEs on surface, therefore the theory of Sobolev spaces on surface is requisite. Assume that $\mathcal{S}$ is a sufficiently smooth surface. For a given constant $k$ and a function $f \in C^\infty(\mathcal{S})$, denote $\nabla^k f$ the $k$-th order covariant derivative of function $f$, with the convention $\nabla^0 f = f$. Let

$$C_k(\mathcal{S}) = \left\{ f \in C^\infty(\mathcal{S}) : \int_{\mathcal{S}} |\nabla^j f|^2 \mathrm{d}A \leq \infty \text{ for } j = 0, \ldots, k \right\}.$$

We have the following definition of Sobolev space $H^k(\mathcal{S})$.

**Definition 2.2** *Let $\mathcal{S}$ be a compact surface with at least $k$-th order smoothness. Sobolev space $H^k(\mathcal{S})$ is the completion of $C_k(\mathcal{S})$ in the sense of norm*

$$\|f\|_{H^k(\mathcal{S})} := \left( \sum_{j=0}^{k} \int_{\mathcal{S}} |\nabla^j f|^2 \mathrm{d}A \right)^{\frac{1}{2}}. \tag{2.5}$$

For the compact surface $\mathcal{S}$, we have

$$C_k(\mathcal{S}) = C^\infty(\mathcal{S}) \subset C^k(\mathcal{S}) \subset H^k(\mathcal{S}).$$

# 3 Isogeometric Analysis Based on Extended Loop Subdivision

Isogeometric analysis is a recently developed computational approach that the solution space for dependent variables is represented in terms of the same functions which represent the geometry. The isogeometric procedures can be developed based on NURBS, A-patches or subdivision surfaces [2]. In this work, we investigate isogeometric analysis for surface PDEs based on subdivision surfaces.

A subdivision surface is a method of representing a smooth surface, which can be generated through a progressively iterative subdivision process starting from an initial control mesh by a prescribed subdivision scheme. The subdivision schemes used to generate smooth surfaces can be separated into two categories: interpolatory and approximatory. In the first category, the vertex positions of the initial mesh are fixed, and only the positions of new added vertices need to be computed in each subdivision step (see [47, 48]). However, for the second category, both the old and new vertex positions are required to update during each refinement (see [49, 50]). In general, the surfaces generated by approximating schemes have better quality than those generated by interpolating ones. In this article, the extended Loop subdivision is considered, which falls into the second category.

## 3.1 Extended Loop Subdivision Scheme

Let $\Omega_h^0$ be the input closed triangular mesh, which denotes as the initial control mesh of the conventional Loop subdivision. In each refinement step, each triangle is subdivided into four

sub-triangles in which all the new vertices are generated by the weighted average of the old vertices, and all the old vertex positions on the refined mesh $\Omega_h^{k+1}$ are computed by the weighted average of the vertex positions on the mesh $\Omega_h^k$. Let $\boldsymbol{p}_0^k$ be a vertex with one-ring neighbors $\boldsymbol{p}_j^k$ $(j = 1, \ldots, n)$ on $\Omega_h^k$, where $n$ is the valence of $\boldsymbol{p}_0^k$. The positions of the new generated vertices $\boldsymbol{p}_i^{k+1}$ on $\Omega_h^{k+1}$ corresponding to the edges of the previous mesh are computed by

$$\boldsymbol{p}_i^{k+1} = \frac{1}{8}(3\boldsymbol{p}_0^k + 3\boldsymbol{p}_i^k + \boldsymbol{p}_{i-1}^k + \boldsymbol{p}_{i+1}^k), \quad i = 1, \ldots, n, \tag{3.1}$$

where the subscript index $i$ is regarded as modulo $n$. The old vertex gets new position according to the rule

$$\boldsymbol{p}_0^{k+1} = (1 - n\alpha)\boldsymbol{p}_0^k + \alpha \sum_{j=1}^{n} \boldsymbol{p}_j^k, \tag{3.2}$$

where

$$\alpha = \frac{1}{n}\left[\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{n}\right)^2\right] \tag{3.3}$$

proposed by Loop in [50] is named as *Loop's weight* [52].

The conventional Loop subdivision scheme is suitable only for subdividing closed triangular control meshes without boundaries. Actually, for a lot of geometric modeling problems, the surfaces are usually constructed in a piecewise manner with fixed boundaries. In such a case, Loop subdivision scheme cannot be implemented near the boundaries of the control mesh. To overcome this deficiency, the extended Loop subdivision scheme was proposed by Biermann et al. in [51], which can treat triangular control meshes with boundaries. Note that the subdivision rule for boundaries is the same as that of cubic B-spline, and the control vertices on boundaries are treated as the control vertices of cubic B-spline curves with spaced knots. The extended Loop subdivision scheme is described as follows.

### 3.1.1 Vertex Refinement

The vertices can be separated into three types: corner vertex, boundary vertex and interior vertex. For different type of vertex, the refined strategy is given as:

(1) Corner vertex: The corner vertices are to be interpolated, meaning which are fixed.

(2) Boundary vertex: Let $\boldsymbol{x}_i$ be a boundary vertex, and let $\boldsymbol{x}_l$ and $\boldsymbol{x}_r$ be its two neighbor vertices on the boundary, then $\boldsymbol{x}_i$ is updated by $\frac{1}{8}\boldsymbol{x}_l + \frac{3}{4}\boldsymbol{x}_i + \frac{1}{8}\boldsymbol{x}_r$.

(3) Interior vertex: Use the conventional Loop scheme as (3.1) and (3.2).

### 3.1.2 Edge Refinement

Overall, we divide the edges into boundary, sub-boundary and interior edges. Boundary edges lie on the boundaries, which are the features of control mesh in general. Sub-boundary edges are not boundary edges but adjacent to the boundary vertices. The rest are the interior edges. For different type of edge, the edge refinement is performed as:

(1) Boundary edge: Let $[\boldsymbol{x}_l\boldsymbol{x}_r]$ be a boundary edge. The newly added vertex on this edge is the average of $\boldsymbol{x}_l$ and $\boldsymbol{x}_r$, i.e., $\frac{1}{2}(\boldsymbol{x}_l + \boldsymbol{x}_r)$.

(2) Sub-boundary edge: Let $[\boldsymbol{x}_i\boldsymbol{x}_j]$ be a sub-boundary edge with $\boldsymbol{x}_i$ being a boundary vertex, $\boldsymbol{x}_l$ and $\boldsymbol{x}_r$ being the two wing neighbor vertices of $[\boldsymbol{x}_i\boldsymbol{x}_j]$, then the newly added vertex on this edge is defined as $(\frac{3}{4} - \gamma)\boldsymbol{x}_i + \gamma\boldsymbol{x}_j + \frac{1}{8}\boldsymbol{x}_l + \frac{1}{8}\boldsymbol{x}_r$ where $\gamma = \frac{1}{2} - \frac{1}{4}\cos\theta_k$ with $\theta_k = \frac{\pi}{k}$ for a boundary point, $\theta_k = \frac{\alpha}{k}$ for a convex corner point, and $\theta_k = \frac{2\pi - \alpha}{k}$ for a concave corner point.

(3) Interior edges: Use the conventional Loop scheme as (3.1) and (3.2).

## 3.2 Limit Form of the Extended Loop Subdivision

The explicit representation for the limit position of each vertex can be derived, which is stated as the following lemma.

**Lemma 3.1 ([50])** *Let $\boldsymbol{p}_0^0$ be a control vertex of valence $n$ on the mesh $\Omega_h^0$, and $\boldsymbol{p}_i^0$, $i = 1, \ldots, n$, be its one-ring neighbor control vertices. The vertices sequence $\{\boldsymbol{p}_0^k\}$ converges to a unique point*

$$\boldsymbol{p}_0^\infty = (1 - nl)\boldsymbol{p}_0^0 + l\sum_{j=1}^{n}\boldsymbol{p}_j^0, \quad l = \frac{1}{n + 3/(8\alpha)}, \tag{3.4}$$

*as the subdivision step $k \to \infty$, where the $\alpha$ is defined by (3.3).*

The limit surface of the extended Loop subdivision is $C^2$ everywhere except at the irregular vertices (i.e., having valence other than six) where is $C^1$ (see [53]). By Lemma (3.1), we can evaluate the position of the limit surface at any finite subdivision level and at any vertex by weighted averaging the vertex and its neighbors. However, it is not feasible to compute any point on the limit surface. Fortunately, there exists the explicit expression for the regular Loop subdivision surface patch. If all vertices of the triangle have valence six and none of its two-ring neighbor vertices is a boundary vertex, the resulting surface patch is called *regular*. The regular patch can be exactly described by a quartic box-spline, more precisely, which is formulated by the linear combination of 12 basis functions:

$$\boldsymbol{p}(\xi, \eta) = \sum_{i=1}^{12} B_i(\xi, \eta)\boldsymbol{p}_i^k, \tag{3.5}$$

where $(1 - \xi - \eta, \xi, \eta)$ are the barycentric coordinates of the shaded unit reference triangle and $\boldsymbol{p}_i^k$ are the indexed vertices of the control mesh in Figure 1 (a). The analytic expressions of the basis functions $B_i$ are given as those in [52]. Each triangle of the control mesh can be regarded as a parametric domain, which corresponds to one triangular patch of the limit surface.

If a triangle is *irregular*, i.e., at least one of its vertices has a valence other than six or one of its two-ring neighbor vertices is a boundary vertex, the resulting surface patch cannot be represented by a quartic box-spline. For evaluating irregular patches, we use the fast scheme proposed by Stam in [56], in which the mesh needs to be subdivided repeatedly until the parameter values of interest are interior to a regular patch (See Figure 1 (b)). Hence, the basis functions given in (3.5) can be used to represent the geometry exactly.
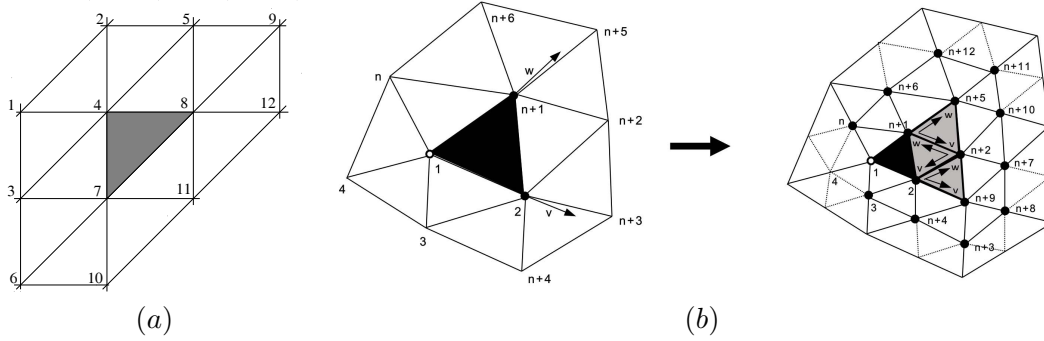


$(a)$                 $(b)$

Fig 1: $(a)$ A single regular triangular patch defined by 12 control vertices. $(b)$ Subdividing an irregular patch once generates three sub-patches and enough control vertices for evaluating three shaded sub-patches. Here the vertex labelled '1' is extraordinary of valence 5.

## 3.3 Finite Element Space for IGA-Loop

Let $\mathcal{S}$ be a considered smooth surface, which is generated by the limit of the extended Loop subdivision scheme. And we denote $\Omega_h^k$ the $k$-th refined control mesh of the discretized representation $\mathcal{S}_h$, where $\boldsymbol{p}_i^k$ denotes its $i$-th control vertex and $h$ is the length of maximal edge. Each control vertex $\boldsymbol{p}_i^k$ on $\Omega_h^k$ can be evaluated recursively, which is described in Section 3.1. Each triangle patch of $\Omega_h^k$ can be locally represented by a linear combination of explicit box-spline tiles as (3.5). The boundaries of $\mathcal{S}_h$ are represented as the cubic B-spline curves which are preserved as the subdivision proceeds. The subdivision step is equivalent to the $h$-refinement of the NURBS-based isogeometric analysis.

Let us define the basis functions of the finite element space for IGA-Loop. For each control vertex $\boldsymbol{p}_i^k$ on the control mesh $\Omega_h^k$ of the surface $\mathcal{S}_h$, including boundary control vertices, we associate it with a basis function $\phi_i$, where $\phi_i$ is defined as the limit of the extended Loop subdivision scheme applying to the zero control values everywhere except at $\boldsymbol{p}_i^k$ where it is one.

9

Note that the basis functions have a compact support within two-ring neighbors. The mesh $\Omega_h^k$, formulated as piecewise patches, is served as the parametric domain of the basis functions. Actually the set of quartic box-splines corresponding to each subdivided computable control patch are utilized to represent the geometry of interest and the solution space for dependent variables.

The control mesh $\Omega_h^k$, as a piecewise linear surface, is served as the definition domain of the basis function $\phi_i$. The mapping from $\Omega_h^k$ to $\phi_i$ is defined by a dual subdivision process. More precisely, when the extended Loop subdivision scheme is applied to the control function values recursively, the linear subdivision scheme (each triangle is partitioned into four equal-sized sub-triangles) is applied to the control mesh correspondingly. The limit of the former is $\phi_i$ and that of later is $\Omega_h^k$ itself.

The basis functions share some properties with the well-known B-spline basis. These properties are important in the proposed method. Now let us describe them as follows.

(1) *Positivity.* The weights of the extended subdivision rules are positive. Hence the basis function $\phi_i$ is nonnegative everywhere and positive around $\boldsymbol{p}_i^k$.

(2) *Locality.* It is known that the limit value at a control vertex is a linear combination of the one-ring neighbor values. Hence, the limit value is zero at a control vertex if the control values on the one-ring neighbor control vertices are zero. Therefore, the support of the basis function is within the two-ring neighborhood.

(3) *Partition of Unity.* Since all the subdivision rules have the properties that the weights are summed to one. Therefore, if we choose all the control values as one. The control values after one subdivision step are still one. This implies that $\sum \phi_i = 1$. This property is called partition of unity.

(4) *Interpolatory Properties at the Boundary.* The extended subdivision rules on the boundary do not involve the interior control vertices. Hence the basis functions for the interior control vertices are zero at the boundary.

(5) *Tangential Property.* Let $\boldsymbol{p}_i^k$ be a control vertex, with non of its one-ring neighbor control vertices are boundary control vertices. Then $\nabla_s \phi_i$ vanishes on the boundary. This fact can be observed by considering the eigen-decomposition of the control vertices. Let $\boldsymbol{x}^k \in \mathbb{R}^{(n+1)\times 3}$ be a vector consisting of one-ring neighbor control vertices of $\boldsymbol{p}_i^k$ at the subdivision level $k$, $S \in \mathbb{R}^{(n+1)\times(n+1)}$ be the local subdivision matrix that converts $\boldsymbol{x}^k$ to $\boldsymbol{x}^{k+1}$, i.e.,

$$\boldsymbol{x}^{k+1} = S\boldsymbol{x}^k = S^k\boldsymbol{x}^1, \quad k = 1, 2, \ldots.$$

Here $n$ stands for the valence of $\boldsymbol{p}_i^k$. Suppose $\boldsymbol{x}^1$ is decomposed into

$$\boldsymbol{x}^1 = \boldsymbol{e}_0\boldsymbol{a}_0^{\mathrm{T}} + \boldsymbol{e}_1\boldsymbol{a}_1^{\mathrm{T}} + \boldsymbol{e}_2\boldsymbol{a}_2^{\mathrm{T}} + \cdots + \boldsymbol{e}_n\boldsymbol{a}_n^{\mathrm{T}}, \quad \boldsymbol{a}_j \in \mathbb{R}^3,$$

where $\boldsymbol{e}_0$, $\boldsymbol{e}_1$, …, $\boldsymbol{e}_n$ are the eigenvectors of $S$. Here we assume that these eigenvectors are arranged in the order of non-increasing eigenvalues $\lambda_j$. Then

$$\boldsymbol{x}^{k+1} = \lambda_0^k\boldsymbol{e}_0\boldsymbol{a}_0^{\mathrm{T}} + \lambda_1^k\boldsymbol{e}_1\boldsymbol{a}_1^{\mathrm{T}} + \lambda_2^k\boldsymbol{e}_2\boldsymbol{a}_2^{\mathrm{T}} + \cdots + \lambda_n^n\boldsymbol{e}_n\boldsymbol{a}_n^{\mathrm{T}},$$

where $\lambda_0 = 1$, $\lambda_1 = \lambda_2 < 1$. It is well-known that the limit position at the center is $\boldsymbol{a}_0$. The tangent direction at this vertex are $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$, and $\boldsymbol{a}_j$ is given by $\boldsymbol{a}_j^{\mathrm{T}} = \tilde{\boldsymbol{e}}_j^{\mathrm{T}} \boldsymbol{x}^1$. $\tilde{\boldsymbol{e}}_j$ are the left eigenvectors of $S$ with normalized condition $\tilde{\boldsymbol{e}}_j^{\mathrm{T}} \boldsymbol{e}_j = 1$. The analysis above is valid for control function values. The fact that $\nabla_{\mathcal{S}} \phi_i$ vanishes on the boundary implies that the tangent vector of the subdivision surface on the boundary determined by the boundary and sub-boundary control vertices. This is similar to Bézier and B-spline surfaces.

(6) *Linear independency.* The functions $\phi_i$, $i = 0, \ldots, m$, are defined by the extended Loop subdivision scheme. This fact can be derived from the unique solvability of the following interpolation problem:

**Lemma 3.2 ([54])** *Let $\boldsymbol{p}_i^{\infty}$ be the limit position of the control vertex $\boldsymbol{p}_i^k$ with its 1-ring neighbor vertices $\boldsymbol{p}_j^k, j = 1, 2, \ldots, n_i$, $u(\boldsymbol{p}_i^{\infty})$ be the i-th interpolation function value, and $\mathrm{u}(\boldsymbol{p}_i^k)$ be the i-th control function value, where $i = 1, \ldots, m$. The system*

$$(1 - n_i l_i)\mathrm{u}(\boldsymbol{p}_i^k) + l_i \sum_{j=1}^{n_i} \mathrm{u}(\boldsymbol{p}_j^k) = u(\boldsymbol{p}_i^{\infty}), \quad i = 1, \ldots, m,$$

*is always solvable uniquely. Here the $l_i$ is defined as (3.4).*

(7) *Interpolant Error.* With the basis of Lemma 3.2, we derive the following interpolant error estimation:

**Lemma 3.3 ([55])** *Assume that $\Omega_h^k$ is the k-th refined control mesh for the smooth surface $\mathcal{S}$ generated by the extended Loop subdivision, where h stands for the length of the maximal edge. For $u \in H^2(\mathcal{S})$, there exists an interpolation function $u_I \in H^2(\mathcal{S}_h)$ such that*

$$\|u - u_I\|_{H^l(\mathcal{S})} \leq ch^{2-l}\|u\|_{H^2(\mathcal{S})}, \quad l = 0, \ 1. \tag{3.6}$$

# 4 Applications to Surface PDEs

We consider the approximations for the Laplace-Beltrami harmonic, biharmonic and triharmonic problems on different surfaces by the proposed IGA-Loop strategy. The numerical examples we provide in Section 5 include both open and closed surfaces, more concretely, which are the Laplace-Beltrami harmonic problem on a quarter of cylinder and one-eighth of sphere, the Laplace-Beltrami biharmonic problem on a cylinder, and the Laplace-Beltrami triharmonic problem on a unit sphere. The numerical error will be analyzed which shows the consistent convergence rate with (4.11). For comparison, we also consider the approximations of these three problems by standard linear element discretization. Hence we adopt the mixed formulations of the high-order Laplace-Beltrami biharmonic and triharmonic problems.

### 4.1 Surface PDEs

**Laplace-Beltrami Harmonic Problem**   The Laplace-Beltrami harmonic equation with zero boundary condition can be written as

$$\begin{cases} -\Delta_{\mathcal{S}} u = f, & \text{in } \mathcal{S}, \\ u = 0, & \text{on } \partial\mathcal{S}, \end{cases} \tag{4.1}$$

where the boundary $\partial\mathcal{S}$ of surface $\mathcal{S}$ is Lipschitz continuous. Here $f : \mathcal{S} \to \mathbb{R}$ is a given sufficiently regular function, and $\Delta_{\mathcal{S}}$ is the Laplace-Beltrami operator. The weak form of (4.1) is given as follows:

$$\begin{cases} \text{Find } u \in H_0^1(\mathcal{S}) \text{ such that} \\ \displaystyle\int_{\mathcal{S}} \nabla_{\mathcal{S}} u \cdot \nabla_{\mathcal{S}} \varphi \ \mathrm{d}A = \int_{\mathcal{S}} f\varphi \ \mathrm{d}A, \quad \forall \varphi \in H_0^1(\mathcal{S}). \end{cases} \tag{4.2}$$

**Laplace-Beltrami Biharmonic Problem**   The Laplace-Beltrami biharmonic equation with homogeneous boundary conditions reads as

$$\begin{cases} \Delta_{\mathcal{S}}^2 u = f, & \text{in } \mathcal{S}, \\ u = \frac{\partial u}{\partial \boldsymbol{n}} = 0, & \text{on } \partial\mathcal{S}, \end{cases} \tag{4.3}$$

where $\boldsymbol{n}$ is the outward directed unit vector normal to the boundary $\partial\mathcal{S}$. Let $\Delta_{\mathcal{S}} u = -v$, then the mixed weak formulation of (4.3) reads as

$$\begin{cases} \text{Find } (u, v) \in H_0^1(\mathcal{S}) \times H^1(\mathcal{S}) \text{ such that} \\ \displaystyle\int_{\mathcal{S}} \nabla_{\mathcal{S}} u \cdot \nabla_{\mathcal{S}} \varphi \ \mathrm{d}A - \int_{\mathcal{S}} v\varphi \ \mathrm{d}A = 0, \quad \forall \varphi \in H_0^1(\mathcal{S}), \\ \displaystyle\int_{\mathcal{S}} \nabla_{\mathcal{S}} v \cdot \nabla_{\mathcal{S}} \psi \ \mathrm{d}A = \int_{\mathcal{S}} f\psi \ \mathrm{d}A, \qquad \forall \psi \in H^1(\mathcal{S}), \end{cases} \tag{4.4}$$

where $v : \mathcal{S} \to \mathbb{R}$ are the auxiliary unknowns.

**Laplace-Beltrami Triharmonic Problem**   The Laplace-Beltrami triharmonic equation with homogeneous boundary conditions is formulated as

$$\begin{cases} -\Delta_{\mathcal{S}}^3 u = f, & \text{in } \mathcal{S}, \\ u = \frac{\partial u}{\partial \boldsymbol{n}} = \Delta_{\mathcal{S}} u = 0, & \text{on } \partial\mathcal{S}. \end{cases} \tag{4.5}$$

Let $\Delta_{\mathcal{S}} u = -v$ and $\Delta_{\mathcal{S}} v = -w$, then the mixed formulation of (4.5) reads as

$$
\begin{cases}
\text{Find } (u, v, w) \in H_0^1(\mathcal{S}) \times H_0^1(\mathcal{S}) \times H^1(\mathcal{S}) \text{ such that} \\[2mm]
\displaystyle \int_{\mathcal{S}} \nabla_{\mathcal{S}} u \cdot \nabla_{\mathcal{S}} \varphi \; \mathrm{d}A - \int_{\mathcal{S}} v\varphi \; \mathrm{d}A = 0, \quad \forall \varphi \in H_0^1(\mathcal{S}), \\[2mm]
\displaystyle \int_{\mathcal{S}} \nabla_{\mathcal{S}} v \cdot \nabla_{\mathcal{S}} \psi \; \mathrm{d}A - \int_{\mathcal{S}} w\psi \; \mathrm{d}A = 0, \quad \forall \psi \in H_0^1(\mathcal{S}), \\[2mm]
\displaystyle \int_{\mathcal{S}} \nabla_{\mathcal{S}} w \cdot \nabla_{\mathcal{S}} \zeta \; \mathrm{d}A = \int_{\mathcal{S}} f\zeta \; \mathrm{d}A, \qquad \forall \zeta \in H^1(\mathcal{S}),
\end{cases}
\tag{4.6}
$$

where $v, w : \mathcal{S} \to \mathbb{R}$ are the auxiliary unknowns.

## 4.2  IGA-Loop Discrertization of Surface PDEs

The finite element space defined by the limit form of the extended Loop subdivision is employed to represent the geometry and also represent the solution space for the discretization of the surface PDEs. The input triangular mesh serves as the initial control mesh of the extended Loop subdivision. Such function is $C^2$-continuous everywhere except at the extraordinary vertices with $C^1$-continuity. Using globally $C^1$-continuous basis functions induced by the extended Loop subdivision yields our IGA finite dimensional space that is the subspace of the Sobolev space $H^2(\mathcal{S})$. For each control vertex $\boldsymbol{x}_i$ of the limit surface $\mathcal{S}$, we associate it with a basis function $\phi_i$, where $\phi_i$ is defined by the limit of the extended Loop subdivision for the zero control values everywhere except at $\boldsymbol{x}_i$ , where it is one. Hence the support of $\phi_i$ is a piecewise function, and covers the two-ring neighborhood of the vertex $\boldsymbol{x}_i$.

The function can be locally parameterized on the unit triangle defined by $T = \{(\xi, \eta) \in \mathbb{R}^2 : \xi \geq 0, \eta \geq 0, \xi + \eta \leq 1\}$ where $(1 - \xi - \eta, \xi, \eta)$ are the barycentric coordinates of the unit triangle. Using this parameterization, the discretized representation of the surface domain is $\mathcal{S}_h = \bigcup_{i=1}^{k} \mathcal{T}_i$, $\mathring{\mathcal{T}}_i \cap \mathring{\mathcal{T}}_j = \varnothing$ for $i \neq j$, where $\mathring{\mathcal{T}}_i$ is the interior of triangular patch $\mathcal{T}_i$. Each triangular patch can be parameterized locally as

$$
\boldsymbol{e}_i : \quad T \to \mathcal{T}_i; \quad (\xi, \eta) \;\mapsto\; \boldsymbol{e}_i(\xi, \eta).
$$

Denote $\boldsymbol{e}_j, j = 1, \ldots, m_i$ be the two-ring neighborhood patches around the control vertex $\boldsymbol{x}_i$. Then if $\boldsymbol{e}_j$ is regular, the explicit box-spline expression as in (3.5) exists for $\phi_i$ on $\boldsymbol{e}_j$. If $\boldsymbol{e}_j$ is irregular, local subdivision, as described in Section 3.2, is needed around $\boldsymbol{e}_j$ until the parameter values of interest are interior to a regular patch. The parameterization has no overlap. Each control vertex $\boldsymbol{x}_i \in \mathcal{S}$ has its unique parameter coordinates. Using the set of the basis functions $\{\phi_i\}$, the limit surface $\mathcal{S}$ of the extended Loop subdivision is expressed as

$$
\mathcal{S} = \boldsymbol{x}(\xi, \eta) = \boldsymbol{x}(x(\xi, \eta), y(\xi, \eta), z(\xi, \eta)) = \sum_{i=1}^{n} \phi_i(\xi, \eta) \boldsymbol{x}_i.
\tag{4.7}
$$

Each control triangular surface patch of the surface $\mathcal{S}$ is defined locally by only a few related basis functions, since the supports of the basis are compact. The surface boundaries are represented as the cubic B-spline curves which are preserved as the subdivision proceeds. With

13

the parameterization, the differential operators on the surface as described in Section 2 can be computed directly, and the computation of the function integration on the surface is replaced by

$$\int_{\mathcal{S}} u \ \mathrm{d}A := \sum_i \int_T u(\boldsymbol{e}_i(\xi, \eta))\sqrt{g} \ \mathrm{d}\xi\mathrm{d}\eta.$$

### 4.2.1 Numerical Discretization

In what follows, we discretize the weak formulations of the Laplace-Beltrami based problems. Let $\boldsymbol{x}_j, j = 1, \ldots, n_0$ be the interior control vertices of the surface $\mathcal{S}$, and $\boldsymbol{x}_j, j = n_0+1, \ldots, n$ be its boundary control vertices. Recalling (4.2), (4.4) and (4.6), by means of the basis functions $\phi_j \in H^2(\mathcal{S}_h)$ introduced by the limit form of the extended Loop subdivision, we have the following discrete description for the function $u^h$ to be determined

$$u^h = \sum_{j=1}^{n_0} \phi_j u_j^h + \sum_{j=n_0+1}^{n} \phi_j u_j^h,$$

where $u_j^h$ $(j = n_0+1, \ldots, n)$ is the known boundary conditions. The other two auxiliary functions $v^h$ and $w^h$ are represented as

$$v^h = \sum_{j=1}^{n} \phi_j v_j^h \quad \text{and} \quad w^h = \sum_{j=1}^{n} \phi_j w_j^h.$$

For simplicity, denote

$$\boldsymbol{U}_K = [u_1^h, \ldots, u_K^h]^\mathrm{T} \in \mathbb{R}^K$$

the solution vector, and the other two auxiliary solution vectors

$$\boldsymbol{V}_K = [v_1^h, \ldots, v_K^h]^\mathrm{T} \in \mathbb{R}^K \quad \text{and} \quad \boldsymbol{W}_K = [w_1^h, \ldots, w_K^h]^\mathrm{T} \in \mathbb{R}^K.$$

The coefficient matrices and the right-hand side terms are represented respectively as

$$\boldsymbol{M}_{K \times L} = [m_{ij}]_{ij=1}^{K,L}, \quad \boldsymbol{K}_{K \times L} = [k_{ij}]_{ij=1}^{K,L} \quad \text{and} \quad \boldsymbol{B}_K = [b_i]_{i=1}^{K},$$

whose elements are

$$m_{ij} = -\int_{\mathcal{S}} \phi_i \phi_j \ \mathrm{d}A, \quad k_{ij} = \int_{\mathcal{S}} [(\nabla_{\mathcal{S}}\phi)_i^\mathrm{T} \nabla_{\mathcal{S}}\phi_j] \ \mathrm{d}A \quad \text{and} \quad b_i = \int_{\mathcal{S}} f\phi_i \ \mathrm{d}A.$$

Take the test functions $\varphi = \phi_i, i = 1, \ldots, n_0$, for the Laplace-Beltrami harmonic problem (4.2) on a quarter of cylinder and one-eighth of sphere in Test Suite 1. Considering the zero boundary condition, we obtain the linear system as

$$\boldsymbol{K}_{n_0 \times n_0}\boldsymbol{U}_{n_0} = \boldsymbol{B}_{n_0}. \tag{4.8}$$

Choose the two sets of the test functions $\varphi = \phi_i, i = 1, \ldots, n_0$, and $\psi = \phi_i, i = 1, \ldots, n$, for the Laplace-Beltrami biharmonic problem (4.4) on a cylinder in Test Suite 2. Considering the homogeneous boundary condition, we achieve the following linear system

$$\begin{bmatrix} \boldsymbol{K}_{n_0 \times n_0} & \boldsymbol{M}_{n_0 \times n} \\ \boldsymbol{0} & \boldsymbol{K}_{n \times n} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_{n_0} \\ \boldsymbol{V}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{B}_n \end{bmatrix}. \tag{4.9}$$

Taking the three sets of the test functions $\varphi = \psi = \zeta = \phi_i, i = 1, \ldots, n$, for the Laplace-Beltrami triharmonic problem (4.6) on a unit sphere in Test Suite 3, the linear system reads as

$$\begin{bmatrix} \boldsymbol{K}_{n \times n} & \boldsymbol{M}_{n \times n} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_{n \times n} & \boldsymbol{M}_{n \times n} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{K}_{n \times n} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_n \\ \boldsymbol{V}_n \\ \boldsymbol{W}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{B}_n \end{bmatrix}. \tag{4.10}$$

### 4.2.2 Precompute the Basis Functions

The framework of our IGA-Loop is the same standard process as the classical FEM. The related basis functions and their derivatives need to be precomputed for each control patch of $\mathcal{S}_h$ before solving the linear systems. However, those computations are not intuitive in comparison with standard linear elements since the required two-ring neighbors around each patch have arbitrary topological structure. And additional geometric data are reflected in the subdivision schemes around boundaries. We classify the control mesh into interior patches, sub-boundary patches, and boundary patches. The patches containing boundary vertices are named as boundary patches. The patches adjacent to boundary patches are called sub-boundary patches. And all of the other patches are called interior pathes.

We use the following algorithms to treat those patches. For interior patches, the Stam's algorithm is applied to this case (see [56]). The sub-boundary patches can be subdivided into four interior sub-patches by once, then the algorithm for interior case can be used. In addition, the boundary patches can be subdivided repeatedly till their sub-patches belong to the sub-boundary case, then the above methods can be applied to evaluate them.

Note that the Stam's fast evaluation scheme is always suitable for interior patches with only one extraordinary vertex. Therefore, it is necessary to first subdivide once each patch of the initial mesh. The evaluation of basis functions over their support elements uses general Gaussian integration, which just needs a few subdivision steps to bring Gaussian quadrature knots into a box-spline patch. The integrations for computing the matrix elements are computed by a 12-point Gaussian quadrature rule. That is, each triangle is subdivided into four sub-triangles and a 3-point Gaussian quadrature rule is employed on each of the sub-triangles. The 3-point Gaussian quadrature rule has error bound $O(h^3)$, where $h$ is the maximal edge length. We adopt the adaptive numerical method developed in our former work [55].

### 4.2.3 Error Estimate

The interpolant error (3.6) obtained in our former work can be applied to the case of second-order surface PDEs with zero boundary condition (4.1). By means of the general analysis method of

15

the classical finite elements, we can obtain the $H^1$-norm estimation of the exact solution $u$ and the approximate solution $u^h$ resulting from IGA-Loop, and the $H^0$-norm estimation by means of the Aubin-Nitsche duality argument. The error estimation is stated as the following theorem.

**Theorem 4.1** *Consider the Laplace-Beltrami harmonic problem defined on the limit form $\mathcal{S}$ of the extended Loop subdivision, endowed with the zero boundary condition. Let $u \in H^2(\mathcal{S})$ be the exact solution of the problem, and $u^h$ be the approximate solution obtained with IGA-Loop strategy, the following error estimates hold:*

$$\|u - u^h\|_{H^l(\mathcal{S})} \le ch^{2-l}\|u\|_{H^2(\mathcal{S})}, \quad l = 0,\ 1, \tag{4.11}$$

*where $c$ is a constant independent of $h$.*

Furthermore, we also analyze the numerical errors for the cases of the fourth-order and the sixth-order surface PDEs, even defined on closed surface like the sphere, which show the consistent convergence rates.

# 5 Numerical Experiments

In this section, we present several numerical experiments to show the performance of our IGA-Loop method by solving the three surface PDEs on different surfaces. It should be indicated that the numerical computations are conducted on the limit surface generated by the extended Loop subdivision, therefore the integration evaluation of the Gauss-Legendre knots is performed on the triangulation of the limit surface.

   We compare the proposed method (IGA-Loop) with the linear finite element method (FEM-Linear) in terms of accuracy, convergence and computational cost. That is because FEM-Linear is the most often used finite elements in industry, which has the same number of control vertices and error estimate as those of IGA-Loop. The following test suites seek to assess the performance of the proposed IGA-Loop against different surface PDEs.

## 5.1 Test Suite 1: Surface Laplace-Beltrami Harmonic Equation

Here we consider the tests for the Laplace-Beltrami harmonic equation. The first example is

$$\begin{cases} -\Delta_{\mathcal{S}}u = f, & \text{in } \mathcal{S}, \\ u = 0, & \text{on } \partial\mathcal{S}, \end{cases} \tag{5.1}$$

with the exact solution

$$u = (1-x)(1-y)\sin(\pi z)$$

on a quarter of the cylinder $\mathcal{S}_1 = \{(x, y, z) : x^2 + y^2 = 1\ \&\ x \ge 0\ \&\ y \ge 0\ \&\ 0 \le z \le 2\}$ with calculated $f$.

We solve the above problem based on three different control meshes from coarse to refined, as shown in the top row of Figure 1. The total numbers of vertices/patches are 221/384, 825/1536, and 3185/6144 respectively. They have the same limit surface $\mathcal{S}_1$, namely, one quarter of a cylinder. The computational accuracy is indicated by the error distribution $u - u^h$. As shown in Figure 1, the accuracy of IGA-Loop is higher than that of FEM-Linear. In addition, we also compare the convergence rate of IGA-Loop against successive refinement. The error is progressively decreasing as the control mesh becoming finer and finer as shown in Figure 2. Apart from the visual comparison in Figure 1, the quantitative comparison of the error between IGA-Loop and FEM-Linear is performed in Figure 2. As we can see that IGA-Loop has higher accuracy.

Consider the second example as the same Laplace-Beltrami harmonic equation as the first one, but with different exact solution

$$u = xyz$$

on a one-eighth of the sphere $\mathcal{S}_2 = \{(x, y, z) : x^2 + y^2 + z^2 = 1 \ \& \ x \geq 0 \ \& \ y \geq 0 \ \& \ z \geq 0\}$ for suitable $f$.

As before, we compute the above surface PDE based on three different control meshes from coarse to refined, as shown in the top row of Figure 3, which has the same limit surface $\mathcal{S}_2$. The total numbers of vertices/patches are 107/176, 389/704, and 1481/2816 respectively. For numerical comparison, FEM-Linear is used to solve the same problem. As shown in Figure 3, the accuracy of IGA-Loop is higher than that of FEM-Linear by the profile of error distribution. We further compare the convergence rate of the proposed method against successive refinement. The error is gradually decreasing as the control mesh becoming finer and finer as shown in Figure 4. The quantitative comparison of the error between the proposed method and FEM-Linear is given in Figure 4. As we can see that IGA-Loop has higher accuracy.

In Figure 2 and 4, the errors $e = \|u - u^h\|_{L^2(\mathcal{S})}$ are computed with the $L^2$-norm of the surface operator, where both methods have the convergence rate 2. From the comparison of the approximation errors involved in FEM-Linear and IGA-Loop discretizations, for which the errors versus the number of subdivision times are plotted, we observe the former is approximately 1.5 times more than the latter. The FEM-Linear discretization requires the number of degree of freedoms more than the IGA-Loop method, which means IGA-Loop is potentially more efficient. The explanation of the phenomenon is that the approximation error is introduced by the discretization of the surface geometries using the linear finite elements, instead, exact representation of the surfaces can be achieved by means of IGA-Loop.

We performed the above two numerical tests to demonstrate the mathematical results in Theorem 4.1. For the Laplace-Beltrami biharmonic problem defined on a cylinder and the Laplace-Beltrami triharmonic problem defined on a unit sphere, the error estimate (4.11) in norm $L^2(\mathcal{S})$ of Theorem 4.1 holds, which will be studied in the following contents.
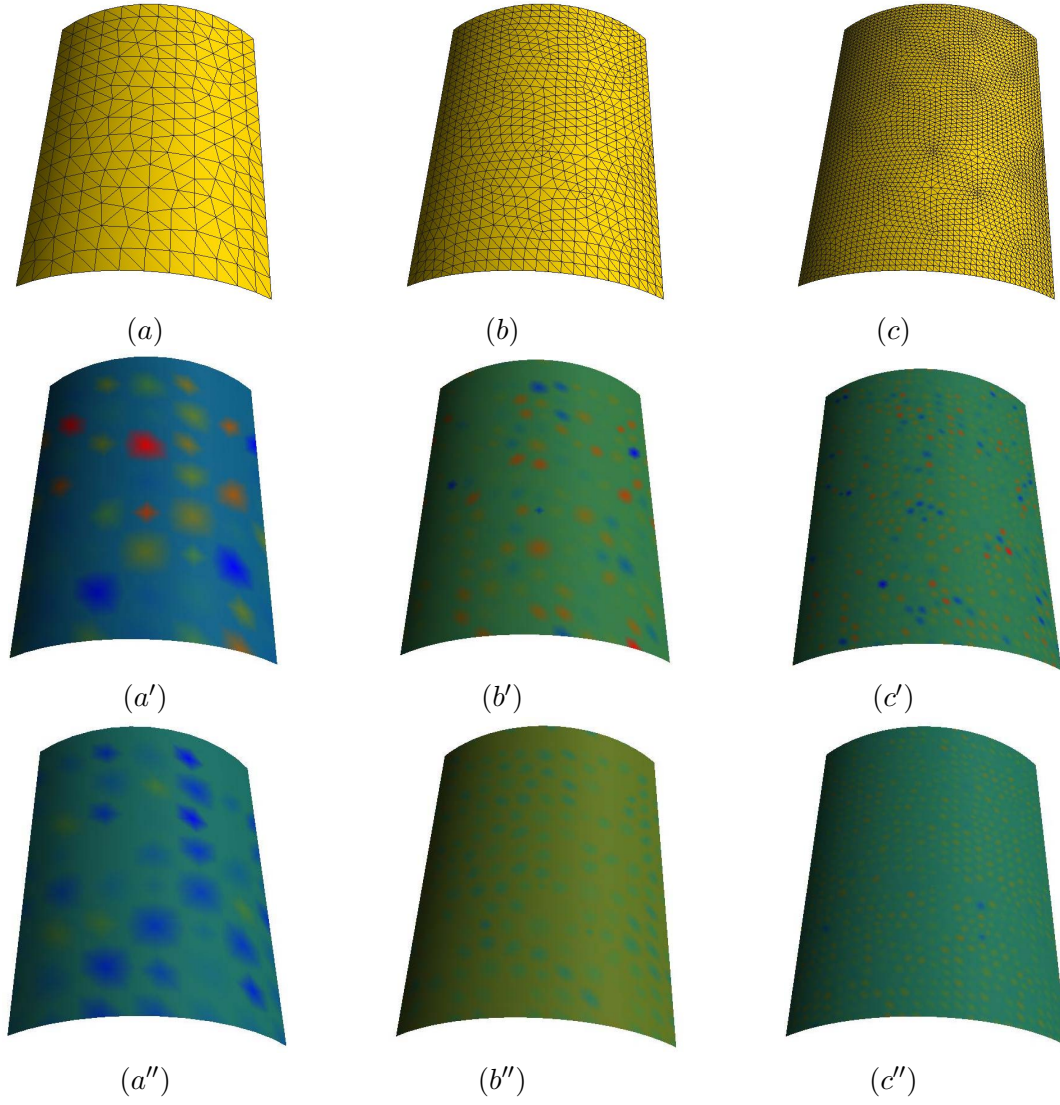
Fig 1: Surface $\mathcal{S}_1$. Top row: $(a)$, $(b)$ and $(c)$ show control meshes from coarse to refined, where only once refinement is implemented from $(a)$ to $(b)$, and $(b)$ to $(c)$. Middle row: The corresponding error distribution of $u - u^h$ results from FEM-Linear shown in $(a')$, $(b')$, and $(c')$. Bottom row: The corresponding error distribution of the proposed IGA-Loop is respectively shown in $(a'')$, $(b'')$, and $(c'')$.
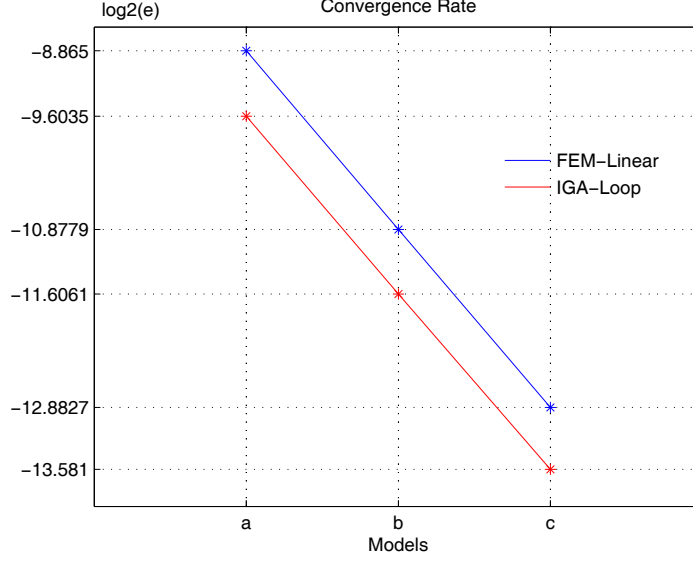
Fig 2: Surface $\mathcal{S}_1$. Comparison of the convergence rate of the errors versus the refinement times between FEM-Linear and the proposed IGA-Loop. Here the numbers a, b and c on the $x$-axis correspond to the models of Figure 1 $(a)$, $(b)$ and $(c)$ respectively, and $e$ on the $y$-axis is the $L^2$-norm error.

## 5.2 Test Suite 2: Surface Laplace-Beltrami Biharmonic Equation

In what follows, we solve a fourth-order example of the Laplace-Beltrami biharmonic equation as

$$\begin{cases} \Delta_{\mathcal{S}}^2 u = f, & \text{in } \mathcal{S}, \\ u = \frac{\partial u}{\partial \boldsymbol{n}} = 0, & \text{on } \partial \mathcal{S}, \end{cases} \tag{5.2}$$

where the exact solution

$$u = \sin^2(2\theta)\sin^2(2z)$$

with $\theta = \operatorname{atan}(y/x)$ on the surface $\mathcal{S}_3 = \{(\theta, z) : 0 \le \theta \le 2\pi \ \& \ 0 \le z \le 1\}$ for suitable $f$.

As shown in the top row of Figure 5, three different control meshes from coarse to refined are given. The total numbers of vertices/patches are 432/768, 1632/3072, and 6336/12288 respectively. They have the same limit surface $\mathcal{S}_3$ as the extended Loop subdivision proceeds. We solve the Laplace-Beltrami biharmonic equation based on the proposed IGA-Loop. The computational accuracy is shown by the error distribution $u - u^h$. As shown in Figure 5, the accuracy of IGA-Loop is higher than that of FEM-Linear.

We further compare the convergence rate of IGA-Loop against successive refinement. The error is gradually decreasing as the control mesh becoming finer and finer as shown in Figure 6. The quantitative comparison of the error in norm $L^2(\mathcal{S})$ versus the number of subdivision
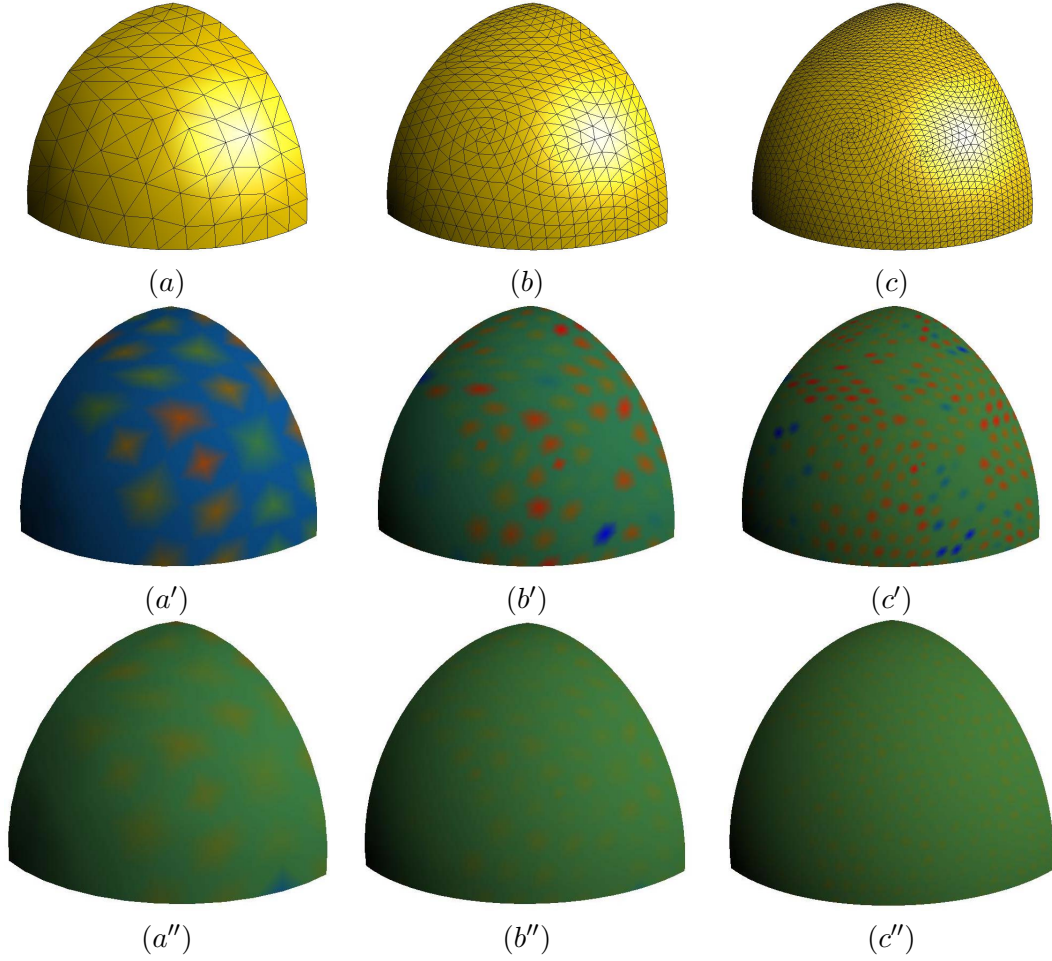
19

Fig 3: Surface $\mathcal{S}_2$. Top row: $(a)$, $(b)$ and $(c)$ are three control meshes where once refinement is implemented from $(a)$ to $(b)$, and $(b)$ to $(c)$. Middle row: The corresponding distribution of the error $u - u^h$ results from FEM-Linear shown in $(a')$, $(b')$, and $(c')$. Bottom row: The error distribution $u - u^h$ results from IGA-Loop shown in $(a'')$, $(b'')$, and $(c'')$.

times, obtained with IGA-Loop and FEM-Linear is given in Figure 6. As shown that IGA-Loop has higher accuracy. In Figure 6, the errors of norm $L^2(\mathcal{S})$ under three times refinement level is plotted. We observe that the convergence rate is 2 for the error in the sense of norm $L^2(\mathcal{S})$, which is consistent with the error estimate (4.11). As shown in Figure 6, the approximation errors obtained by the FEM-Linear discretization is approximately 1.6 times more than that of our IGA-Loop discretization. It means that the IGA-Loop approximation only requires a smaller number of degree of freedoms than FEM-Linear to achieve the same accuracy.
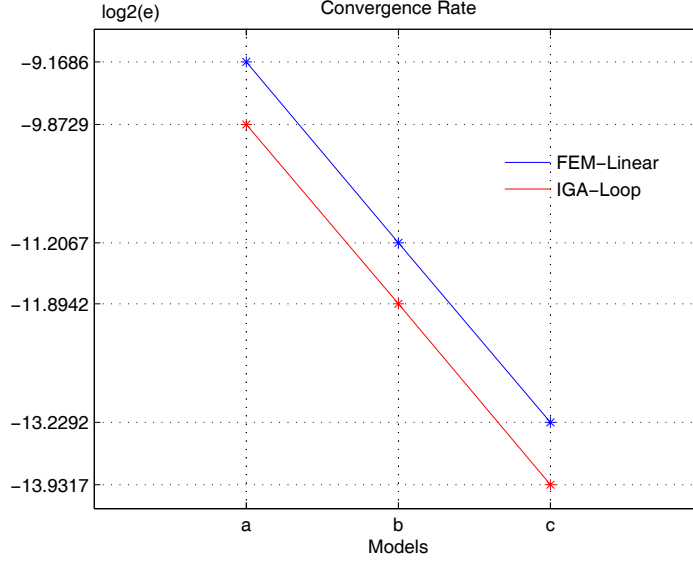
Fig 4: Surface $\mathcal{S}_2$. Comparison of the convergence rate of the errors versus the refinement times between FEM-Linear and IGA-Loop. Here the numbers a, b and c on the $x$-axis correspond to the models of Figure 3 $(a)$, $(b)$ and $(c)$ respectively, and $e$ on the $y$-axis is the $L^2$-norm error.

### 5.3 Test Suite 3: Surface Laplace-Beltrami Triharmonic Equation

Furthermore, a sixth-order Laplace-Beltrami triharmonic equation is solved as following

$$-\Delta_{\mathcal{S}}^3 u = f, \tag{5.3}$$

where for calculated $f$, the exact solution is

$$u = xyz$$

on the closed sphere $\mathcal{S}_4$ with the radius $r = 1$.

As shown in the top row of Figure 7, three different control meshes from coarse to refined are provided. The total numbers of vertices/patches are 706/1408, 2818/5632, and 11266/22528, respectively. They have the same limit surface $\mathcal{S}_4$ by the extended Loop subdivision. We solve the Laplace-Beltrami triharmonic equation based on the proposed IGA-Loop. The computational accuracy is shown by the error distribution $u - u^h$. As shown in Figure 7, the accuracy of IGA-Loop is higher than that of FEM-Linear.

Different from the above two tests, the test model for the sixth-order problem is on a closed surface. We further observe the convergence rate of IGA-Loop against successively finer meshes of the unit sphere in Figure 8. The approximate error (4.11) in norm $L^2(\mathcal{S})$ holds and the same convergence rate 2 is obtained. Furthermore, we can observe the approximation error obtained with the FEM-Linear discretization is approximately 1.8 times more than that of the IGA-Loop
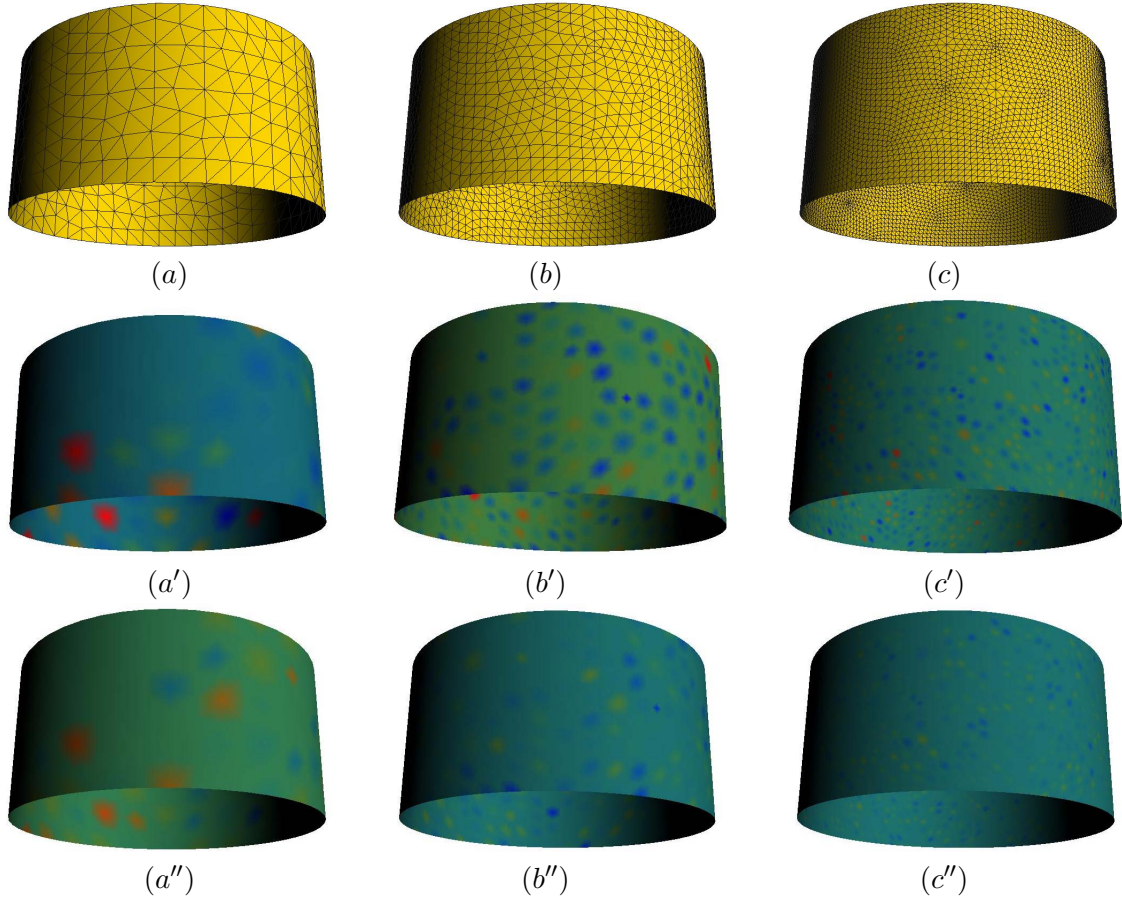
21

Fig 5: Surface $\mathcal{S}_3$. $(a)$, $(b)$ and $(c)$ are three control meshes where one time refinement is implemented from $(a)$ to $(b)$, and $(b)$ to $(c)$. The corresponding distribution of the error $u - u^h$ resulting from FEM-Linear and our IGA-Loop is respectively shown in $(a')$, $(b')$, and $(c')$ of the second row, and $(a'')$, $(b'')$, and $(c'')$ of the third row.

discretization. Therefore, IGA-Loop obtains the same level of accuracy more efficiently than FEM-Linear.

## 5.4   Computational Cost

Actually, the solver of our IGA-Loop can be merged into the framework of FEM-Linear, however the support of each basis function of IGA-Loop is two-ring neighbors described in Section 3. In this section, for the four test examples of Figures 1, 3, 5 and 7, we list the corresponding time costs of computing the basis functions in Tables 1 and 2. To show that the extended Loop subdivision scheme does not require structured meshes and can support the same meshes with any topological structure as the standard finite elements, the valences of the control vertices in
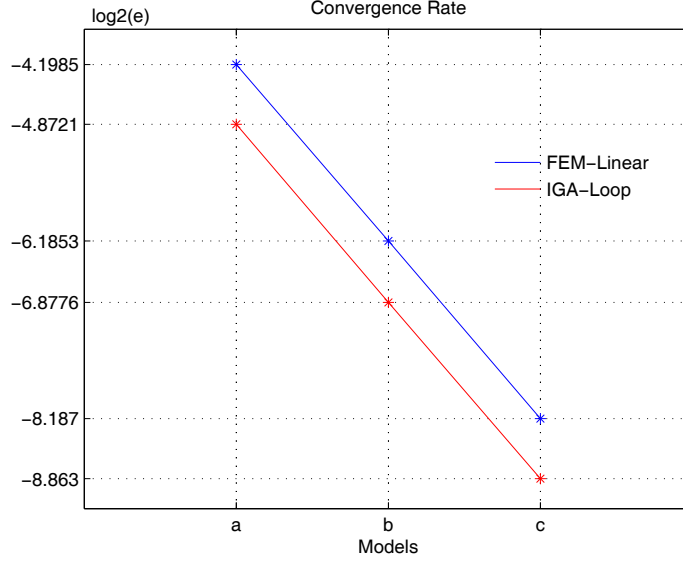
Fig 6: Surface $\mathcal{S}_3$. Comparison of the convergence rate of the errors versus the refinement times between FEM-Linear and IGA-Loop. Here the numbers a, b and c on the $x$-axis correspond to the models of Figure 5 $(a)$, $(b)$ and $(c)$ respectively, and $e$ on the $y$-axis is the $L^2$-norm error.

these surfaces are in the range of 3 to 12 depicted in Figure 1, 3, 5 and 7.

Table 1 corresponds to the examples of the Laplace-Beltrami harmonic problem where the number of control vertices and triangular patches is shown in the first and the third columns, the second and the fourth columns list the time cost (in seconds) of computing the basis functions and their derivatives because they can be pre-computed and saved in a data structure. Once refinement makes the number of triangular patches on the refined meshes increases four times and their sizes approximately decrease by half. The computational cost of IGA-Loop with the same control meshes is larger than FEM-Linear. One reason is the computation of the derivatives of the linear basis functions is unnecessary, as it happens instead with IGA-Loop. The other reason is the computation around the boundary patches is more complex for our IGA-Loop than FEM-Linear because the extended Loop subdivision scheme embodies the angle information different of every patch.

The data for the examples of the Laplace-Beltrami biharmonic problem and the Laplace-Beltrami triharmonic problem are listed in Table 2. Similar results to the Laplace-Beltrami harmonic problem can be observed. Here we notice the data of the Laplace-Beltrami triharmonic problem which is listed in the third and the fourth columns. The example is a closed sphere, the computation cost of IGA-Loop is almost equal to FEM-Linear. We will explain the phenomenon. Firstly the computation of the basis functions around boundary patches for IGA-Loop is removed, and the rest is only the case of computing on the interior patches. Secondly, interior patches share the same set of basis functions which depend only on the valence list of
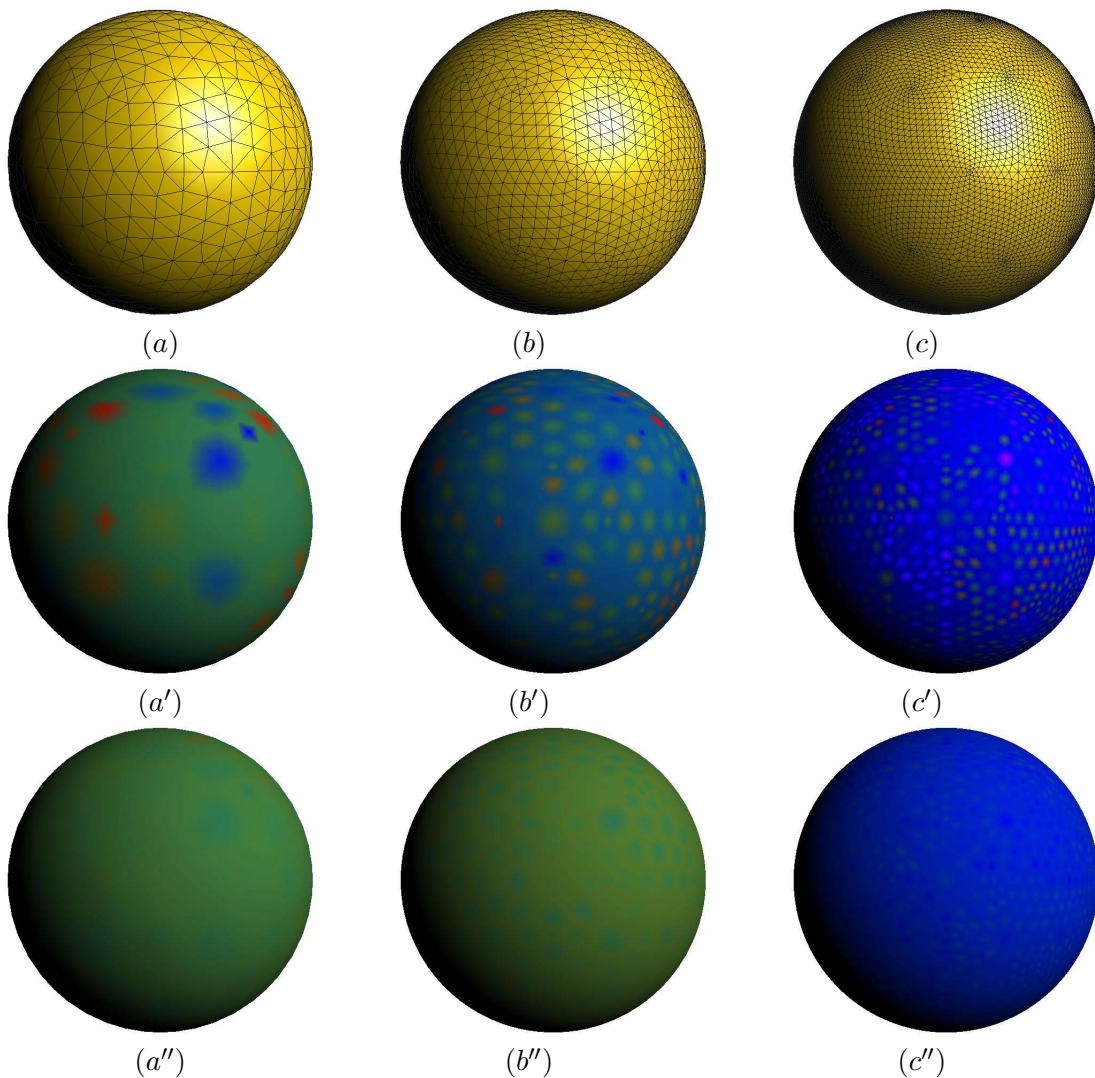
Fig 7: Surface $\mathcal{S}_4$. $(a)$, $(b)$ and $(c)$ are three control meshes where one time refinement is implemented from $(a)$ to $(b)$, and $(b)$ to $(c)$. The corresponding distribution of the error $u - u^h$ resulting from FEM-Linear and IGA-Loop is respectively shown in $(a')$, $(b')$, and $(c')$ of the second row, and $(a'')$, $(b'')$, and $(c'')$ of the third row.

their control vertices, so we can merge the patches according to their valence list, and then use Stam's fast evaluation to treat them. Finally, as the mesh refinement proceeds, the added patches are ordinary whose valences of the three control vertices are six.
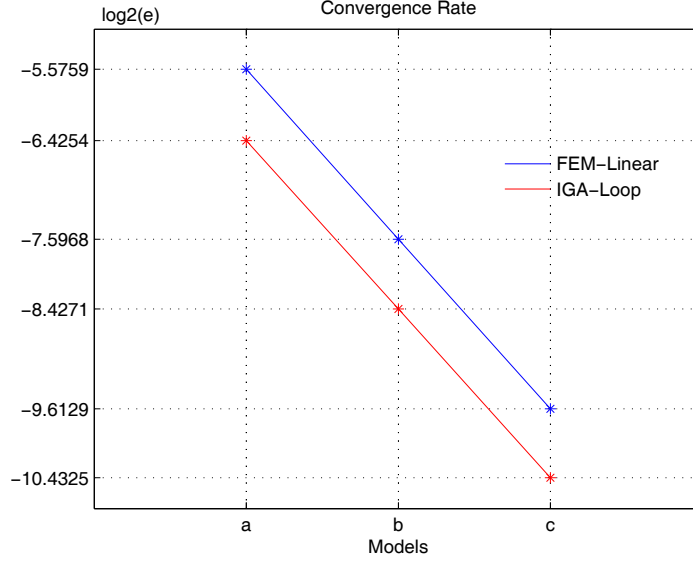
Fig 8: Surface $\mathcal{S}_4$. Comparison of the convergence rate of the errors versus the refinement times between FEM-Linear and IGA-Loop. Here the numbers a, b and c on the $x$-axis correspond to the models of Figure 7 $(a)$, $(b)$ and $(c)$ respectively, and $e$ on the $y$-axis is the $L^2$-norm error.

Table 1: Data of Test Suite 1

| vertices/patches | basis func.(s) | | vertices/patches | basis func.(s) | |
|---|---|---|---|---|---|
| Figure 1 | FEM-Linear | IGA-Loop | Figure 3 | FEM-Linear | IGA-Loop |
| 221/384 | 0.02 | 0.04 | 107/176 | 0.01 | 0.05 |
| 825/1536 | 0.06 | 0.11 | 389/704 | 0.03 | 0.09 |
| 3185/6144 | 0.11 | 0.20 | 1481/2816 | 0.05 | 0.14 |

Table 2: Data of Test Suite 2 and Test Suite 3

| vertices/patches | basis func.(s) | | vertices/patches | basis func.(s) | |
|---|---|---|---|---|---|
| Figure 5 | FEM-Linear | IGA-Loop | Figure 7 | FEM-Linear | IGA-Loop |
| 432/768 | 0.04 | 0.10 | 706/1408 | 0.02 | 0.02 |
| 1632/3072 | 0.11 | 0.21 | 2818/5632 | 0.04 | 0.05 |
| 6336/12288 | 0.23 | 0.42 | 11266/22528 | 0.08 | 0.10 |

# 6   Conclusion

The isogeometric analysis for surface PDEs based on the extended Loop subdivision is presented. The set of quartic box-splines corresponding to each subdivided control mesh are utilized to

represent the geometry of interest, and to construct the solution space for dependent variables as well. The finite elements induced by the extended Loop subdivision possess the ability to represent the geometry exactly which fully agrees with the concept of isogeometric analysis.

We have evaluated the performance of the proposed IGA-Loop in dealing with the various surface PDEs, including the Laplace-Beltrami harmonic/biharmonic/triharmonic equations on different limit surfaces. As shown in the visual and quantitative results, the proposed method can yield desirable performance in terms of the accuracy, convergence and computational cost for solving the above classical surface PDEs. As demonstrated with both the theoretical and numerical results, the proposed IGA-Loop approach is proved to be second-order accuracy in the sense of $L^2$ norm of the surface. Through various comparisons, the performance of IGA-Loop is also outperformed over the standard linear finite elements. The solution of these problems will help to establish a computational framework for the geometric PDEs solved by IGA-Loop in our future work.

# Acknowledgments

# References

[1] Y. Bazilevs, L. Beirão da Veiga, J. A. Cottrell, T. J. R. Hughes and G. Sangalli. Isogeometric analysis: approximation, stability and error estimates for $h$-refined meshes. *Mathematical Models and Methods in Applied Sciences*, 16(7):1031–1090, 2006.

[2] T. J. R. Hughes, J. A. Cottrell and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.

[3] L. A. Piegl and W. Tiller. *The NURBS Book (Monographs in Visual Communication) 2nd ed*. Spinger-Verlag, New york, 1997.

[4] T. W. Sederberg, G. T. Finnigan, X. Li, H. Lin and H. Ipson. Watertight trimmed NURBS., *ACM Transactions on Graphics*, 27(3), article 79, 2008.

[5] T. W. Sederberg, J. Zheng, A. Bakenov and A. Nasri. T-splines and T-NURCCs. *ACM Transactions on Graphics*, 22(3):477–484, 2003.

[6] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng and T. Lyche. T-splines simplification and local refinement., In *ACM SIGGRAPH2004*, pages 276–283, 2004.

[7] W. Li, N. Ray and B. Lévy. Automatic and interactive mesh to T-spline conversion. In *Proceedings of the fourth Eurographics Symposium on Geometry Processing*, pages 191–200, 2006.

[8] Y. Bazilevs, V. M. Calo, J. A. Cottrell, J. A. Evans, T. J. R. Hughes, S. Lipton, M. A. Scott and T. W. Sederberg. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 5-8:229–263, 2010.

[9] C. Giannelli, B. Jüttler, S. K. Kleiss, A. Mantzaflaris, B. Simeon and J. Špeh. THB-splines: An effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 299:337–365, 2016.

[10] T. Dokken, T. Lyche and K. F. Pettersen. Polynomial splines over locally refined box-partitions, *Computer Aided Geometric Design*, 30(3):331–356, 2013.

[11] K. A. Johannessen, T. Kvamsdal and T. Dokken. Isogeometric analysis using LR B-splines. *Computer Methods in Applied Mechanics and Engineering*, 269:471–514, 2014.

[12] N. Nguyen-Thanh, J. Kiendl, H. Nguyen-Xuan, R. Wüchner, K. Bletzinger, Y. Bazilevs and T. Rabczuk. Rotation free isogeometric thin shell analysis using PHT-splines, *Computer Methods in Applied Mechanics and Engineering*, 200(47-48):3410–3424, 2011.

[13] N. Nguyen-Thanh, H. Nguyen-Xuan, S. Bordas and T. Rabczuk, Isogeometric analysis using polynomial splines over hierarchical T-meshes for two-dimensional elastic solids, *Computer Methods in Applied Mechanics and Engineering*, 200(21-22):1892–1908, 2011.

[14] N. Nguyen-Thanh, N. Valizadeh, N. M. Nguyen, H. Nguyen-Xuan, X. Zhuang, P. Areias, G. Zi, Y. Bazilevs, L. De Lorenzis and T. Rabczuk, An extended isogeometric thin shell analysis based on Kirchho-Love theory, *Computer Methods in Applied Mechanics and Engineering*, 284:265–291, 2015.

[15] N. Nguyen-Thanh, K. Zhou, X. Zhuang, P. Areias, H. Nguyen-Xuan, Y. Bazilevs and T. Rabczuk, Isogeometric analysis of large-deformation thin shells using RHTsplines for multiple-patch coupling, *Computer Methods in Applied Mechanics and Engineering*, 316:1157–1178, 2017.

[16] J. Deng, F. Chen, X. Li, C. Hu, W. Tong, Z. Yang and Y. Feng. Polynomial splines over hierarchical T-meshes, *Graphical Models*, 70:76–86, 2008.

[17] V. P. Nguyen, C. Anitescu, S. P. A. Bordas and T. Rabczuk, Isogeometric analysis: An overview and computer implementation aspects, *Mathematics & Computers in Simulations*, 117:89–116, 2015.

[18] G. Xu, Tsz-Ho Kwok and C.L. Wang. Isogeometric computation reuse method for complex objects with topology-consistent volumetric parameterization. *Computer-Aided Design*, 91:1–13, 2017.

[19] X. Ding and C. C. Lim. First-order phase transition and high energy cyclonic spots in a shallow water model on a rapidly rotating sphere. *Physics of Fluids*, 21(4):045102, 2009.

[20] J. Y.-K. Cho and L. M. Polvani. The emergence of jets and vortices in freely evolving, shallowwater turbulence on a sphere. *Physics of Fluids*, 8(6):1531–1552, 1996.

[21] C. D. Cantwell, S. Yakovlev, R. M. Kirby, N. S. Peters and S. J. Sherwin. High-order spectral/hp element discretisation for reaction-diffusion problems on surfaces: Application to cardiac electrophysiology. *Journal of Computational Physics*, 257:813–829, 2014.

[22] A. S. Patelli, L. Dedè, T. Lassila, A. Bartezzaghi and A. Quarteroni. Isogeometric approximation of cardiac electrophysiology models on surfaces: An accuracy study with application to the human left atrium. *Computer Methods in Applied Mechanics and Engineering*, 317:248–273, 2017.

[23] R. P. Fedkiw, G. Sapiro and C.-W. Shu. Shock capturing, level sets, and PDE based methods in computer vision and image processing: a review of oshers contributions. *Journal of Computational Physics*, 185(2):309–341, 2003.

[24] R. Lai and T. F. Chan. A framework for intrinsic image processing on surfaces. *Computer Vision and Image Understanding*, 115(12):1647–1661, 2011.

[25] A. Rätz and A. Voigt. PDEs on surfaces C- a diffuse interface approach. *Communications in Mathematical Sciences*, 4(3):575–590, 2006.

[26] G. Dziuk and C. M. Elliott. Eulerian finite element method for parabolic PDEs on implicit surfaces. *Interfaces and Free Boundaries*, 10:119–138, 2008.

[27] M. Reuter, F.-E. Wolter, M. Shenton and M. Niethammer. Laplace-Beltrami eigenvalues and topological features of eigenfunctions for statistical shape analysis. *Computer-Aided Design*, 41(10):739–755, 2009.

[28] 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 Mathematics, Springer-Verlag, Berlin and Heidelberg*, pages 142–155, 1988.

[29] G. Dziuk and C. M. Elliott. Surface finite elements for parabolic equations. *Journal of Computational Mathematics*, 25(4):385–407, 2007.

[30] M. A. Olshanskii, A. Reusken and X. Xu. A stabilized finite element method for advection-diffusion equations on surfaces. *IMA Journal of Numerical Analysis*, 34(2):732–758, 2014.

[31] K. Mekchay, P. Morin and R. Nochetto. AFEM for the LaplaceCBeltrami operator on graphs: design and conditional contraction property. *Mathematics of Computation*, 80(274):625–648, 2011.

[32] A. Bartezzaghi. *Isogeometric Analysis for High Order Geometric Partial Differential Equations with Applications*. PhD thesis, EPFL, Swizerland, 2017.

[33] H. Hoppe, T. DeRose, T. Duchamp, M. Halstend, H. Jin, J. McDonald, J. Schweitzer and W. Stuetzle. *Piecewise smooth surfaces reconstruction*. In *Computer Graphics Proceedings, Annual Conference series, ACM SIGGRAPH94*, pages 295–302, 1994.

[34] F. Cirak, M. Ortiz and P. Schröder. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *Internat. J. Numer. Methods Engrg.*, 47:2039–2072, 2000.

[35] C. Bajaj and G. Xu. Anisotropic diffusion of surface and functions on surfaces. *ACM Transactions on Graphics*, 22(1):4–32, 2003.

[36] Q. Pan and G. Xu. Construction of quadrilateral subdivision surfaces with prescribed $G^1$ boundary conditions. *Computer-Aided Design*, 43(6):605–611, 2011.

[37] X. Wei, Y. Zhang, T. J. R. Hughes and M. A. Scott. Truncated hierarchical Catmull-Clark subdivision with local refinement. *Computer Methods in Applied Mechanics and Engineering*, 291:1–20, 2015.

[38] B. Hamann, D. Burkhart and G. Umlauf. Isogeometric finite element analysis based on Catmull-Clark subdivision solids. *Computer Graphics Forum*, 29:1575–1584, 2010.

[39] H. Speleers, C. Manni, F. Pelosi and M. L. Sampoli. Isogeometric analysis with Powell-Sabin splines for advection-diffusion-reaction problems. *Computer Methods in Applied Mechanics and Engineering*, 221-222:132–148, 2012.

[40] N. Jaxon and X. Qian. Isogeometric analysis on triangulations. *Computer-Aided Design*, 46(1):45–57, 2014.

[41] Y. Jia, Y. Zhang, G. Xu, X. Zhuang and T. Rabczuk. Reproducing kernel triangular B-spline-based FEM for solving PDEs. *Computer Methods in Applied Mechanics and Engineering*, 267:342–358, 2013.

[42] M. A. Scott, R. N. Simpson, J. A. Evans, S. Lipton, S. Bordas, T. J. R. Hughes and T. W. Sederberg. Isogeometric boundary element analysis using unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering*, 254(2):197–221, 2013.

[43] K. Karciauskas, T. Nguyen and J. Peters. Generalizing bicubic splines for modeling and IGA with irregular layout. *Computer-Aided Design*, 70:23–35, 2016.

[44] D. Toshniwal, H. Speleers and T. J. R. Hughes. Smooth cubic spline spaces on unstructured quadrilateral meshes with particular emphasis on extraordinary points: Geometric design and isogeometric analysis considerations. *Computer Methods in Applied Mechanics and Engineering*, 327:411–458, 2017.

[45] M. Wu, B. Mourrain, A. Galligo and B. Nkonga. Hermite type spline spaces over rectangular meshes with arbitrary topology. *Communications in Computational Physics*, 21(3):835–866, 2017.

[46] L. Dedè and A. Quarteroni. Isogeometric Analysis for second order Partial Differential Equations on surfaces. *Computer Methods in Applied Mechanics and Engineering*, 284:807–834, 2015.

[47] L. Kobbelt, T. Hesse, H. Prautzsch and K. Schweizerhof. Iterative mesh generation for FE-computation on free form surfaces. *Eng. Comput.*, 14:806–820, 1997.

[48] D. Zorin, P. Schröder and W. Sweldens. Subdivision for meshes with arbitrary topology. In *SIGGRAPH '96 Proceedings*, pages 71–78, 1996.

[49] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350–355, 1978.

[50] C. Loop. *Smooth subdivision surfaces based on triangles. Master's thesis.* Department of Mathematices, University of Utah, 1978.

[51] H. Biermann, A. Levin and D. Zorin. Piecewise smooth subdivision surfaces with normal control. In *SIGGRAPH*, pages 113–120, 2000.

[52] G. Xu and Q. Zhang. *Geometric partial Differential Equation Methods in Computational Geometry.* Science Press, Beijing, 2012.

[53] J. E. Schweitzer. *Analysis and Application of Subdivision Surfaces.* PhD thesis, Department of Computer Science and Engineering, University of Washington, Seattle, 1996.

[54] G. Xu. Interpolation by Loop's Subdivision Functions. *Journal of Computational Mathematics*, 23(3): 247–260, 2005.

[55] Q. Pan, G. Xu, G. Xu and Y. Zhang. Isogeometric Analysis Based on Extended Loop's Subdivision. *Journal of Computational Physics*, 299(15):731–746, 2015.

[56] J. Stam. Fast evaluation of Loop triangular subdivision surfaces at arbitrary parameter values. In *SIGGRAPH '98*, CD-ROM supplement, 1998.