

G^1 B-Spline Surface Construction By Geometric Partial Differential Equations^{*}

Ming Li Guoliang Xu[†]

LSEC, Institute of Computational Mathematics, Academy of Mathematics
and System Sciences, Chinese Academy of Sciences, Beijing 100190, China

Abstract

In this paper, we propose a dynamic B-spline technique using general form fourth order geometric PDEs. Basing on discretizations of Laplace-Beltrami operator and Gaussian curvature over triangular and quadrilateral meshes and their convergence analysis, we propose in this paper a novel approach for constructing geometric PDE B-spline surfaces, using general form fourth order geometric flows. Four-sided Spline surface patches are constructed with G^1 boundary constraint conditions. Convergence properties of the proposed method are numerically investigated, which justify that the method is effective and mathematically sound.

Key words: B-Spline surface, discretizations, geometric PDE, Convergence.

1 Introduction

There are two important methods in surface modeling, one is PDE-based approach, and the other is spline technique. Surface modeling using partial differential equations(PDEs) first was proposed by Bloor and Wilson [2] and have developed a special technique called PDE-based techniques. In recent years, a kind of new technique using geometric PDEs have been developed and successfully applied in solving surface smoothing, surface blending, free-form surface modeling, and etc. For example, Mean curvature flow(MCF) is a second order geometric PDE and have been intensively used for smoothing or fairing noisy surfaces(See [6, 13, 1]). However, for solving the surface modeling and designing problem, MCF cannot achieve G^1 smooth jointing of different patches. Therefore, fourth order flows have been used to solve surface blending and free-form surface design(See [5, 21, 18]). These methods all use discrete surface mesh presentation and are improper for spline surface presentation.

In 1973, Riesenfeld [14] proposed a B-spline approximation method in his PHD thesis. Two years later, Versprille(See [17]) generalized Riesenfeld's B-spline to rational form, called NURBS. Today, NURBS have become the industry standard for the surface representation and design in the CAD/CAM. B-spline and NURBS provide a unified mathematical formulation for free-form curves, surfaces using control points. By adjusting the position of control points, one could produce a large variety of shapes. But it is not a easy job to adjust the control points to obtain ideal shapes.

There is another way to integrate the two kinds of approaches. Bloor and Wilson [3] developed an algorithm that represented PDE surfaces - the solution of biharmonic equation - in terms of B-splines. In

^{0*} Partially supported by Natural Science Foundation of China (60773165) and National Key Basic Research Project of China (2004CB318000).

^{0†} Corresponding author.

E-mail address: xuguo@lsec.cc.ac.cn

recent years, Qin and Terzopoulos proposed a dynamic NURBS method(D-NURBS) [16]. By applying Lagrangian mechanics, they obtained a set of second order differential equations that determined the evolution of the D-NURBS. However, the equation they derived is a second order ordinary differential equation and not geometry intrinsic. And it cannot achieve G^1 smooth jointing of NURBS patches.

In this paper, we propose a novel technique that combines B-spline with a general form fourth order geometric PDEs to obtain G^1 smooth PDE B-spline surfaces. Our method is dynamic, the geometric PDE govern the evolution of B-spline surface. And since the B-spline surface produced by our method is the solution of corresponding geometric PDE, it possesses the optimization property of used geometric PDE, such as minimum of area, minimum of total square mean curvature and etc. Besides, the boundary conditions including boundary curves and tangent vectors on them partly control the shape of B-spline. Especially the tangent vector length, not only affects the surface shape, but also the control mesh quality. Hence, in this paper, we employ a unified method to give the length of tangent vector, which could produce good mesh quality. The geometric PDE B-spline surface can be applied in solving surface blending and complex free-form surface design, such as airplane and auto body design. In the end, Our method can be easily generalized to NURBS.

The rest of these paper is organized as followed. Section 2 introduces some basic materials on differential geometry, B-spline and a general form fourth order geometric PDE. In Section 3, we give the process of constructing geometric PDE B-spline surface. Section 4 derives the simplified process for minimal B-spline surface. Numerical experiments are given in Section 5 including numerical convergence experiments, comparative examples to illustrate the different effects, surface blending and airplane body design.

2 Notations and Differential Geometry Preliminaries

In this section, we introduce the used notations, curvatures and several geometric differential operators, including Laplace-Beltrami operator and Giaquinta-Hildebrandt operator etc. We also give the definitions of B-spline surface and general form geometric PDE.

Let $\mathcal{S} = \{\mathbf{x}(u^1, u^2) \in \mathbb{R}^3 : (u^1, u^2) \in \mathcal{D} \subset \mathbb{R}^2\}$ be a regular sufficiently smooth parametric surface. Let $g_{\alpha\beta} = \langle \mathbf{x}_{u^\alpha}, \mathbf{x}_{u^\beta} \rangle$ and $b_{\alpha\beta} = \langle \mathbf{n}, \mathbf{x}_{u^\alpha u^\beta} \rangle$ be the coefficients of the first and the second fundamental forms of \mathcal{S} , where

$$\begin{aligned} \mathbf{x}_{u^\alpha} &= \frac{\partial \mathbf{x}}{\partial u^\alpha}, \quad \mathbf{x}_{u^\alpha u^\beta} = \frac{\partial^2 \mathbf{x}}{\partial u^\alpha \partial u^\beta}, \quad \alpha, \beta = 1, 2, \\ \mathbf{n} &= (\mathbf{x}_u \times \mathbf{x}_v) / \|\mathbf{x}_u \times \mathbf{x}_v\|, \quad (u, v) := (u^1, u^2), \end{aligned}$$

where $\langle \cdot, \cdot \rangle$ and $\cdot \times \cdot$ denote the usual scalar and cross product of two vectors respectively in Euclidean space \mathbb{R}^3 . Set

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

Then the mean curvature H and Gaussian curvature K of \mathcal{S} can be written as follows :

$$H = \frac{1}{2}[g^{\alpha\beta}]:[b_{\alpha\beta}] \quad \text{and} \quad K = \frac{b}{g}, \quad (1)$$

where $A:B$ is the trace of $A^T B$. And

$$\mathbf{H} = H\mathbf{n}, \quad \mathbf{K} = K\mathbf{n}$$

are called as the mean curvature normal vector and Gaussian curvature normal vector respectively.

Tangential gradient operator. Suppose f be a C^1 smooth function on \mathcal{S} , then the tangent gradient operator ∇_s acting on f is given by (see [10], page 102)

$$\nabla_s f = [\mathbf{x}_u, \mathbf{x}_v][g^{\alpha\beta}][f_u, f_v]^T \in \mathbb{R}^3 \quad (2)$$

$$= g_u^\nabla f_u + g_v^\nabla f_v, \quad (3)$$

where

$$g_u^\nabla = \frac{1}{g}(g_{22}\mathbf{x}_u - g_{12}\mathbf{x}_v), \quad g_v^\nabla = \frac{1}{g}(g_{11}\mathbf{x}_v - g_{12}\mathbf{x}_u).$$

Second tangent operator. Let $f \in C^1(\mathcal{S})$, the second tangent operator \diamond acting on f is defined by (see [19])

$$\diamond f = [\mathbf{x}_u, \mathbf{x}_v][h_{\alpha\beta}][f_u, f_v]^T \in \mathbb{R}^3 \quad (4)$$

$$= \frac{1}{g}(b_{22}f_u\mathbf{x}_u + b_{11}f_v\mathbf{x}_v - b_{12}f_u\mathbf{x}_v - b_{12}f_v\mathbf{x}_u) \quad (5)$$

$$= g_u^\diamond f_u + g_v^\diamond f_v \quad (6)$$

where

$$[h_{\alpha\beta}] := \frac{1}{g} \begin{bmatrix} b_{22} & -b_{12} \\ -b_{12} & b_{11} \end{bmatrix}, \quad (7)$$

and

$$g_u^\diamond = \frac{1}{g}(b_{22}\mathbf{x}_u - b_{12}\mathbf{x}_v), \quad g_v^\diamond = \frac{1}{g}(b_{11}\mathbf{x}_v - b_{12}\mathbf{x}_u).$$

Tangent divergence operator. Let \mathbf{v} be a smooth vector field on \mathcal{S} , Then the tangent divergence of \mathbf{v} is defined by

$$\text{div}_s(\mathbf{v}) = \frac{1}{\sqrt{g}} \left[\frac{\partial}{\partial u}, \frac{\partial}{\partial v} \right] \left[\sqrt{g} [g^{\alpha\beta}] [\mathbf{x}_u, \mathbf{x}_v]^T \mathbf{v} \right]. \quad (8)$$

Laplace-Beltrami operator(LBO). Let f be a C^2 smooth function on \mathcal{S} , then the Laplace-Beltrami operator applying to f is given by (see [11], p. 83)

$$\Delta_s f = \text{div}_s(\nabla_s f).$$

From the definitions of (2) and (8) we can derive that

$$\Delta_s f = \frac{1}{\sqrt{g}} \left[\frac{\partial}{\partial u}, \frac{\partial}{\partial v} \right] \left[\sqrt{g} [g^{\alpha\beta}] [f_u, f_v]^T \right] \quad (9)$$

$$= g_u^\Delta f_u + g_v^\Delta f_v + g_{uu}^\Delta f_{uu} + g_{uv}^\Delta f_{uv} + g_{vv}^\Delta f_{vv}, \quad (10)$$

where

$$g_u^\Delta = -(g_{11}(g_{22}g_{122} - g_{12}g_{222}) + 2g_{12}(g_{12}g_{212} - g_{22}g_{112}) + g_{22}(g_{22}g_{111} - g_{12}g_{211}))/g^2,$$

$$g_v^\Delta = -(g_{11}(g_{11}g_{222} - g_{12}g_{122}) + 2g_{12}(g_{12}g_{112} - g_{11}g_{212}) + g_{22}(g_{11}g_{211} - g_{12}g_{111}))/g^2,$$

$$g_{uu}^\Delta = g_{22}/g, \quad g_{uv}^\Delta = -2g_{12}/g, \quad g_{vv}^\Delta = g_{11}/g.$$

Δ_s is a second order differential operator. And $\Delta_s \mathbf{x} = 2\mathbf{H}$.

Giaquinta-Hildebrandt operator (GHO). Let f be a C^2 smooth function on \mathcal{S} , the Giaquinta-Hildebrandt operator \square is defined by (See [12])

$$\square f = \operatorname{div}_s(\diamond f).$$

From the definitions of \diamond and div_s , we can derive that

$$\square f = \frac{1}{\sqrt{g}} \left[\frac{\partial}{\partial u}, \frac{\partial}{\partial v} \right] \left[\sqrt{g} [h_{\alpha\beta}] [f_u, f_v]^T \right] \quad (11)$$

$$= g_u^\square f_u + g_v^\square f_v + g_{uu}^\square f_{uu} + g_{uv}^\square f_{uv} + g_{vv}^\square f_{vv}, \quad (12)$$

where

$$\begin{aligned} g_u^\square &= -[b_{11}(g_{22}g_{122} - g_{12}g_{222}) + 2b_{12}(g_{12}g_{212} - g_{22}g_{112}) + b_{22}(g_{22}g_{111} - g_{12}g_{211})]/g^2, \\ g_v^\square &= -[b_{11}(g_{11}g_{222} - g_{12}g_{122}) + 2b_{12}(g_{12}g_{112} - g_{11}g_{212}) + b_{22}(g_{11}g_{211} - g_{12}g_{111})]/g^2, \\ g_{uu}^\square &= b_{22}/g, \quad g_{uv}^\square = -2b_{12}/g, \quad g_{vv}^\square = b_{11}/g. \end{aligned}$$

2.1 Spline surface

There are many ways to define B-spline basis functions such as by truncated power functions(See[8, 15]), by blossoming [4] and by a recurrence formula(See [7, 9]).In this paper, we adopt the recurrence formula.

Definition. Given a positive integer m , non-negative integer k and knot vector

$$u_0 \leq \cdots \leq u_i \leq u_{i+1} \leq u_{i+2} \leq \cdots \leq u_{m+2k}.$$

$U = \{u_0, \cdots, u_{m+2k}\}$ is called a knot vector. B-splines base functions are defined as follows

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1, & \text{for } u \in [u_i, u_{i+1}), \\ 0, & \text{otherwise,} \end{cases} & i = 0, 1, \cdots, m + 2k - 1, \\ N_{i,k}(u) = \frac{u - u_i}{u_{i+k} - u_i} N_{i,k-1}(u) + \frac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} N_{i+1,k-1}(u), & i = 0, 1, \cdots, m + k - 1, \\ \text{Assume } \frac{0}{0} = 0 \end{cases} \quad (13)$$

where i is the index of $N_{i,k}(u)$, k is the degree.

Spline surface. Given a positive integer m , n , non-negative k and knot sequence

$$U = \{u_0, \cdots, u_{m+2k}\}, \quad V = \{v_0, \cdots, v_{n+2k}\},$$

a k degree quadrilateral B-spline surface is defined by

$$\mathbf{x}(u, v) = \sum_{i=0}^{m+k-1} \sum_{j=0}^{n+k-1} \mathbf{p}_{ij} N_{i,k}(u) N_{j,k}(v), \quad (u, v) \in \Omega := [0, 1]^2,$$

where $\mathbf{p}_{ij} \in \mathbb{R}^3$ are the control points of $\mathbf{x}(u, v)$. When $i = 0$ or m and $j = 0$ or n , \mathbf{p}_{ij} are called as the boundary control points, The adjacent vertices of boundary control points are second boundary control points. To ensure the C^2 smooth of B-spline surface, we take $k \geq 3$.

2.2 Geometric PDE

In order to construct G^1 smooth B-spline surface patch, we employ a general form fourth order geometric partial differential equation:

$$\begin{cases} \frac{\partial \mathbf{x}}{\partial t} + \square(f_K \mathbf{n}) + \frac{1}{2} \Delta_s(f_H \mathbf{n}) - f_H \square \mathbf{x} + (2H f_H + 2K f_K - f) \Delta_s \mathbf{x} \\ \quad + 2f_H \nabla_s H - \diamond f_H + 2H \nabla_s f_H - \nabla_s(f - 2K f_K) = \mathbf{0}, \\ \mathcal{S}(0) = \mathcal{S}_0, \quad \partial \mathcal{S}(t) = \Gamma, \end{cases} \quad (14)$$

which is the L^2 gradient flow of the following energy function defined on \mathcal{S}

$$\mathcal{F}(\mathcal{S}) = \int_{\mathcal{S}} f(K, H) dA = \iint_{\Omega} f(K, H) \sqrt{g} \, dudv$$

(See [20]), where $f(H, K) \in C^1(\mathbb{R} \times \mathbb{R})$ is the given Lagrange function. Suppose f_H and f_K could be represented as

$$f_H = 2\alpha H + 2\beta K + \mu, \quad f_K = 2\gamma H + 2\delta K + \nu, \quad (15)$$

substituting them into first equation of (14) we have

$$\begin{aligned} \frac{\partial \mathbf{x}}{\partial t} &+ \square(\gamma \Delta_s \mathbf{x} + \delta \square \mathbf{x} + \nu \mathbf{n}) + \frac{1}{2} \Delta_s(\alpha \Delta_s \mathbf{x} + \beta \square \mathbf{x} + \mu \mathbf{n}) \\ &- f_H \square \mathbf{x} + (2H f_H + 2K f_K - f) \Delta_s \mathbf{x} \\ &+ 2f_H \nabla_s H - \diamond f_H + 2H \nabla_s f_H - \nabla_s(f - 2K f_K) = \mathbf{0}. \end{aligned} \quad (16)$$

The first row of (16) are fourth order items, producing normal and tangent direction movements. The second row is second order items, only leading to normal movement. The last row items derive tangent direction movement.

3 Geometric PDE B-spline surface construction

Problem description. Given boundary curves and tangent directions of quadrilateral B-spline surface, we hope to construct quadrilateral B-spline surface, which interpolate the boundary conditions and meet some kind of geometric PDE.

Geometric PDE B-spline surface construction. Now we give the steps of constructing geometric PDE B-spline surface. In the next section, we will give the details of each step.

1. **Discretize the differential operator.** We discretize all employed differential geometric operators on quadrilateral mesh (See§3.1).
2. **Construct second boundary control points.** Using the given tangent vectors on the boundary control points, we compute the second boundary control points (See§3.2). If the given boundary conditions include second boundary control points, we can omit this step.
3. **Construct initial inner control points.** In general, the initial inner control points can be chosen arbitrarily. In order to enhance the evolution efficiency, proper initial inner control points is very important. As we all know, If the knots are sufficiently dense, the control polygon of B-spline surface converge to B-spline surface, hence B-spline surface control polygon can be viewed as the

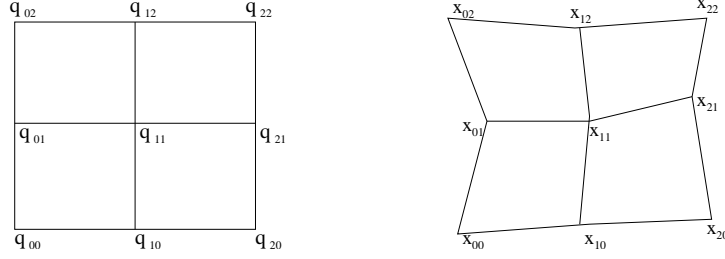


Fig 3.1: Biquadratic interpolation definition domain and mesh vertices index.

approximation of B-spline surface. In this paper, in order to obtain the initial approximation of geometric PDE surface, we use the same geometric PDE to construct the initial inner control points (See§3.3).

4. **Evolve initial inner control points.** Use the selected geometric PDE to evolve the initial inner control points and not stop until the stability status is achieved. We employ the time direction semi-implicit and spatial direction semi-discrete technique to solve the geometric PDE (See§3.4).

3.1 Differential operator discretization

In order to solve geometric PDE numerically on quadrilateral surface patch, we need discretize all the differential operators appeared in the equation on quadrilateral mesh. In this paper, we introduce a kind of discrete scheme based on biquadratic interpolation, which is simple, easy to compute and have the approximaiton attribution. Let \mathbf{x}_{11} is a vertex of quadrilateral mesh. \mathbf{x}_{00} , \mathbf{x}_{10} , \mathbf{x}_{20} , \mathbf{x}_{01} , \mathbf{x}_{21} , \mathbf{x}_{02} , \mathbf{x}_{12} and \mathbf{x}_{22} are the 1-ring neighbor vertices of \mathbf{x}_{11} (See figure3.1). Using the following nine vertices

$$\begin{aligned} \mathbf{q}_{00} &= (-1, -1), & \mathbf{q}_{10} &= (0, -1), & \mathbf{q}_{20} &= (1, -1), \\ \mathbf{q}_{01} &= (-1, 0), & \mathbf{q}_{11} &= (0, 0), & \mathbf{q}_{21} &= (1, 0), \\ \mathbf{q}_{02} &= (-1, 1), & \mathbf{q}_{12} &= (0, 1), & \mathbf{q}_{22} &= (1, 1) \end{aligned}$$

we could construct the biquadratic interpolation function

$$\mathbf{x}(u, v) = \sum_{i=0}^2 \sum_{j=0}^2 \mathbf{c}_{ij} u^i v^j, \quad \mathbf{c}_{ij} \in \mathbb{R}^3,$$

which satisfies the conditions

$$\mathbf{x}(\mathbf{q}_{ij}) = \mathbf{x}_{ij}, \quad i, j = 0, 1, 2.$$

then we could easily derive the following results

$$\begin{aligned} \mathbf{c}_{10} &= \frac{1}{2}(\mathbf{x}_{21} - \mathbf{x}_{01}), & \mathbf{c}_{01} &= \frac{1}{2}(\mathbf{x}_{12} - \mathbf{x}_{10}), \\ \mathbf{c}_{20} &= \frac{1}{2}(\mathbf{x}_{21} + \mathbf{x}_{01} - 2\mathbf{x}_{11}), & \mathbf{c}_{02} &= \frac{1}{2}(\mathbf{x}_{12} + \mathbf{x}_{10} - 2\mathbf{x}_{11}), \\ \mathbf{c}_{11} &= \frac{1}{4}(\mathbf{x}_{00} + \mathbf{x}_{22} - \mathbf{x}_{20} - \mathbf{x}_{02}) \end{aligned}$$

Hence at $(0, 0)$,

$$\begin{aligned}\mathbf{x}_u &= \mathbf{c}_{10}, & \mathbf{x}_v &= \mathbf{c}_{01}, \\ \mathbf{x}_{uu} &= 2\mathbf{c}_{20}, & \mathbf{x}_{uv} &= \mathbf{c}_{11}, & \mathbf{x}_{vv} &= 2\mathbf{c}_{02}.\end{aligned}$$

Let f_{ij} is the function value at \mathbf{x}_{ij} , based on the differential operators definitions of section 2, we construct their discrete scheme as follows:

$$\begin{aligned}\nabla_s f &\approx \frac{1}{2}g_u^\nabla(f_{21} - f_{01}) + \frac{1}{2}g_v^\nabla(f_{12} - f_{10}), \\ \diamond f &\approx \frac{1}{2}g_u^\diamond(f_{21} - f_{01}) + \frac{1}{2}g_v^\diamond(f_{12} - f_{10}), \\ \Delta_s f &\approx \frac{1}{2}g_u^\Delta(f_{21} - f_{01}) + \frac{1}{2}g_v^\Delta(f_{12} - f_{10}) \\ &\quad + g_{uu}^\Delta(f_{21} + f_{01} - 2f_{11}) + \frac{1}{4}g_{uv}^\Delta(f_{00} + f_{22} - f_{20} - f_{02}) + g_{vv}^\Delta(f_{12} + f_{10} - 2f_{11}), \\ \square f &\approx \frac{1}{2}g_u^\square(f_{21} - f_{01}) + \frac{1}{2}g_v^\square(f_{12} - f_{10}) \\ &\quad + g_{uu}^\square(f_{21} + f_{01} - 2f_{11}) + \frac{1}{4}g_{uv}^\square(f_{00} + f_{22} - f_{20} - f_{02}) + g_{vv}^\square(f_{12} + f_{10} - 2f_{11}),\end{aligned}$$

where the coefficients of f_{ij} are defined as in 2 from $\mathbf{x}_u, \mathbf{x}_v, \mathbf{x}_{uu}, \mathbf{x}_{uv}$ and \mathbf{x}_{vv} .

We can prove the following theorem.

Theorem 3.1 *Let \mathbf{x}_i be a vertex of a quadrilateral mesh M with valence 4, $\mathbf{x}_1, \dots, \mathbf{x}_4$ be its neighbor vertices and $\mathbf{x}_{1'}, \dots, \mathbf{x}_{4'}$ be its opposite vertices in the quadrilateral $[\mathbf{x}_i \mathbf{x}_j \mathbf{x}_{j'} \mathbf{x}_{i'}]$. Suppose $\mathbf{x}_i, \mathbf{x}_j$ ($j = 1, \dots, 4$) and $\mathbf{x}_{j'}$ ($j' = 1', \dots, 4'$) are on a sufficiently smooth parametric surface $F(u, v) \in \mathbb{R}^3$, and there exist $\mathbf{q}_i, \mathbf{q}_1, \dots, \mathbf{q}_4, \mathbf{q}_{1'}, \dots, \mathbf{q}_{4'} \in \mathbb{R}^2$ such that*

$$\begin{aligned}\mathbf{x}_i &= F(\mathbf{q}_i), \quad \mathbf{x}_j = F(\mathbf{q}_j), \quad \mathbf{x}_{j'} = F(\mathbf{q}_{j'}), \\ \mathbf{q}_{j'} &= \mathbf{q}_j + \mathbf{q}_{j+1} - \mathbf{q}_i, \quad j = 1, \dots, 4, \quad \mathbf{q}_{j+2} - \mathbf{q}_i = -(\mathbf{q}_j - \mathbf{q}_i), \quad j = 1, 2,\end{aligned}\tag{1}$$

then the above defined discrete differential operators have second-order convergence rate.

Let \mathbf{x}_{ij} be a three dimensional point and $f_{ij} := f(\mathbf{x}_{ij})$ ($i = 0, 1, \dots, m, j = 0, 1, \dots, n$) are the corresponding function values at these points, Denote

$$\begin{aligned}N(i, j) &= \{(k, l) \in \mathbb{N}^2 : |k - i| + |l - j| \leq 1\}, \\ N_\lambda(i, j) &= \{(k, l) \in \mathbb{N}^2 : |k - i| \leq \lambda, \|l - j\| \leq \lambda\},\end{aligned}$$

where λ is a non-negative integer, then the above discrete operators could be represented as :

$$\begin{aligned}\nabla_s f(\mathbf{x}_{ij}) &\approx \frac{1}{2}g_u^\nabla(f_{i+1,j} - f_{i-1,j}) + \frac{1}{2}g_v^\nabla(f_{i,j+1} - f_{i,j-1}), \\ &= \sum_{(k,l) \in N(i,j)} w_{kl}^\nabla f_{kl},\end{aligned}\tag{2}$$

$$\begin{aligned}\diamond f(\mathbf{x}_{ij}) &\approx \frac{1}{2}g_u^\diamond(f_{i+1,j} - f_{i-1,j}) + \frac{1}{2}g_v^\diamond(f_{i,j+1} - f_{i,j-1}), \\ &= \sum_{(k,l) \in N(i,j)} w_{kl}^\diamond f_{kl},\end{aligned}\tag{3}$$

$$\begin{aligned}
\Delta_s f(\mathbf{x}_{ij}) &\approx \frac{1}{2}g_u^\Delta(f_{i+1,j} - f_{i-1,j}) + \frac{1}{2}g_v^\Delta(f_{i,j+1} - f_{i,j-1}) \\
&+ g_{uu}^\Delta(f_{i+1,j} + f_{i-1,j} - 2f_{i,j}) + g_{vv}^\Delta(f_{i,j+1} + f_{i,j-1} - 2f_{i,j}) \\
&+ \frac{1}{4}g_{uv}^\Delta(f_{i-1,j-1} + f_{i+1,j+1} - f_{i+1,j-1} - f_{i-1,j+1}), \\
&= \sum_{(k,l) \in N_1(i,j)} w_{kl}^\Delta f_{kl},
\end{aligned} \tag{4}$$

$$\begin{aligned}
\Box f(\mathbf{x}_{ij}) &\approx \frac{1}{2}g_u^\Box(f_{i+1,j} - f_{i-1,j}) + \frac{1}{2}g_v^\Box(f_{i,j+1} - f_{i,j-1}) \\
&+ g_{uu}^\Box(f_{i+1,j} + f_{i-1,j} - 2f_{i,j}) + g_{vv}^\Box(f_{i,j+1} + f_{i,j-1} - 2f_{i,j}) \\
&+ \frac{1}{4}g_{uv}^\Box(f_{i-1,j-1} + f_{i+1,j+1} - f_{i+1,j-1} - f_{i-1,j+1}) \\
&= \sum_{(k,l) \in N_1(i,j)} w_{kl}^\Box f_{kl},
\end{aligned} \tag{5}$$

where the coefficients of f_{ij} are defined as in section 2 from \mathbf{x}_u , \mathbf{x}_v , \mathbf{x}_{uu} , \mathbf{x}_{uv} and \mathbf{x}_{vv} .

3.2 Construct second boundary control points

Suppose we would construct a tensor product B-spline surface with degree m in the u -direction and degree n in the v -direction, then the given boundary curve is the B-spline curve with corresponding degree. Let boundary control points are given by

$$\mathbf{p}_{0j}, \mathbf{p}_{mj}, \mathbf{p}_{i0}, \mathbf{p}_{in}, \quad i = 0, 1, \dots, m, \quad j = 0, 1, \dots, n.$$

We also need to have tangent vectors on the four boundary edges, which is in the form of B-spline curves with the same degree. Let the tangent vector B-spline curves in the v -direction corresponding to edge $u = 0$ and $u = 1$ be

$$\mathbf{V}^{(0)}(v) = \sum_{j=0}^{n+k-1} \mathbf{v}_j^{(0)} N_{j,k}(v), \quad \mathbf{V}^{(1)}(v) = \sum_{j=0}^{n+k-1} \mathbf{v}_j^{(1)} N_{j,k}(v),$$

respectively, and tangent vector B-spline curves in the u -direction corresponding to edge $v = 0$ and $v = 1$ be

$$\mathbf{U}^{(0)}(u) = \sum_{i=0}^{m+k-1} \mathbf{u}_i^{(0)} N_{i,k}(u), \quad \mathbf{U}^{(1)}(u) = \sum_{i=0}^{m+k-1} \mathbf{u}_i^{(1)} N_{i,k}(u).$$

In order to be consistent with four boundary curves, their coefficients must satisfy the following equations

$$\begin{aligned}
\mathbf{v}_0^{(0)} &= km(\mathbf{p}_{10} - \mathbf{p}_{00}), & \mathbf{u}_0^{(0)} &= kn(\mathbf{p}_{01} - \mathbf{p}_{00}), \\
\mathbf{v}_0^{(1)} &= km(\mathbf{p}_{m+k-1,0} - \mathbf{p}_{m+k-2,0}), & \mathbf{u}_{m+k-1}^{(0)} &= kn(\mathbf{p}_{m+k-1,1} - \mathbf{p}_{m+k-1,0}), \\
\mathbf{v}_{n+k-1}^{(0)} &= km(\mathbf{p}_{1,n+k-1} - \mathbf{p}_{0,n+k-1}), & \mathbf{u}_0^{(1)} &= kn(\mathbf{p}_{0,n+k-1} - \mathbf{p}_{0,n+k-2}), \\
\mathbf{v}_{n+k-1}^{(1)} &= km(\mathbf{p}_{m+k-1,n+k-1} - \mathbf{p}_{m+k-2,n+k-1}), & \mathbf{u}_{m+k-1}^{(1)} &= kn(\mathbf{p}_{m+k-1,n+k-1} - \mathbf{p}_{m+k-1,n+k-2}).
\end{aligned}$$

Which mean that the eight coefficients are defined by boundary curves.

From

$$\left. \frac{\partial \mathbf{x}(u, v)}{\partial u} \right|_{u=0} = km \sum_{j=0}^{n+k-1} (\mathbf{p}_{1j} - \mathbf{p}_{0j}) N_{j,k}(v) = \mathbf{V}^{(0)}(v),$$

we could derive second boundary control points \mathbf{p}_{1j}

$$\mathbf{p}_{1j} = \frac{1}{km} \mathbf{v}_j^{(0)} + \mathbf{p}_{0j}, \quad j = 1, \dots, n+k-2.$$

The other three groups of second boundary control points can be obtained in the similar way. But the four second boundary control points adjacent to four corner points have been computed twice. To obtain the same second boundary control points, the control points of tangent vector curve adjacent to four corner points should satisfy the following conditions:

$$\begin{aligned} \frac{1}{km} \mathbf{v}_1^{(0)} + \mathbf{p}_{01} &= \frac{1}{kn} \mathbf{u}_1^{(0)} + \mathbf{p}_{10}, \\ -\frac{1}{km} \mathbf{v}_1^{(1)} + \mathbf{p}_{m+k-1,1} &= \frac{1}{kn} \mathbf{u}_{m+k-2}^{(0)} + \mathbf{p}_{m+k-2,0}, \\ \frac{1}{km} \mathbf{v}_{n+k-2}^{(0)} + \mathbf{p}_{0,n+k-2} &= -\frac{1}{kn} \mathbf{u}_1^{(1)} + \mathbf{p}_{1,n+k-1}, \\ -\frac{1}{km} \mathbf{v}_{n+k-2}^{(1)} + \mathbf{p}_{m+k-1,n+k-2} &= -\frac{1}{kn} \mathbf{u}_{m+k-2}^{(1)} + \mathbf{p}_{m+k-2,n+k-1} \end{aligned}$$

If the above conditions are not satisfied, we take the average value of twice results as the four second boundary control points. That is

$$\begin{aligned} \mathbf{p}_{11} &= \frac{1}{2} \left[\left(\frac{1}{km} \mathbf{v}_1^{(0)} + \mathbf{p}_{01} \right) + \left(\frac{1}{kn} \mathbf{u}_1^{(0)} + \mathbf{p}_{10} \right) \right], \\ \mathbf{p}_{m+k-2,1} &= \frac{1}{2} \left[\left(-\frac{1}{km} \mathbf{v}_1^{(1)} + \mathbf{p}_{m+k-1,1} \right) + \left(\frac{1}{kn} \mathbf{u}_{m+k-2}^{(0)} + \mathbf{p}_{m+k-2,0} \right) \right], \\ \mathbf{p}_{1,n+k-2} &= \frac{1}{2} \left[\left(\frac{1}{km} \mathbf{v}_{n+k-2}^{(0)} + \mathbf{p}_{0,n+k-2} \right) + \left(-\frac{1}{kn} \mathbf{u}_1^{(1)} + \mathbf{p}_{1,n+k-1} \right) \right], \\ \mathbf{p}_{m+k-2,n+k-2} &= \frac{1}{2} \left[\left(-\frac{1}{km} \mathbf{v}_{n+k-2}^{(1)} + \mathbf{p}_{m+k-1,n+k-2} \right) + \left(-\frac{1}{kn} \mathbf{u}_{m+k-2}^{(1)} + \mathbf{p}_{m+k-2,n+k-1} \right) \right]. \end{aligned}$$

As we know, the lengths of tangent vectors affect the mesh quality. In order to obtain good mesh quality, we need provide a way to choose the proper lengths. In this paper, we propose a unified method—the arc length of mesh—as the length of tangent vector. For example, let V_j^0 be the tangent vector at $[0, j]$, its length equals to the length of arc from $\mathbf{x}(0, j)$ to $\mathbf{x}(1, j)$. Now we give some figures to show our methods is effective.

3.3 Construct initial inner control points

1. Construct initial inner control points.

In order to use geometric PDE to construct initial inner control points, we need give the initial values of initial inner control points. These initial values can be obtained by using cubic interpolations along u -direction and v -direction from boundary control points and second boundary control points. Then represent the Hermite interpolation polynomial in terms of B-spline curve. Take the average values of two directions,

$$\mathbf{p}_{ij}^{(0)} = \frac{1}{2} (\mathbf{p}_{ij}^{(u)} + \mathbf{p}_{ij}^{(v)}), \quad i = 2, \dots, m+k-3, \quad j = 2, \dots, n+k-3,$$

where $\mathbf{p}_{ij}^{(u)}$ and $\mathbf{p}_{ij}^{(v)}$ are control points of B-spline presentation of cubic Hermite interpolations along u -direction and v -direction respectively.

2. Evolve initial values of initial inner control points.

In order to employ geometric PDE to evolve initial inner control points, we discretize the used geometric PDE. For every initial inner control points

$$\mathbf{p}_{ij}, \quad i = 2, \dots, m + k - 3, \quad j = 2, \dots, n + k - 3 \quad (6)$$

construct a linear equation. The detailed process is as follows: Let $\mathbf{x}_{ij}^{(k)}$ be the numerical solutions at $t = k\tau$, we hope to obtain the discrete solutions $\mathbf{x}_{ij}^{(k+1)}$ at $t = (k + 1)\tau$. Initial values are $\mathbf{x}_{ij}^{(0)} := \mathbf{p}_{ij}^{(0)}$.

We use semi-implicit discretization method to discretize (16). It is said to be semi-implicit method because at point \mathbf{x}_{ij} , we discretize equation (16) in the form

$$\sum_{(k,l) \in N_2(i,j)} c_{kl} \mathbf{x}_{kl}^{(k+1)} + \mathbf{b}_{ij} = \mathbf{0}, \quad (7)$$

, where coefficients c_{kl} are computed using the previous step solutions. And coefficients c_{kl} include mean curvature and Gaussian curvature, which can be computed by using discrete scheme (4) and (5).

The first item of equation (16) could be approximated:

$$\frac{\partial \mathbf{x}}{\partial t} \approx \frac{\mathbf{x}_{ij}^{(k+1)} - \mathbf{x}_{ij}^{(k)}}{\tau}, \quad (8)$$

where $\mathbf{x}_{ij}^{(k)}$ are known previous step solutions and should be added to \mathbf{b}_{ij} of equation (7), the coefficients of $\mathbf{x}_{ij}^{(k+1)}$ are added to c_{ij} of (7). The second item of equation (16) could be approximated as

$$\square(\gamma \Delta_s \mathbf{x} + \delta \square \mathbf{x} + \nu \mathbf{n}) \approx \sum_{(k,l) \in N_1(i,j)} w_{kl}^\square (\gamma_{kl} \Delta_s \mathbf{x}_{kl} + \delta_{kl} \square \mathbf{x}_{kl} + \nu_{kl} \mathbf{n}_{kl}), \quad (9)$$

where coefficients w_{kl}^\square , γ_{kl} , δ_{kl} , ν_{kl} and \mathbf{n}_{kl} are computed using the previous solutions, and $\nu_{kl} \mathbf{n}_{kl}$ are treated as known and added to \mathbf{b}_{ij} of equation (7). $\Delta_s \mathbf{x}_{kl}$ and $\square \mathbf{x}_{kl}$ can further be discretized as follows

$$\Delta_s \mathbf{x}_{kl} \approx \sum_{(\mu,\nu) \in N_1(k,l)} w_{\mu\nu}^\Delta \mathbf{x}_{\mu\nu}^{(k+1)}, \quad (10)$$

$$\square \mathbf{x}_{kl} \approx \sum_{(\mu,\nu) \in N_1(k,l)} w_{\mu\nu}^\square \mathbf{x}_{\mu\nu}^{(k+1)}, \quad (11)$$

similarly, coefficients $w_{\mu\nu}^\Delta$ and $w_{\mu\nu}^\square$ are computed using the previous solutions. Substitute (10) and (11) into (9), then the second item of equation (16) can be expressed the sum of the linear combination of $\mathbf{x}_{\mu\nu}^{(k+1)}$ and some known items. The coefficients of $\mathbf{x}_{\mu\nu}^{(k+1)}$ are added to $c_{\mu\nu}$ of (7).

The third item of equation (16) can be dealt with in the same way as the second item. The fourth and fifth items are discretized in the way

$$f_H \square \mathbf{x}_{ij} \approx f_H \Big|_{\mathbf{x}_{ij}^{(k)}} \sum_{k,l \in N_1(i,j)} w_{kl}^\square \mathbf{x}_{kl}^{(k+1)}, \quad (12)$$

$$\begin{aligned} (2Hf_H + 2Kf_K - f) \Delta \mathbf{x}_{kl} \\ \approx (2Hf_H + 2Kf_K - f) \Big|_{\mathbf{x}_{ij}^{(k)}} \sum_{k,l \in N_1(i,j)} w_{kl}^\Delta \mathbf{x}_{kl}^{(k+1)}, \end{aligned} \quad (13)$$

respectively, where coefficients of $\mathbf{x}_{kl}^{(k+1)}$ should be added to c_{kl} of equation (7).

The sixth item of (16) is approximated as follows

$$f_H \nabla_s \mathbf{x}_{ij} \approx f_H \Big|_{\mathbf{x}_{ij}^{(k)}} \sum_{k,l \in N(i,j)} w_{kl}^\nabla H(\mathbf{x}_{kl}^{(k)}), \quad (14)$$

and is computed using the previous solutions, then add the results to \mathbf{b}_{ij} of (7). The all left items could be dealt with in the same way as the sixth item.

Substitute the above discrete scheme into equation (16), we could obtain a linear system with unknown points $\mathbf{x}_{ij}^{(k+1)}$. By Solving the system we get $\mathbf{x}_{ij}^{(k+1)}$. Repeat the process, we obtain $\mathbf{x}_{ij}^{(k+2)}$, $\mathbf{x}_{ij}^{(k+3)}$, \dots , $\mathbf{x}_{ij}^{(l)}$, satisfying

$$\max_{ij} \frac{\|\mathbf{x}_{ij}^{(l-1)} - \mathbf{x}_{ij}^{(l)}\|}{\tau} < \epsilon,$$

where ϵ is a given accuracy control constant. Since the initial inner control points only provide initial values for next step, it is not necessary to be very accurate. Hence here we take $\epsilon = 0.01$. When the iteration process stop, we obtain $\mathbf{x}_{ij}^{(l)}$ and they are the control points \mathbf{p}_{ij} . The linear system is sparse and we use iterative method to solve it.

We should point out that the coefficients c_{kl} of linear system (7) are scalars, not a 3×3 matrix.

3.4 Construct inner control points

First we partition the parameter domain $\Omega = [0, 1]^2$ uniformly. Denote partition points as

$$[u_i, v_j] = \left[\frac{i}{M}, \frac{j}{N} \right]^T, \quad i = 0, 1, \dots, M, j = 0, 1, \dots, N,$$

where

$$M \geq m + k - 1, \quad N \geq n + k - 1.$$

Let $\mathbf{x}_{ij} = \mathbf{x}(u_i, v_j)$, where $\mathbf{x}(u, v)$ is the B-spline surface patch to be solved. At each partition point of parameter domain,

$$[u_i, v_j]^T, \quad i = 2, \dots, M - 2, \quad j = 2, \dots, N - 2, \quad (15)$$

we discretize the used geometric PDE to construct a linear system with the inner control points $\mathbf{x}(u, v)$ as unknown. Let

$$\mathbf{x}^{(k)}(u, v) = \sum_{i=0}^{m+k-1} \sum_{j=0}^{n+k-1} \mathbf{p}_{ij}^{(k)} N_{i,k}(u) N_{j,k}(v),$$

be the solution of geometric PDE at $t = k\tau$, we will construct the solution $\mathbf{x}^{(k+1)}(u, v)$ at the next time step $t = (k + 1)\tau$. Denote

$$\mathbf{x}_{ij}^{(k)} = \mathbf{x}^{(k)}(u_i, v_j),$$

We still employ semi-implicit method to discretize equation (16). The discrete steps is similar with the previous subsection except the next some differences.

- (i) These surface points $\mathbf{x}_{\alpha\beta}^{(k+1)}$ in the items (9)–(13) need further be discretized as

$$\mathbf{x}_{\alpha\beta}^{(k+1)} = \sum_{i=0}^{m+k-1} \sum_{j=0}^{n+k-1} \mathbf{p}_{ij}^{(k+1)} N_{i,k}(u_{\alpha\beta}) N_{j,k}(v_{\alpha\beta}).$$

Hence at $[u_i, v_j]^T$, (16) could be discretized into a linear system with the control points $\mathbf{p}_{kl}^{(k+1)}$ unknown:

$$\sum c_{kl} \mathbf{p}_{kl}^{(k+1)} + \mathbf{b}_{ij} = \mathbf{0}, \quad i = 2, \dots, M-2, \quad j = 2, \dots, N-2. \quad (16)$$

- (ii) The coefficients c_{kl} of equation (16) could be computed using the previous solutions. Since the previous solution is a explicit B-spline surface presentation $\mathbf{x}^{(k)}(u, v)$, the mean curvature, Gaussian curvature and normal vector could be obtained from $\mathbf{x}^{(k)}(u, v)$.
- (iii) If $M > m + k - 1$ or $N > n + k - 1$, (16) is super definite. We can derive the least square solution. By solving the normal equation, we obtain $\mathbf{p}_{ij}^{(k+1)}$. Repeat this process and we achieve $\mathbf{p}_{ij}^{(k+2)}$, $\mathbf{p}_{ij}^{(k+3)}$, \dots , $\mathbf{p}_{ij}^{(l)}$. This process will not stop until the condition

$$\max_{ij} \frac{\|\mathbf{p}_{ij}^{(l-1)} - \mathbf{p}_{ij}^{(l)}\|}{\tau} < \epsilon$$

is satisfied. ϵ is a given accuracy control constant and we take $\epsilon = 10^{-5}$.

4 Minimal B-spline surface

It is well known that minimal surface play an important role in geometry design and deserve to discuss it in details. The previous algorithm could be used in constructing minimal B-spline surface by some simplifying.

- (i) In the above problem description, we need not provide the derivative information, therefore the second boundary control points are also kept unknown, which mean the inner control points are

$$\mathbf{p}_{ij}, \quad i = 1, \dots, m + k - 2, \quad j = 1, \dots, n + k - 2 \quad (1)$$

- (ii) Initial inner control points are constructed by using linear interpolation.
- (iii) The initial inner control points to be evolved are (1), not (6). In constructing inner control points, the sample points (15) are changed into

$$[u_i, v_j]^T, \quad i = 1, \dots, M-1, \quad j = 1, \dots, N-1,$$

and equation (16) is turned into

$$\sum c_{kl} \mathbf{p}_{kl}^{(k+1)} + \mathbf{b}_{ij} = \mathbf{0}, \quad i = 1, \dots, M-1, \quad j = 1, \dots, N-1. \quad (2)$$

- (iv) In discretizing equation (16), we take $f = 1$, then the all items vanish except the fifth item $\Delta_s \mathbf{x}$.

5 Numerical experiments

5.1 Numerical convergence experiments

Because of the high nonlinear property of used geometric PDE, it is very difficult to prove the convergence of our methods theoretically. But the numerical experiments show the numerical convergence of our methods. we choose the exact solution of geometric PDE as the approximation model, and compute the maximal error between the discrete and exact solution when n is taken to be 4, 6, \dots to observe the asymptotic property.

Example 1. In equation (16), take $f = K^2$, then the column surface is the solution of used geometric PDE. We use two quadrilateral PDE B-spline patches to approximate the column, and observe the approximation results with n increasing. The results are showed in column 2 of table 5.1.

Example 2. Take $f = H^2$, we get the complete variation Willmore flow(WF). Sphere and ring are the solutions of Willmore flow. We use six and eight quadrilateral Willmore B-spline surface patches to approximate sphere and ring respectively. The results are put in column 3 and 4 of table 5.1.

Example 3. Take $f = 1$, the resulted geometric PDE is mean curvature flow(MCF). Minimal surface is the solution of MCF. Let \mathcal{S} is a minimal surface defined by

$$\mathbf{x}(u, v) = (3u(1 + v^2) - u^3, 3v(1 + u^2) - v^3, 3(u^2 - v^2)), \quad (1)$$

we employ a quadrilateral PDE B-spline surface patch to approximate \mathcal{S} . See the column 5 of table 5.1 for the results.

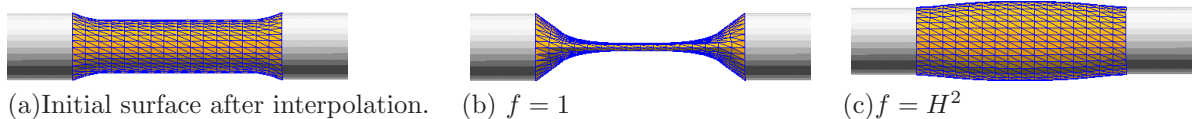
Table 5.1: The maximal asymptotic errors

n	Column- $f = K^2$	Sphere-WM	Ring-WM	Minimal-MCF
4	3.567e-01	8.091e-03	1.906e-01	8.245e-02
6	2.422e-01	3.493e-03	1.242e-01	2.456e-02
8	1.745e-01	2.084e-03	1.045e-01	1.699e-02
10	1.350e-01	1.450e-03	9.814e-02	1.142e-02
12	1.267e-01	1.020e-03	9.408e-02	5.798e-03
14	1.054e-01	8.700e-04	9.127e-02	4.083e-03

From the above table 5.1, we can see that the approximation effects are very excellent. That means our proposed methods are numerical convergence.

5.2 Comparative examples of different geometric PDEs.

In order to show the different effects of different geometric PDEs, we take five different fourth order geometric flows to evolve the same initial surface. The results are as follows.



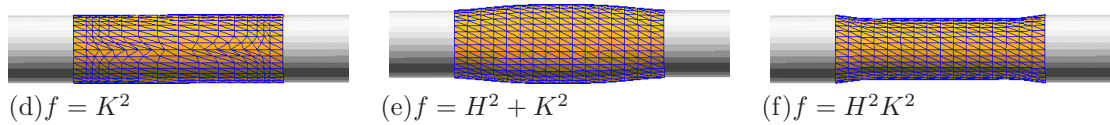


Fig 5.1: Comparison of five different fourth order geometric PDEs.

Figure 5.1 (a) shows the example of minimal surface construction. Figure 5.1 (b) is the example of Willmore surface. Figure 5.1 (b) and (e) result in the same convex surface for the same initial surface. Figure 5.1 (d) and 5.1 (f) show that the column and initial surface are the solutions of used corresponding geometric PDEs respectively.

5.3 Surface blending and complex surface design.

Figure 5.2(a), 5.3(a) and 5.4(a) show the initial input surface with holes. Figure 5.2(b), 5.3(b) and 5.4(b) show the interpolation effects of input surfaces. They are not G^1 smooth. Figure 5.2(c), 5.3(c) and 5.4(c) show the evolution effects of interpolation surfaces. The isophote shows they are G^1 smooth.

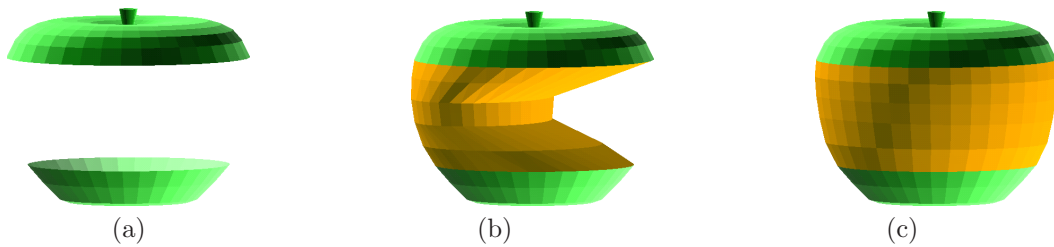


Fig 5.2: (a)Initial surface with a hole. (b) Initial B-spline blending surface. (c)Evolution result using WF.

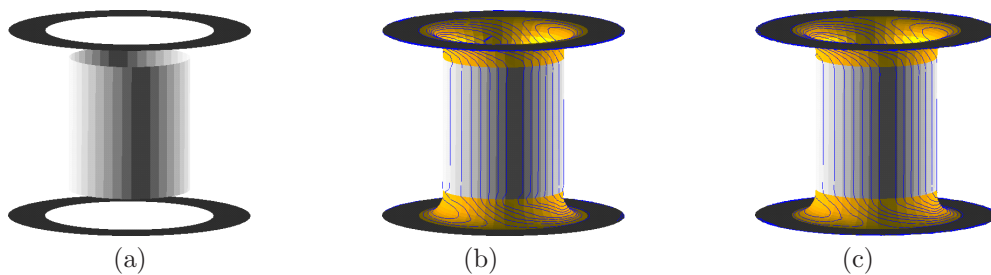


Fig 5.3: (a)Initial surface with holes. (b)Initial B-spline blending surface. (c)Evolution result using WF.

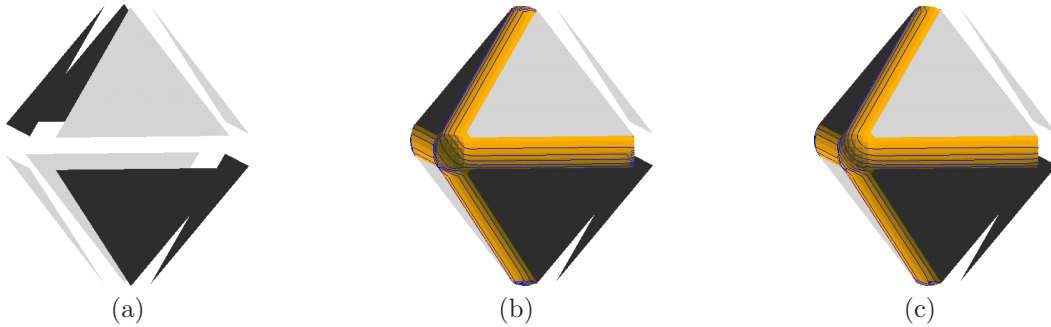


Fig 5.4: (a)Initial surface with holes. (b) Initial B-spline blending surface. (c)Evolution result using WF.

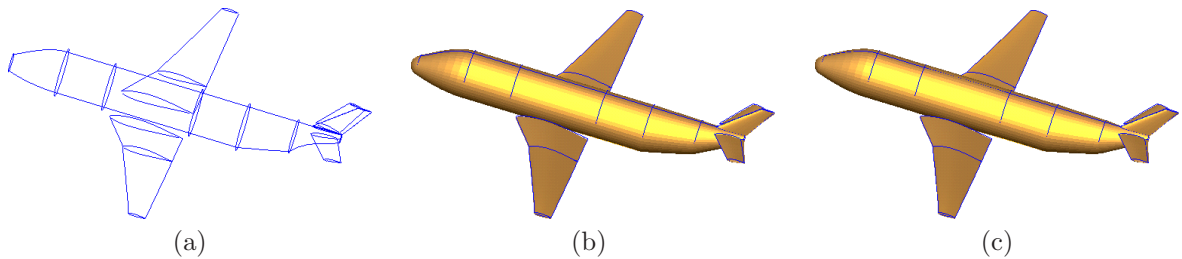


Fig 5.5: (a)Input wire frame of airplane body. (b) Initial construction. (c)Evolved surface.

Figure 5.5 (a) is the wire frame of airplane body. The boundary curves and tangent vectors have been given. Figure 5.5 (b) shows the result of interpolation. Figure 5.5 (c) is the evolution results using WF and MCF.

6 Conclusions

In this paper, We have presented a novel technique that integrates the B-spline and geometric PDEs. We use geometric PDEs to adjust the positions of inner control points and produce PDE B-spline surface with some optimization property. Numerical experiments show our method is effective and have been successfully applied in solving surface blending and airplane body design with G^1 smooth.

References

- [1] C. Bajaj and G. Xu. Anisotropic Diffusion of Subdivision Surfaces and Functions on Surfaces. *ACM Transactions on Graphics*, 22(1):4–32, 2003.
- [2] M. I. G. Bloor and M. J. Wilson. Generating blend surfaces using partial differential equations. *Computer Aided Design*, 21(3):165–171, 1989.
- [3] M.I.G. Bloor and M.J.Wilson. Representing PDE surfaces in terms of B-splines. *Computer Aided Design*, 22(6):324–331, 1990.
- [4] Ramshaw L. Blossoming. a connect-the-dots approach to spline. Report 19, Digital,, System Research Center, Palo Alto, CA, 1987.
- [5] U. Clarenz, U. Diewald, G. Dziuk, M. Rumpf, and R. Rusu. A finite element method for surface restoration with smooth boundary conditions. *Computer Aided Geometric Design*, 21(5):427–445, 2004.

- [6] U. Clarenz, U. Diewald, and M. Rumpf. Anisotropic geometric diffusion in surface processing. In *VIS '00: Proceedings of the Conference on Visualization*, pages 397–405, Los Alamitos, CA, USA, 2000.
- [7] M. G. Cox. The numerical evaluation of B-splines. *Jour. Inst. Math. Applic.*, 10:134–149, 1972.
- [8] H. B. Curry and I. J. Schoenberg. On spline distribution and their limits: the Pólya distribution functions. *Bull. Amer. Math. Soc.*, 53:109, 1947.
- [9] C. DeBoor. On calculating with B-splines. *Journal of Approximation Theory*, 6:50–62, 1972.
- [10] M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, NJ, 1976.
- [11] M. P. do Carmo. *Riemannian Geometry*. Birkhäuser, Boston, Basel, Berlin, 1992.
- [12] M. Giaquinta and S. Hildebrandt. *Calculus of Variations*, Vol. I. Number 310 in A Series of Comprehensive Studies in Mathematics. Springer-Verlag, Berlin, 1996.
- [13] Y. Ohtake, A. G. Belyaev, and I. A. Bogaevski. Polyhedral surface smoothing with simultaneous mesh regularization. In *Geometric Modeling and Processing 2000 Proceedings*, pages 229–237.
- [14] R.F. Riesenfeld. *Applications of B-spline Approximation to Geometric Problems of Computer-Aided Design*. PhD thesis, Syracuse University, 1973.
- [15] I. J. Schoenberg. Contributions to the problem of approximation of equidistance data by analytic functions. *Quart. Appl. Math.*, 4:45–99, 1946.
- [16] D. Terzopoulos and H. Qin. Dynamic NURBS with geometric constraints for interactive sculpting. *ACM Transactions on Graphics*, 13(2):103–136, 1994.
- [17] K.J. Versprille. *Computer-Aided Design Applications of the Rational B-Spline Approximation form*. PhD thesis, Syracuse University, 1975.
- [18] G. Xu and Q. Pan. G^1 Surface Modelling Using Fourth Order Geometric Flows. *Computer-Aided Design*, 38(4):392–403, 2006.
- [19] G. Xu and Q. Zhang. Construction of Geometric Partial Differential Equations in Computational Geometry. *Mathematica Numerica Sinica*, 28(4):337–356, 2006.
- [20] G. Xu and Q. Zhang. A General Framework for Surface Modeling Using Geometric Partial Differential Equations. *Computer Aided Geometric Design*, 25(3):181–202, 2008.
- [21] S. Yoshizawa and A. G. Belyaev. Fair triangle mesh generation with discrete elastica. In *Geometric Modeling and Processing*, pages 119–123, Saitama, Japan, 2002.