Computing (2009) 86:235–255
DOI 10.1007/s00607-009-0056-5

Local and singularity-free G^1 triangular spline surfaces using a minimum degree scheme

Wei-hua Tong · Tae-wan Kim

Received: 26 June 2009 / Accepted: 14 July 2009 / Published online: 19 August 2009

Abstract We develop a scheme for constructing G^1 triangular spline surfaces of arbitrary topological type. To assure that the scheme is local and singularity-free, we analyze the selection of scalar weight functions and the construction of the boundary curve network in detail. With the further requirements of interpolating positions, normals, and surface curvatures, we show that the minimum degree of such a triangular spline surface is 6. And we present a method for constructing boundary curves network, which consists of cubic Bézier curves. To deal with certain singular cases, the base mesh must be locally subdivided and we proposed an adaptive subdivision strategy for it. An application of our G^1 triangular spline surfaces to the approximation of implicit surfaces is described. The visual quality of this scheme is demonstrated by some examples.

Keywords G^1 continuity · Arbitrary topology · Vertex enclosure constraint · Singularity analysis · Geometric Hermite interpolation · Adaptive subdivision

Mathematics Subject Classification (2000) 53A04 · 65D05 · 65D07 · 65D17

Communicated by C.H. Cap.

W. Tong
Department of Naval Architecture and Ocean Engineering, Seoul National University,
Seoul 151-744, Republic of Korea

T. Kim (✉)
Department of Naval Architecture and Ocean Engineering, Research Institute of Marine Systems Engineering, Seoul National University, Seoul 151-744, Republic of Korea
e-mail: taewan@snu.ac.kr

1 Introduction

In computer-aided geometric design (CAGD), computer graphics, and computer-aided design (CAD), designing complicated free-form shapes can be challenging and burdensome. Constructing a surface from a given mesh or a boundary curves network (which provides an intuitive way of defining a surface) has been one of the most popular methods of modeling freeform surfaces. The resulting surface is required to interpolate or approximate the given mesh or boundary curves network and satisfy some smoothness constraints such as C^1, G^1 or C^2.

Over the past few decades, this problem has been considered in many studies. Some have used multivariate spline theory, such as Sabin [29], de Boor and Höllig [3], Boehm et al. [2], but these classic methods do not allow the representation of surfaces of arbitrary topological type. To overcome this, recursive subdivision surfaces have been devised, and interested readers are referred to [33] and [30], among others. The subdivision schemes often generate pleasing shapes but their analytic definitions are still work in progress.

Another widely used technique is the ‘filling in’ or ‘joining’ of patches. In this technique, the control points of adjacent patches are adjusted to satisfy the smoothness constraints. The method presented in this paper falls into this category, and we briefly review some related work in Sect. 2. The advantages of our method are as follows:

- *Local scheme*: all of the continuity constraints are grouped into a local linear system, and the resulting surface is local.
- *Singularity-free*: the final spline surface is exactly G^1, and no singularities exist.
- *Arbitrary topological type*: no restrictions exist on the genus of surface and the valences of the vertices in the base mesh.
- *High order interpolation*: the resulting surface can interpolate the given positions, normals, and surface curvatures at the vertices of the base mesh.

Our method involves only local and linear solvers, which leads to a straightforward implementation. Tong and Kim [32] proposed a similar scheme of degree 7 of piecewise polynomial G^1 triangular spline surface, which bears the above desirable properties. In that paper, the degree of boundary Bézier curves is quartic, and the base mesh is immutable after they been constructed. In that situation, we believe that the degree 7 strikes the optimal balance between shape quality, singularity, computation time and robustness. In this paper, we show that the degree of G^1 triangular spline surface can be reduced to 6 which is the minimum degree required if the base mesh is permitted to be locally subdivided to eliminate singularity. And we present a method for constructing boundary curves network that consists of cubic Bézier curves, together with an adaptive subdivision strategy of the base mesh.

This paper is organized as follows. In the next section, some related work is reviewed. In Sect. 3, we present some preliminaries on triangular spline surfaces and geometric continuity constraints. In Sects. 4 and 5, local and singularity-free G^1 triangular spline surfaces of minimum degree are derived, and the choice of scalar weight functions and boundary curves network is analyzed. Section 6 describes an application of our G^1 triangular spline surfaces to the approximation of implicit surfaces. Section 7 concludes the paper with a summary and some proposals for future research.

2 Related work

Geometric or visual continuity has been studied extensively, and excellent surveys can be found in [9,15,27]. G^1 continuity (i.e., geometric tangent-plane continuity) is particularly commonplace in parametric free-form surface design. Accordingly, a large body of related research exists on construction of geometric continuous spline surfaces. For the sake of clarity, we will review some of this, focusing on techniques that are closest to our own.

In the construction of G^1 spline surfaces of arbitrary topological type, one of the most important issues is the vertex enclosure constraint, or twist compatibility condition, which has been widely addressed by Watkins [34,24], among others. As discussed in [24], several feasible approaches are used, such as *domain splitting* [8,16], *convex combination* [21,13], *singular parameterizations* [20] and *surface splines* [25,26]. Each method has its own advantages and drawbacks. A recent technique [10] achieves G^1 triangles on GPU, named PNG1, which enhances PN patches using rational blending functions and geometry shaders.

The method described in this paper is of a type known as *boundary curves schemes* in which a C^2 consistent boundary curves network is constructed and the patches are then filled in. Sabin [28] devised a technique of constructing a C^2 consistent boundary curves network with given curvature tensors at the vertices and filling in the patches by bi-cubic Hermite patches. Although the resulting surface is of a low degree, Sabin's scheme is global and not singularity-free. In [24], Peters proposed a sufficient condition for satisfying vertex enclosure constraints, and presented an algorithm for the interpolation of a cubic curve mesh by a bi-quartic G^1 surface. Unfortunately, this scheme has several restrictions on its input data, e.g. the ratios of the two coefficients from the linear scalar weight function at either endpoints of the boundary curve must be equal.

By employing the theory of circulant matrices, Loop [18] proposed another way to construct a C^2 consistent boundary curves network using affine combinations of the vertices of the base mesh. In that scheme, the degree of the boundary curves network is 4, and the degree of the fill-in patches is 6. It is one of the earliest schemes which bear local and singularity-free properties for arbitrary topological type. As the author pointed out, interpolation is theoretically possible, but leads to unwanted surface undulations on occasion. Moreover, Loop's scheme does not accommodate itself to high-order data, such as normals, surface curvatures. Hahmann and Bonneau [14] generalized Loop's scheme with the help of a domain-splitting technique, and here each macro-patch consists of 4 quintic triangular Bézier patches. Further, Yvart et al. [35] introduced hierarchical triangular splines for the purpose of local adaption of the level of detail. Although ingenious, these schemes still have the the problems common to all domain-splitting schemes, e.g. they generate more patches than other methods and have problems to select shape parameters in a reasonable way, refer to Hoschek and Lasser [15]. Another way to improve Loop's scheme was proposed by Liu and Mann [17]. By replacing G^1 continuity with approximate G^1 continuity, they reduced the degree of fill-in patches from 6 to 5, and improved the shape quality of surface by setting the middle control points of boundary curves as the average of centroids from the virtual mesh. Several examples show that their scheme can trade off between

Table 1 Comparison of the present study with related work on constructing G^1 spline surfaces of arbitrary topological type

References	Boundary curves network			Filling in patches		Features	
	Valence	Degree	Restrictions	Degree	Type	Singularity free	Solution type
Sabin [28]	n	3	No	Bi-3	Rect.	No	Local
Peters [24]	n	3	Yes	4, Bi-4	Tri., Rect.	No	Global
Loop [18]	n	4	No	6	Tri.	Yes	Local
Hahmann et al. [14]	n	3	No	5	Macro-Tri.	Yes	Local
Liu and Mann [17]	n	4	No	5	Tri.	Yes	Local, $\varepsilon - G^1$
Cho et al. [5]	3, 4, 5	n	Yes	(n+3), Bi-(n+3)	Tri., Rect.	Yes	Local
Present study	n	3	No	6	Tri.	Yes	Local

continuity and shape properties, and results in surfaces of reasonable quality for sparse data sets. Like Loop's scheme, Liu–Mann's scheme only interpolates the positions at the mesh vertices.

Recently, Cho et al. [5] proposed a local method of constructing G^1 Bézier surfaces for ship hull design by interpolating a given curves network with odd- and 4-valent vertices. In that paper, several subdivision schemes were devised for preprocessing the initial curves network to remove topological irregularities. Analysis and avoidance of singularities for their method are described in [6]. However, the preprocessed curves network may contain many more patches than original one. Moreover there are some restrictions on its input data, e.g. none of the two bordering boundary curves are collinear. For a brief summary, see Table 1.

3 Preliminaries

3.1 G^1 triangular spline surfaces

Let $\mathcal{M} = (V, E, F)$ be a given triangular mesh, where V is the set of vertices, E the set of edges and F the set of faces. For each face $\triangle \in \mathcal{M}$, every point $\mathbf{p} \in \triangle$ has barycentric coordinates $\mathbf{u} = (u, v, w)$ with respect to the vertices of $\triangle$. A *triangular Bézier patch* [9,15] is a mapping from $\triangle$ to $\mathbb{R}^3$ defined by

$$\mathbf{b}(\mathbf{u}) = \sum_{|\mathbf{i}|=n} \mathbf{b}_{\mathbf{i}} B^n_{\mathbf{i}}(\mathbf{u}), \quad \mathbf{u} \in \triangle,$$

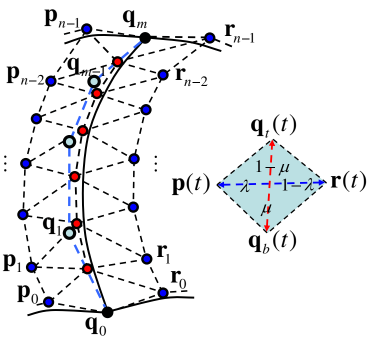


Fig. 1 The G^1 continuity constraints

where $\mathbf{i} = (i, j, k)$, $B^n_{\mathbf{i}}(\mathbf{u}) = \frac{n!}{i!j!k!}u^i v^j w^k$ and $\mathbf{b_i}$ are the control points. A G^1 *triangular spline surface* is composed of triangular Bézier patches that are in a one-to-one correspondence with the base mesh $\mathcal{M}$ and satisfy the G^1 continuity condition.

3.2 The G^1 continuity condition

Let two adjacent triangular Bézier patches of degree n share a common boundary Bézier curve $\mathbf{q}(t)$ of degree m,

$$\mathbf{q}(t) = \sum_{i=0}^{m} \mathbf{q}_i B^m_i(t) = \sum_{i=0}^{n} \mathbf{c}_i B^n_i(t), \quad n \geq m,$$

where $B^m_i(t)$ and $B^n_i(t)$ are the Bernstein polynomials, and $\mathbf{q}_i$ and $\mathbf{c}_i$ are the control points. Here, we should be aware of that a triangular patch of degree n may have boundaries of lower degree m, which eases satisfying the G^1 continuity condition.

The G^1 continuity constraints require that they have a shared and continuously varying tangent plane along $\mathbf{q}(t)$, and can be represented by the following equation [9]:

$$[1 - \lambda(t)]\mathbf{p}(t) + \lambda(t)\mathbf{r}(t) = [1 - \mu(t)]\mathbf{q}_b(t) + \mu(t)\mathbf{q}_t(t), \tag{1}$$

where $\mathbf{p}(t) = \sum_{i=0}^{n-1} \mathbf{p}_i B^{n-1}_i(t)$, $\mathbf{r}(t) = \sum_{i=0}^{n-1} \mathbf{r}_i B^{n-1}_i(t)$, $\mathbf{q}_b(t) = \sum_{i=0}^{m-1} \mathbf{q}_i B^{m-1}_i(t)$ and $\mathbf{q}_t(t) = \sum_{i=0}^{m-1} \mathbf{q}_{i+1} B^{m-1}_i(t)$. The $\mathbf{p}_i$ and $\mathbf{r}_i$ are the respective control points from the first-left and first-right row of the control net along this boundary curve, see Fig. 1.

3.3 Scalar weight functions

The terms $\lambda(t)$ and $\mu(t)$ in Eq. (1) are called *scalar weight functions*. Although any continuous functions can be used, we choose that $\lambda(t)$ and $\mu(t)$ are polynomials, as is typical of many other studies [5,7,9,24].

To write the G^1 continuity condition in terms of control points, we express the functions as linear combinations of Bernstein polynomials, as follows:

$$\begin{aligned}\lambda(t) = \sum_{i=0}^{r} \lambda_i B_i^r(t),\ 1-\lambda(t) = \sum_{i=0}^{r} \bar{\lambda}_i B_i^r(t),\ \mu(t)\\ = \sum_{i=0}^{s} \mu_i B_i^s(t),\ 1-\mu(t) = \sum_{i=0}^{s} \bar{\mu}_i B_i^s(t).\end{aligned} \tag{2}$$

Here r and s are the degrees of $\lambda(t)$ and $\mu(t)$.

If we let $r+n=s+m$ [i.e. the polynomial degree on each side of Eq. (1) is the same] and substitute Eqs. (2) into Eq. (1), then the G^1 continuity condition can be rewritten as

$$\sum_{i=0}^{k}(\bar{\lambda}_i \mathbf{p}_{k-i} + \lambda_i \mathbf{r}_{k-i})\binom{r}{i}\binom{n-1}{k-i} = \sum_{i=0}^{k}(\bar{\mu}_i \mathbf{q}_{k-i} + \mu_i \mathbf{q}_{k-i+1})\binom{s}{i}\binom{m-1}{k-i} \tag{3}$$

where $k=0,1,2,\ldots,r+n-1$. The details are explained in Appendix A.

Many possible configurations exist for r, n, s and m, for example, $\{r=1, s=4, m=4, n=7\}$, $\{r=2, s=4, m=4, n=6\}$, and so on. The choice of r, n, s and m will dramatically affect the properties of the triangular spline surface such as locality, singularity, and approximation power.

3.4 Minimum degree configuration

In this paper, our aim is to devise a minimum degree scheme for constructing triangular spline surfaces that are local and singularity-free. To achieve this (as will be explained in Sects. 4 and 5), we choose the configuration $\{r=1, s=4, m=3, n=6\}$. In this case, the Eq. (3) becomes

$$\begin{aligned}
(\bar{\lambda}_0\mathbf{p}_0 + \lambda_0\mathbf{r}_0) &= (\bar{\mu}_0\mathbf{q}_0 + \mu_0\mathbf{q}_1), && k=0\\
5\cdot(\bar{\lambda}_0\mathbf{p}_1 + \lambda_0\mathbf{r}_1) &= -(\bar{\lambda}_1\mathbf{p}_0 + \lambda_1\mathbf{r}_0) + 2\cdot(\bar{\mu}_0\mathbf{q}_1 + \mu_0\mathbf{q}_2)\\
&\quad +4\cdot(\bar{\mu}_1\mathbf{q}_0 + \mu_1\mathbf{q}_1), && k=1\\
10\cdot(\bar{\lambda}_0\mathbf{p}_2 + \lambda_0\mathbf{r}_2) &= -5\cdot(\bar{\lambda}_1\mathbf{p}_1 + \lambda_1\mathbf{r}_1) + (\bar{\mu}_0\mathbf{q}_2 + \mu_0\mathbf{q}_3)\\
&\quad +8\cdot(\bar{\mu}_1\mathbf{q}_1 + \mu_1\mathbf{q}_2) + 6\cdot(\bar{\mu}_2\mathbf{q}_0 + \mu_2\mathbf{q}_1), && k=2\\
10\cdot(\bar{\lambda}_0\mathbf{p}_3 + \lambda_0\mathbf{r}_3) + 10\cdot(\bar{\lambda}_1\mathbf{p}_2 + \lambda_1\mathbf{r}_2) &= 4\cdot(\bar{\mu}_1\mathbf{q}_2 + \mu_1\mathbf{q}_3) + 12\cdot(\bar{\mu}_2\mathbf{q}_1 + \mu_2\mathbf{q}_2)\\
&\quad +4\cdot(\bar{\mu}_3\mathbf{q}_0 + \mu_3\mathbf{q}_1), && k=3\\
10\cdot(\bar{\lambda}_1\mathbf{p}_3 + \lambda_1\mathbf{r}_3) &= -5\cdot(\bar{\lambda}_0\mathbf{p}_4 + \lambda_0\mathbf{r}_4) + 6\cdot(\bar{\mu}_2\mathbf{q}_2 + \mu_2\mathbf{q}_3)\\
&\quad +8\cdot(\bar{\mu}_3\mathbf{q}_1 + \mu_3\mathbf{q}_2) + (\bar{\mu}_4\mathbf{q}_0 + \mu_4\mathbf{q}_1), && k=4\\
5\cdot(\bar{\lambda}_1\mathbf{p}_4 + \lambda_1\mathbf{r}_4) &= -(\bar{\lambda}_0\mathbf{p}_5 + \lambda_0\mathbf{r}_5) + 4\cdot(\bar{\mu}_3\mathbf{q}_2 + \mu_3\mathbf{q}_3)\\
&\quad +2\cdot(\bar{\mu}_4\mathbf{q}_1 + \mu_4\mathbf{q}_2), && k=5\\
(\bar{\lambda}_1\mathbf{p}_5 + \lambda_1\mathbf{r}_5) &= (\bar{\mu}_4\mathbf{q}_2 + \mu_4\mathbf{q}_3), && k=6.
\end{aligned} \tag{4}$$

In the above equations, $\{\mathbf{q}_i\}_{i=0}^{3}$ and $\mathbf{p}_0$, $\mathbf{p}_5$, $\mathbf{r}_0$, $\mathbf{r}_5$ are given by the boundary curves network. By the equations for $k=0$ and $k=6$, we determine λ_0, λ_1, μ_0, μ_4. The remaining variables $\{\mathbf{p}_i\}_{i=1}^4$, $\{\mathbf{r}_i\}_{i=1}^4$, $\{\mu_i\}_{i=1}^3$ are unknowns which we are going to be solved for.

4 Selection of scalar weight functions

4.1 Solving G^1 continuity constraints

For each common boundary curve $\mathbf{q}(t)$ corresponding to one edge of the base mesh $\mathcal{M}$, the G^1 continuity constraints have the form of Eq. (3). If we collect all these equations and re-index all the control points, we obtain a global linear system. Such a system is not easily solved, and may not even be solvable because there may be more independent equations than unknowns.

Fortunately, we can avoid such a global system by carefully separating the G^1 continuity constraints into two types, that is vertex and edge G^1 constraints. For example, in Eqs. (4), the constraints with $k = 0, 1, 5, 6$ are the vertex G^1 constraints, and the constraints with $k = 2, 3, 4$ are the edge G^1 constraints.

The G^1 triangular spline surface can be constructed as follows:

1. For each vertex of the base mesh $\mathcal{M}$, collect all the related G^1 continuity constraints (i.e., vertex G^1 constraints) into a vertex G^1 continuity system and solve it. If the valence of vertex is n, then there is a linear system of n equation in n unknowns. The unknowns are control points related to the twist vectors at the corners of patches. For further details, refer to Sect. 5.1.
2. For each edge of the base mesh $\mathcal{M}$, collect all the related G^1 continuity constraints (i.e., edge G^1 constraints) into an edge G^1 continuity system and solve it. In the case of Eqs. (4), there is a linear system of three equations in four unknowns. The unknowns are called off-boundary control points of two adjacent patches. For further details, refer to Sect. 4.3.
3. In every triangular Bézier patch, the inner control points, which are not involved in the G^1 continuity constraints, can be determined by optimizing the local shape. There exist serval approaches for choosing them. One is to locate them so that resulting triangular Bézier patches have quintic precision, just like Loop [18] did. Another approach is to solve least-squares fitting problems, as done by Tong and Kim [32].

For an illustration, see Fig. 2. Since the scheme proposed in this paper uses a method for solving the G^1 continuity constraints similar to the method described in a previous paper [32], we omit some details and refer the interested reader to that earlier paper. In the following sections, we focus on presenting the reasoning as to why our choice of degree is minimal, the method for constructing a boundary Bézier curves network and the adaptive subdivision strategy of the base mesh.

4.2 Degree of the scalar weight function $\lambda(t)$

As assumed in Sect. 3, r, s, m and n should satisfy $r + n = s + m$. If the value of $s + m$ is fixed, then increasing r will lead to a smaller value of n. However, this will

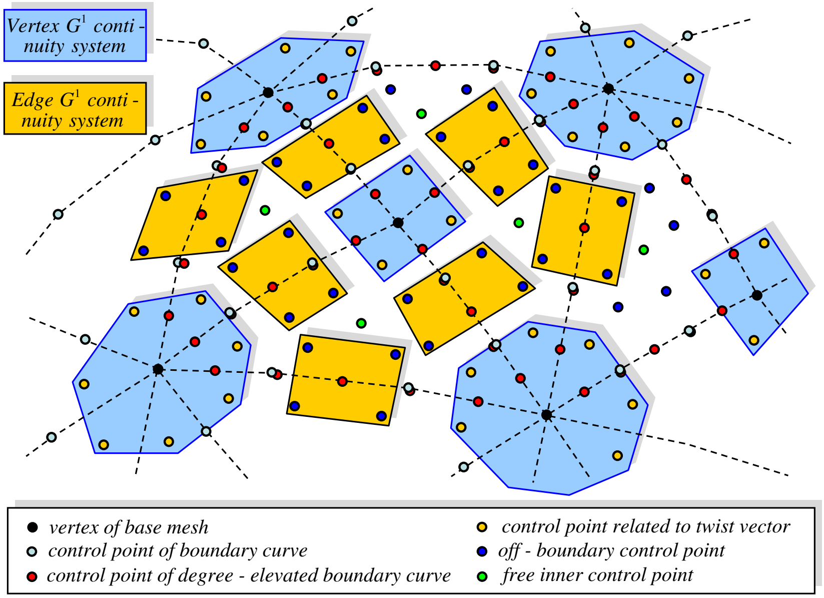

Vertex G^1 continuity system
Edge G^1 continuity system
● vertex of base mesh
○ control point of boundary curve
● control point of degree - elevated boundary curve
● control point related to twist vector
● off - boundary control point
● free inner control point

Fig. 2 Vertex and edge G^1 continuity system

result in a global scheme. For example, if we increase r by 1 in the configuration $\{r = 1, s = 4, m = 3, n = 6\}$, we have $\{r = 2, s = 4, m = 3, n = 5\}$, and Eq. (3) becomes

$$
\begin{aligned}
(\bar{\lambda}_0\mathbf{p}_0 + \lambda_0\mathbf{r}_0) &= (\bar{\mu}_0\mathbf{q}_0 + \mu_0\mathbf{q}_1), && k = 0\\
4\cdot(\bar{\lambda}_0\mathbf{p}_1 + \lambda_0\mathbf{r}_1) + 2\cdot(\bar{\lambda}_1\mathbf{p}_0 + \lambda_1\mathbf{r}_0) &= 2\cdot(\bar{\mu}_0\mathbf{q}_1 + \mu_0\mathbf{q}_2) + 4\cdot(\bar{\mu}_1\mathbf{q}_0 + \mu_1\mathbf{q}_1), && k = 1\\
6\cdot(\bar{\lambda}_0\mathbf{p}_2 + \lambda_0\mathbf{r}_2) + 8\cdot(\bar{\lambda}_1\mathbf{p}_1 + \lambda_1\mathbf{r}_1) &= -(\bar{\lambda}_2\mathbf{p}_0 + \lambda_2\mathbf{r}_0) + (\bar{\mu}_0\mathbf{q}_2 + \mu_0\mathbf{q}_3)\\
&\quad +8\cdot(\bar{\mu}_1\mathbf{q}_1 + \mu_1\mathbf{q}_2) + 6\cdot(\bar{\mu}_2\mathbf{q}_0 + \mu_2\mathbf{q}_1), && k = 2\\
12\cdot(\bar{\lambda}_1\mathbf{p}_2 + \lambda_1\mathbf{r}_2) &= -4\cdot(\bar{\lambda}_0\mathbf{p}_3 + \lambda_0\mathbf{r}_3) - 4\cdot(\bar{\lambda}_2\mathbf{p}_1 + \lambda_2\mathbf{r}_1)\\
&\quad +4\cdot(\bar{\mu}_1\mathbf{q}_2 + \mu_1\mathbf{q}_3) + 12\cdot(\bar{\mu}_2\mathbf{q}_1 + \mu_2\mathbf{q}_2)\\
&\quad +4\cdot(\bar{\mu}_3\mathbf{q}_0 + \mu_3\mathbf{q}_1), && k = 3\\
6\cdot(\bar{\lambda}_2\mathbf{p}_2 + \lambda_2\mathbf{r}_2) + 8\cdot(\bar{\lambda}_1\mathbf{p}_3 + \lambda_1\mathbf{r}_3) &= -\cdot(\bar{\lambda}_0\mathbf{p}_4 + \lambda_0\mathbf{r}_4) + 6\cdot(\bar{\mu}_2\mathbf{q}_2 + \mu_2\mathbf{q}_3)\\
&\quad +8\cdot(\bar{\mu}_3\mathbf{q}_1 + \mu_3\mathbf{q}_2) + (\bar{\mu}_4\mathbf{q}_0 + \mu_4\mathbf{q}_1), && k = 4\\
4\cdot(\bar{\lambda}_2\mathbf{p}_3 + \lambda_2\mathbf{r}_3) + 2\cdot(\bar{\lambda}_1\mathbf{p}_4 + \lambda_1\mathbf{r}_4) &= 4\cdot(\bar{\mu}_3\mathbf{q}_2 + \mu_3\mathbf{q}_3) + 2\cdot(\bar{\mu}_4\mathbf{q}_1 + \mu_4\mathbf{q}_2), && k = 5\\
(\bar{\lambda}_2\mathbf{p}_4 + \lambda_2\mathbf{r}_4) &= (\bar{\mu}_4\mathbf{q}_2 + \mu_4\mathbf{q}_3), && k = 6.
\end{aligned}
\tag{5}
$$

Because λ_1 appears in the equations for $k = 1$ and $k = 5$, we can not separate the vertex G^1 constraints and the edge G^1 constraints, which in turn would lead to a global system of equations. Hence, we make the choice that $\lambda(t)$ is linear.

4.3 Degree of the scalar weight function $\mu(t)$

Another problem is the minimum degree of the scalar weight function $\mu(t)$. Because $r + n = s + m$, we have $n = s + m - r$. Thus, decreasing s will lead to a smaller value of n, but this may result in singular edge G^1 continuity systems. For example, if we decrease s by 1 in the configuration $\{r = 1, s = 4, m = 3, n = 6\}$, we have $\{r = 1, s = 3, m = 3, n = 5\}$, and Eq. (3) becomes

$$\begin{aligned}
(\bar{\lambda}_0\mathbf{p}_0 + \lambda_0\mathbf{r}_0) &= (\bar{\mu}_0\mathbf{q}_0 + \mu_0\mathbf{q}_1), && k=0\\
4\cdot(\bar{\lambda}_0\mathbf{p}_1 + \lambda_0\mathbf{r}_1) &= -(\bar{\lambda}_1\mathbf{p}_0 + \lambda_1\mathbf{r}_0) + 2\cdot(\bar{\mu}_0\mathbf{q}_1 + \mu_0\mathbf{q}_2)\\
&\quad +3\cdot(\bar{\mu}_1\mathbf{q}_0 + \mu_1\mathbf{q}_1), && k=1\\
6\cdot(\bar{\lambda}_0\mathbf{p}_2 + \lambda_0\mathbf{r}_2) &= -4\cdot(\bar{\lambda}_1\mathbf{p}_1 + \lambda_1\mathbf{r}_1) + (\bar{\mu}_0\mathbf{q}_2 + \mu_0\mathbf{q}_3)\\
&\quad +6\cdot(\bar{\mu}_1\mathbf{q}_1 + \mu_1\mathbf{q}_2) + 3\cdot(\bar{\mu}_2\mathbf{q}_0 + \mu_2\mathbf{q}_1), && k=2\\
6\cdot(\bar{\lambda}_1\mathbf{p}_2 + \lambda_1\mathbf{r}_2) &= -4\cdot(\bar{\lambda}_0\mathbf{p}_3 + \lambda_0\mathbf{r}_3) + 3\cdot(\bar{\mu}_1\mathbf{q}_2 + \mu_1\mathbf{q}_3)\\
&\quad +6\cdot(\bar{\mu}_2\mathbf{q}_1 + \mu_2\mathbf{q}_2) + (\bar{\mu}_3\mathbf{q}_0 + \mu_3\mathbf{q}_1), && k=3\\
4\cdot(\bar{\lambda}_1\mathbf{p}_3 + \lambda_1\mathbf{r}_3) &= -(\bar{\lambda}_0\mathbf{p}_4 + \lambda_0\mathbf{r}_4) + 3\cdot(\bar{\mu}_2\mathbf{q}_2 + \mu_2\mathbf{q}_3)\\
&\quad +2\cdot(\bar{\mu}_3\mathbf{q}_1 + \mu_3\mathbf{q}_2), && k=4\\
(\bar{\lambda}_1\mathbf{p}_4 + \lambda_1\mathbf{r}_4) &= (\bar{\mu}_3\mathbf{q}_2 + \mu_3\mathbf{q}_3), && k=5.
\end{aligned} \tag{6}$$

In this case, the edge G^1 continuity system can be written as follows:

$$\begin{bmatrix} 6\bar{\lambda}_0 & 6\lambda_0 \\ 6\bar{\lambda}_1 & 6\lambda_1 \end{bmatrix}\begin{bmatrix} \mathbf{p}_2 \\ \mathbf{r}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{rhs}_1 \\ \mathbf{rhs}_2 \end{bmatrix},$$

where

$$\begin{aligned}
\mathbf{rhs}_1 &= -4\cdot(\bar{\lambda}_1\mathbf{p}_1 + \lambda_1\mathbf{r}_1) + (\bar{\mu}_0\mathbf{q}_2 + \mu_0\mathbf{q}_3) + 6\cdot(\bar{\mu}_1\mathbf{q}_1 + \mu_1\mathbf{q}_2) + 3\cdot(\bar{\mu}_2\mathbf{q}_0 + \mu_2\mathbf{q}_1),\\
\mathbf{rhs}_2 &= -4\cdot(\bar{\lambda}_0\mathbf{p}_3 + \lambda_0\mathbf{r}_3) + 3\cdot(\bar{\mu}_1\mathbf{q}_2 + \mu_1\mathbf{q}_3) + 6\cdot(\bar{\mu}_2\mathbf{q}_1 + \mu_2\mathbf{q}_2) + (\bar{\mu}_3\mathbf{q}_0 + \mu_3\mathbf{q}_1).
\end{aligned}$$

If $\lambda_0 = \lambda_1$, then $rank\left(\begin{bmatrix} 6\bar{\lambda}_0 & 6\lambda_0 \\ 6\bar{\lambda}_1 & 6\lambda_1 \end{bmatrix}\right) = 1$. For the system to be solvable, the right-hand-side vectors must satisfy the condition $\mathbf{rhs}_1 = \mathbf{rhs}_2$, and this is equivalent to the following equation:

$$(\mu_0 - 3\mu_1 + 3\mu_2 - \mu_3)(-\mathbf{q}_0 + 3\mathbf{q}_1 - 3\mathbf{q}_2 + \mathbf{q}_3) = \mathbf{0}.$$

Since the coefficients $\{\mu_i\}_{i=0}^{3}$ are determined by the vertex G^1 continuity system, and the control points $\{\mathbf{q}_i\}_{i=0}^{3}$ of the boundary curve are given, this equation cannot always be satisfied. One more degree of freedom for $\mu(t)$ is needed.

In fact, in the case $\{r = 1, s = 4, m = 3, n = 6\}$, the edge G^1 continuity system can be written as follows:

$$\begin{bmatrix} 10\bar{\lambda}_0 & 10\lambda_0 & & \\ 10\bar{\lambda}_1 & 10\lambda_1 & 10\bar{\lambda}_0 & 10\lambda_0 \\ & & 10\bar{\lambda}_1 & 10\lambda_1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_2 \\ \mathbf{r}_2 \\ \mathbf{p}_3 \\ \mathbf{r}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{rhs}_1 \\ \mathbf{rhs}_2 \\ \mathbf{rhs}_3 \end{bmatrix}, \tag{7}$$

where

$$\begin{aligned} \mathbf{rhs}_1 &= -5 \cdot (\bar{\lambda}_1 \mathbf{p}_1 + \lambda_1 \mathbf{r}_1) + (\bar{\mu}_0 \mathbf{q}_2 + \mu_0 \mathbf{q}_3) + 8 \cdot (\bar{\mu}_1 \mathbf{q}_1 + \mu_1 \mathbf{q}_2) + 6 \cdot (\bar{\mu}_2 \mathbf{q}_0 + \mu_2 \mathbf{q}_1), \\ \mathbf{rhs}_2 &= 4 \cdot (\bar{\mu}_1 \mathbf{q}_2 + \mu_1 \mathbf{q}_3) + 12 \cdot (\bar{\mu}_2 \mathbf{q}_1 + \mu_2 \mathbf{q}_2) + 4 \cdot (\bar{\mu}_3 \mathbf{q}_0 + \mu_3 \mathbf{q}_1), \\ \mathbf{rhs}_3 &= -5 \cdot (\bar{\lambda}_0 \mathbf{p}_4 + \lambda_0 \mathbf{r}_4) + 6 \cdot (\bar{\mu}_2 \mathbf{q}_2 + \mu_2 \mathbf{q}_3) + 8 \cdot (\bar{\mu}_3 \mathbf{q}_1 + \mu_3 \mathbf{q}_2) + (\bar{\mu}_4 \mathbf{q}_0 + \mu_4 \mathbf{q}_1). \end{aligned} \tag{8}$$

If $\lambda_0 = \lambda_1$, to make the edge G^1 continuity system consistent, the right-hand-side vectors must satisfy the condition $\mathbf{rhs}_1 - \mathbf{rhs}_2 + \mathbf{rhs}_3 = \mathbf{0}$. This is equivalent to the following equation:

$$(\mu_0 - 4\mu_1 + 6\mu_2 - 4\mu_3 + \mu_4)(-\mathbf{q}_0 + 3\mathbf{q}_1 - 3\mathbf{q}_2 + \mathbf{q}_3) = \mathbf{0}. \tag{9}$$

This equation can be satisfied by choosing

$$\mu_2 = \frac{-\mu_0 + 4\mu_1 + 4\mu_3 - \mu_4}{6}.$$

Therefore, the minimum degree of $\mu(t)$ is 4.

To solve the off-boundary control points, i.e. $\mathbf{p}_2, \mathbf{r}_2, \mathbf{p}_3, \mathbf{r}_3$, the equality constrained quadratic programs similar to those presented in [32] can be used.

5 Construction of the boundary curves network

5.1 Singularity of vertex G^1 continuity systems

Let $\mathbf{o}$ be a vertex of the triangular base mesh $\mathcal{M}$, at which the n boundary Bézier curves $\{\mathbf{q}^i(t)\}_{i=1}^{n}$ meet. The control points of cubic boundary curve $\mathbf{q}^i(t)$ are $\mathbf{q}_0^i$, $\mathbf{q}_1^i$, …, $\mathbf{q}_3^i$ and the elevated control points are $\mathbf{c}_0^i$, $\mathbf{c}_1^i$, …, $\mathbf{c}_6^i$. The control points $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n$ related to the twist vectors at the corner $\mathbf{o}$ must satisfy the vertex G^1 constraints. In the case of Eqs. (4), the vertex G^1 continuity system can be written as follows:

$$5\begin{bmatrix} \bar{\lambda}_0^1 & \lambda_0^1 & 0 & \cdots & 0 & 0 \\ 0 & \bar{\lambda}_0^2 & \lambda_0^2 & \cdots & 0 & 0 \\ 0 & 0 & \bar{\lambda}_0^3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \bar{\lambda}_0^{n-1} & \lambda_0^{n-1} \\ \lambda_0^n & 0 & 0 & \cdots & 0 & \bar{\lambda}_0^n \end{bmatrix} \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \mathbf{t}_3 \\ \vdots \\ \mathbf{t}_{n-1} \\ \mathbf{t}_n \end{bmatrix} = \begin{bmatrix} \mathbf{rhs}_1 \\ \mathbf{rhs}_2 \\ \mathbf{rhs}_3 \\ \vdots \\ \mathbf{rhs}_{n-1} \\ \mathbf{rhs}_n \end{bmatrix}, \tag{10}$$

where $\{\lambda_0^i\}_{i=1}^n$ are the coefficients of the scalar weight function $\lambda(t)$ that satisfy the G^1 continuity constraints corresponding to $k = 0$ or 6:

$$\bar{\lambda}_0^i \mathbf{c}_1^{i-1} + \lambda_0^i \mathbf{c}_1^{i+1} = \bar{\mu}_0^i \mathbf{o} + \mu_0^i \mathbf{q}_1^i,$$

and $\{\mathbf{rhs}_i\}_{i=1}^n$ are the residual vectors:

$$\mathbf{rhs}_i = -\left(\bar{\lambda}_1^i \mathbf{c}_1^{i-1} + \lambda_1^i \mathbf{c}_1^{i+1}\right) + 2\left(\bar{\mu}_0^i \mathbf{q}_1^i + \mu_0^i \mathbf{q}_2^i\right) + 4\left(\bar{\mu}_1^i \mathbf{o} + \mu_1^i \mathbf{q}_1^i\right),$$

where the subscript $i := (n+i) \bmod n$ is applied.

Let the coefficient matrix of the linear system of Eq. (10) be $\mathbf{A}$. The determinant of $\mathbf{A}$ is

$$\det \mathbf{A} = \begin{cases} 5^n \cdot 2 \prod_{i=1}^n \lambda_0^i, & \text{if } n \text{ is odd,} \\ 0, & \text{if } n \text{ is even.} \end{cases}$$

Without a loss of generality, we can assume that $\lambda_0^i \neq 0$ for $i = 1, 2, \ldots, n$. Then we have

$$\operatorname{rank}(\mathbf{A}) = \begin{cases} n, & \text{if } n \text{ is odd,} \\ n-1, & \text{if } n \text{ is even.} \end{cases}$$

When n is even, to have solutions, the $\{\mathbf{rhs}_i\}_{i=1}^n$ must satisfy

$$(-1)\bar{\lambda}_0^2\bar{\lambda}_0^3 \cdots \bar{\lambda}_0^n \mathbf{rhs}_1 + (-1)^2 \lambda_0^1 \bar{\lambda}_0^3 \cdots \bar{\lambda}_0^n \mathbf{rhs}_2 + \cdots + (-1)^n \lambda_0^1 \lambda_0^2 \cdots \lambda_0^{n-1} \mathbf{rhs}_n = \mathbf{0}. \tag{11}$$

This is a necessary and sufficient condition, which is usually called the vertex enclosure constraint, or twist compatibility condition [18,24].

Although Eq. (11) is derived from our special scheme, note that the existence of such a constraint does not depend on the degree of the patch being used, but on a principle that is inherent whenever 3, 4, or N-sided patches (of either the polynomial or rational type) are used to join an even number of patches at vertices.

For the sake of obtaining a pleasing shape, the coefficients $\{\mu_1^i\}_{i=1}^n$ must be variable and serve as shape parameters. To determine the control points $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n$, we use a similar method presented in [32], i.e. solve Eq. (10) through equivalent tangent and normal component subproblems.

5.2 Degree of the boundary curve $\mathbf{q}(t)$

To make the linear system of Eq. (10) consistent, we must carefully construct the boundary curves to satisfy the vertex enclosure constraint of Eq. (11). A sufficient condition to ensure this is contained in the following result, which was proposed by Peters [24]. We restate it as follows:

Theorem 1 (C^2 consistent boundary curves network) *Let the n boundary Bézier curves $\{\mathbf{q}^i(t)\}_{i=1}^n$ meet at vertex* **o**, *and share a common tangent plane with normal* **N**. *If there exists a curvature tensor* **T** *satisfying*

$$\mathbf{N} \cdot \mathbf{q}_{tt}^i(t) = \mathbf{q}_t^i(t)^T \mathbf{T} \mathbf{q}_t^i(t) \text{ at } t = 0, \quad i = 1, 2, \ldots, n, \tag{12}$$

then the vertex G^1 continuity system is consistent.

The geometric interpretation of Eq. (12) is normal curvature interpolation. Thus, we can fulfill this condition by adopting a geometric Hermite interpolation (GHI) technique. In comparison to the method of de Boor et al. [4], we use normal curvature in our GHI in place of curvature. Although Schaback [31] asserted that the degree of the polynomial curves for general GHI must be at least 5 to handle all possible geometrical situations, by using local subdivision we are able to use degree 3 in our GHI technique. This will be shown in the following section.

5.3 Geometric Hermite interpolation

For each edge e of the base mesh $\mathcal{M}$, we construct a boundary curve $\mathbf{q}(t)$ to interpolate the given position, normal, and normal curvature at the vertices. In the case $\{r = 1, s = 4, m = 3, n = 6\}$, this curve has the form $\mathbf{q}(t) = \sum_{i=0}^3 \mathbf{q}_i B_i^3(t)$. From the nature of the interpolation, the control points $\mathbf{q}_0$ and $\mathbf{q}_3$ coincide with the two vertices of edge e. Two unit normal vectors will be denoted by $\mathbf{N}_0$ and $\mathbf{N}_1$, two pairs of principal curvatures by $\kappa_{\min}^0, \kappa_{\max}^0$ and $\kappa_{\min}^1, \kappa_{\max}^1$, and two pairs of principal directions by $\mathbf{t}_{\min}^0, \mathbf{t}_{\max}^0$ and $\mathbf{t}_{\min}^1, \mathbf{t}_{\max}^1$, at $\mathbf{q}_0$ and $\mathbf{q}_3$ (Fig. 3).

Then the curvature tensors can be written as

$$\mathbf{T}_0 = \begin{bmatrix} \kappa_{\max}^0 & 0 & 0 \\ 0 & \kappa_{\min}^0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{T}_1 = \begin{bmatrix} \kappa_{\max}^1 & 0 & 0 \\ 0 & \kappa_{\min}^1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

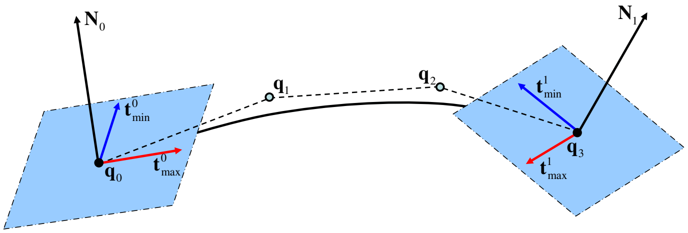

$\mathbf{N}_0$
$\mathbf{t}^0_{\text{min}}$
$\mathbf{q}_0$
$\mathbf{t}^0_{\text{max}}$
$\mathbf{q}_1$
$\mathbf{q}_2$
$\mathbf{N}_1$
$\mathbf{t}^1_{\text{min}}$
$\mathbf{q}_3$
$\mathbf{t}^1_{\text{max}}$

Fig. 3 Geometric Hermite interpolation

Let the unknown control points $\mathbf{q}_1$ and $\mathbf{q}_2$ be given by

$$\mathbf{q}_1 = \mathbf{q}_0 + l_0\mathbf{t}^0_{\text{max}} + m_0\mathbf{t}^0_{\text{min}},\ \ \mathbf{q}_2 = \mathbf{q}_3 + l_1\mathbf{t}^1_{\text{max}} + m_1\mathbf{t}^1_{\text{min}}.$$

With some calculation, it can be shown that our GHI problem may be restated as follows:

$$\begin{aligned} &3\left(l_0^2\kappa^0_{\text{max}} + m_0^2\kappa^0_{\text{min}}\right) - 2\left(l_1\mathbf{t}^1_{\text{max}} + m_1\mathbf{t}^1_{\text{min}}\right)\cdot\mathbf{N}_0 + 2(\mathbf{q}_0 - \mathbf{q}_3)\cdot\mathbf{N}_0 = 0,\\ &3\left(l_1^2\kappa^1_{\text{max}} + m_1^2\kappa^1_{\text{min}}\right) - 2\left(l_0\mathbf{t}^0_{\text{max}} + m_0\mathbf{t}^0_{\text{min}}\right)\cdot\mathbf{N}_1 + 2(\mathbf{q}_3 - \mathbf{q}_0)\cdot\mathbf{N}_1 = 0. \end{aligned} \tag{13}$$

Equations (13) are a pair of quadratic equations in four unknown variables l_0, l_1, m_0, m_1 and could be solved iteratively by a local SQP method [22] which results in a series of linear systems. In practice, we find that just 2 to 4 iterations are usually enough with the stopping criterion $\epsilon = 10^{-6}$. In addition, this quadratic system is solvable in most instances because the number of unknowns is greater than the number of equations. However, there exist some extraordinary cases, such as those presented in [28]. In such cases, local subdivision of the base mesh is required to remedy the singularity. In fact, if we let $\mathbf{q}_1 = \mathbf{q}_0 + a_0\mathbf{t}_0$ and $\mathbf{q}_2 = \mathbf{q}_3 + a_1\mathbf{t}_1$ (i.e. fix tangent directions), then Eqs. (13) reduce to a system of quadratic equations in two unknown variables as follows:

$$\begin{aligned} n_0a_0^2 + m_0a_1 + l_0 = 0,\\ n_1a_1^2 + m_1a_0 + l_1 = 0, \end{aligned} \tag{14}$$

where $n_0 = 3(\cos^2\theta_0\cdot\kappa^0_{\text{max}} + \sin^2\theta_0\cdot\kappa^0_{\text{min}})$, $m_0 = -2(\mathbf{t}_1\cdot\mathbf{N}_0)$, $l_0 = 2(\mathbf{q}_0 - \mathbf{q}_3)\cdot\mathbf{N}_0$, $\cos\theta_0 = \mathbf{t}_0\cdot\mathbf{t}^0_{\text{max}}$, and $n_1 = 3(\cos^2\theta_1\cdot\kappa^1_{\text{max}} + \sin^2\theta_1\cdot\kappa^1_{\text{min}})$, $m_1 = -2(\mathbf{t}_0\cdot\mathbf{N}_1)$, $l_0 = 2(\mathbf{q}_3 - \mathbf{q}_0)\cdot\mathbf{N}_1$, $\cos\theta_1 = \mathbf{t}_1\cdot\mathbf{t}^1_{\text{max}}$. The solvability of Eqs. (14) was discussed by Sabin [28]. Applying similar analysis on asymptotic behavior presented in [4] to Eqs. (14) or more generally Eqs. (13), we make sure that local subdivision can remove the singularity. To solve Eqs. (13), a similar local SQP method presented in [32] may be used. If we use fixed tangent directions with $\mathbf{q}_1 = \mathbf{q}_0 + a_0\mathbf{t}_0$ and $\mathbf{q}_2 = \mathbf{q}_3 + a_1\mathbf{t}_1$, Eqs. (14) can also be solved by an equivalent quartic equation.

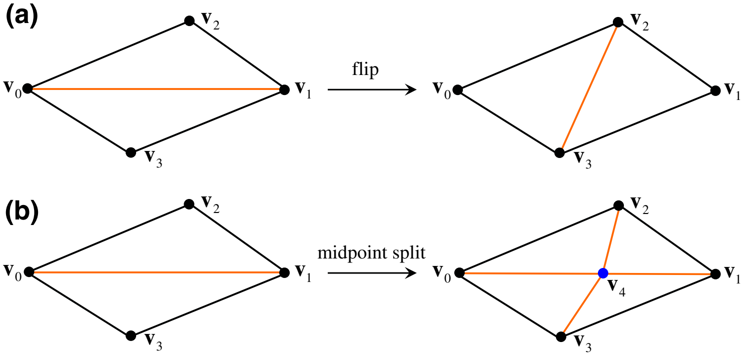


Fig. 4 Edge-based operators: **a** flip; **b** midpoint split

5.4 Adaptive subdivision strategy

In the process of constructing boundary curves network, some undesirable situations may occur:

- The quadratic system Eqs. (13) has no solution, i.e. an extraordinary case;
- Even if there are solutions, the quality of the resulting boundary curve may not be satisfying, such as the fitting error exceeds the threshold or unwanted oscillations appear.

To deal with these problems, the relevant edges of base mesh must be subdivided locally. To this aim, we use edge-based operators that include flip and midpoint split operations,[1] see Fig. 4.

In the situation of this paper, we prefer flip to midpoint split since the previous one does not increase the number of faces in the base mesh. But, there are three exceptional cases:

(a) Topology constraint: the flip operation results in two new triangles that coincide with each other, see Fig. 5a.
(b) Normal constraint: the flip operation results in a new triangle that has the inverse orientations, see Fig. 5b.
(c) Angle ratio constraint: to maintain the angle ratio, it is required that the minimum (maximum) angle of two new triangles is larger (smaller) than original one, see Fig. 5c.

And so, we must perform midpoint split operation carefully in order to maintain the shape quality of the base mesh, such as the avoidance of skinny triangles. Let the candidate edge is $\overline{\mathbf{v}_0\mathbf{v}_1}$, and the two angles opposite to it are θ_2^l and θ_2^r, see Fig. 6. The rules of midpoint split are as follows:

(a) If $\theta_2^l \geq 60°$, $\theta_2^r \geq 60°$, split edge $\overline{\mathbf{v}_0\mathbf{v}_1}$.
(b) Else if $\theta_2^l < 60°$, $\theta_2^r \geq 60°$, then

[1] In the application of approximating implicit surfaces, the projection process presented in [32] is used to located the split-point on the input surface.

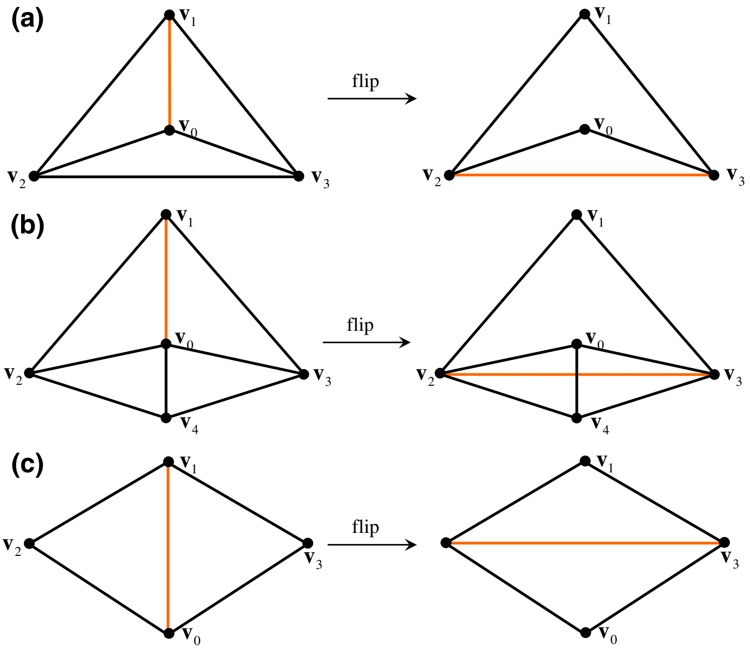
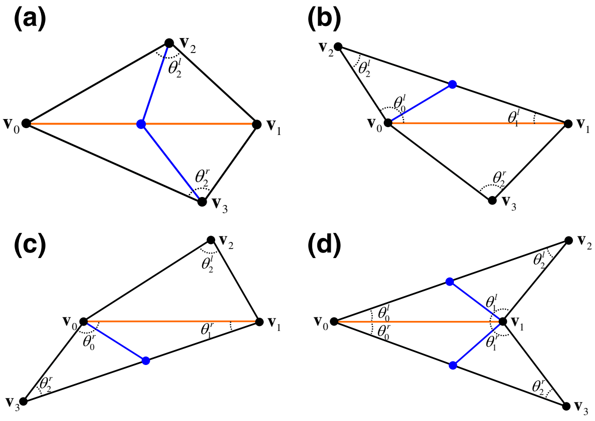


Fig. 5 Exceptional cases of edge flip: **a** topology constraint; **b** normal constraint; **c** angle ratio constraint

Fig. 6 Four cases of midpoint split: **a** $\theta_2^l \geq 60°, \theta_2^r \geq 60°$; **b** $\theta_2^l < 60°, \theta_2^r \geq 60°$; **c** $\theta_2^l \geq 60°, \theta_2^r < 60°$; **d** $\theta_2^l < 60°, \theta_2^r < 60°$



- if $\theta_0^l > \theta_1^l$, split edge $\overline{\mathbf{v}_1\mathbf{v}_2}$;
- else split edge $\overline{\mathbf{v}_0\mathbf{v}_2}$.

(c) Else if $\theta_2^l \geq 60°, \theta_2^r < 60°$, then
  - if $\theta_0^r > \theta_1^r$, split edge $\overline{\mathbf{v}_1\mathbf{v}_3}$;
  - else split edge $\overline{\mathbf{v}_0\mathbf{v}_3}$.

(d) Else then
  - if $\theta_0^l > \theta_1^l$, split edge $\overline{\mathbf{v}_1\mathbf{v}_2}$;
  - else split edge $\overline{\mathbf{v}_0\mathbf{v}_2}$,

and

- if $\theta_0^r > \theta_1^r$, split edge $\overline{\mathbf{v}_1\mathbf{v}_3}$;
- else split edge $\overline{\mathbf{v}_0\mathbf{v}_3}$.

And for the (b)(c)(d) cases, the edge $\overline{\mathbf{v}_0\mathbf{v}_1}$ must be considered for further refinement.

To accomplish the above subdivision strategy, a recursive procedure can be used. We implemented it and observed that the subdivided base mesh and the associated boundary curves network achieve good shape.

6 Application to the approximation of an implicit surface

In this section, we present an application of our G^1 triangular spline surfaces to the approximation of implicit surfaces.

6.1 Application framework

In this application, an implicit surface S defined by an implicit function f and a regular value c, i.e., $S = f^{-1}(c)$, is given. The objective is to approximate S by a G^1 triangular spline surface S'. We briefly describe the steps of a generic procedure as follows:

1. Generating the base mesh $\mathcal{M}$: Polygonize the implicit surface S with Bloothmenthal's polygonizer [1] to obtain a triangular mesh $\mathcal{P}$. Decimate $\mathcal{P}$ with the QEM algorithm [11,12] to get a coarse mesh $\mathcal{P}'$. The vertices of $\mathcal{P}'$ are then projected onto S to obtain the base mesh $\mathcal{M}$.
2. Constructing the boundary curves network $\mathcal{C}$ with local subdivision: for each edge of $\mathcal{M}$, construct a Bézier curve by solving the GHI presented in Sect. 5.3. Then perform local subdivisions of the edges in $\mathcal{M}$ based on approximation error and other criteria pertaining to shape quality.
3. Fitting a G^0 triangular surface: map each face of $\mathcal{M}$ onto S by projection, and fit the resulting image with a triangular Bézier patch that interpolates the corresponding boundary curves from $\mathcal{C}$.
4. Solving for the G^1 triangular spline surface: relocate the control points of the G^0 triangular surface to make them satisfy the G^1 constraint. To accomplish this, use the method described in Sect. 4.1.

For the details of each step, refer to [32]. This technique is capable of achieving a high order of precision at a relatively low computational cost, and its implementation is straightforward since it involves only local and linear solvers.

6.2 Results and discussion

We have implemented the method described in the previous section, and have tested it on a range of increasingly complex implicit surfaces. Here, because of space limit we present only three examples.

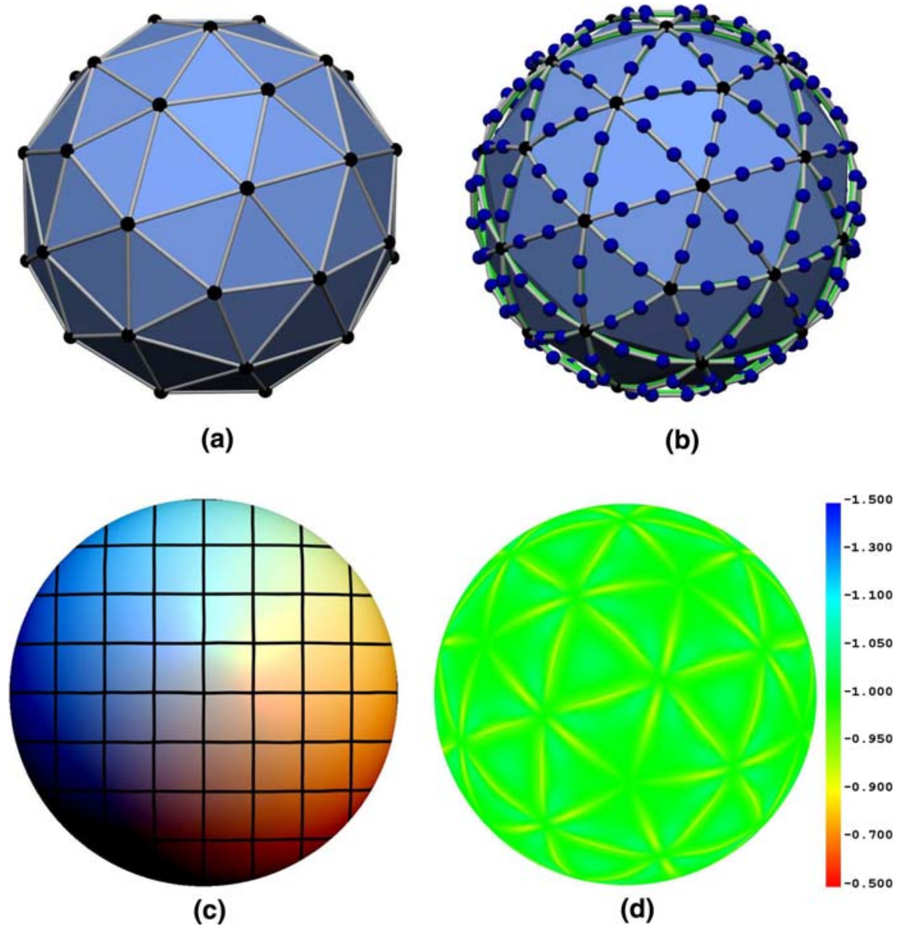


Fig. 7 Approximation of a unit sphere: $f(x, y, z) = x^2 + y^2 + z^2 - 1$. **a** Triangular base mesh, **b** Boundary curves network, **c** Reflection lines, **d** Mean curvature

Figure 7 depicts the approximation of a unit sphere. The initial base mesh is shown in Fig. 7a. In this mesh, the vertex valence is 5 or 6. The boundary curves network is displayed in Fig. 7b. The resulting G^1 triangular spline surface and reflection lines[2] are shown in Fig. 7c. The mean curvature plot is depicted in Fig. 7d and reveals that the approximation of mean curvature of unit sphere is well, except in the areas directly adjacent to the boundary curves network. This phenomenon is common among boundary curves schemes, and has been discussed by [19].

A somewhat more complex example involving the Bretzel surface is shown in Fig. 8. The genus of this surface is 5. The initial base mesh and the boundary curves network are depicted in Fig. 8a and b. The number of triangles in the initial base mesh

[2] Under the assumption that the light source is infinitely far away from the object, sphere mapping (i.e. environment mapping) is used to display reflection lines approximately [23].

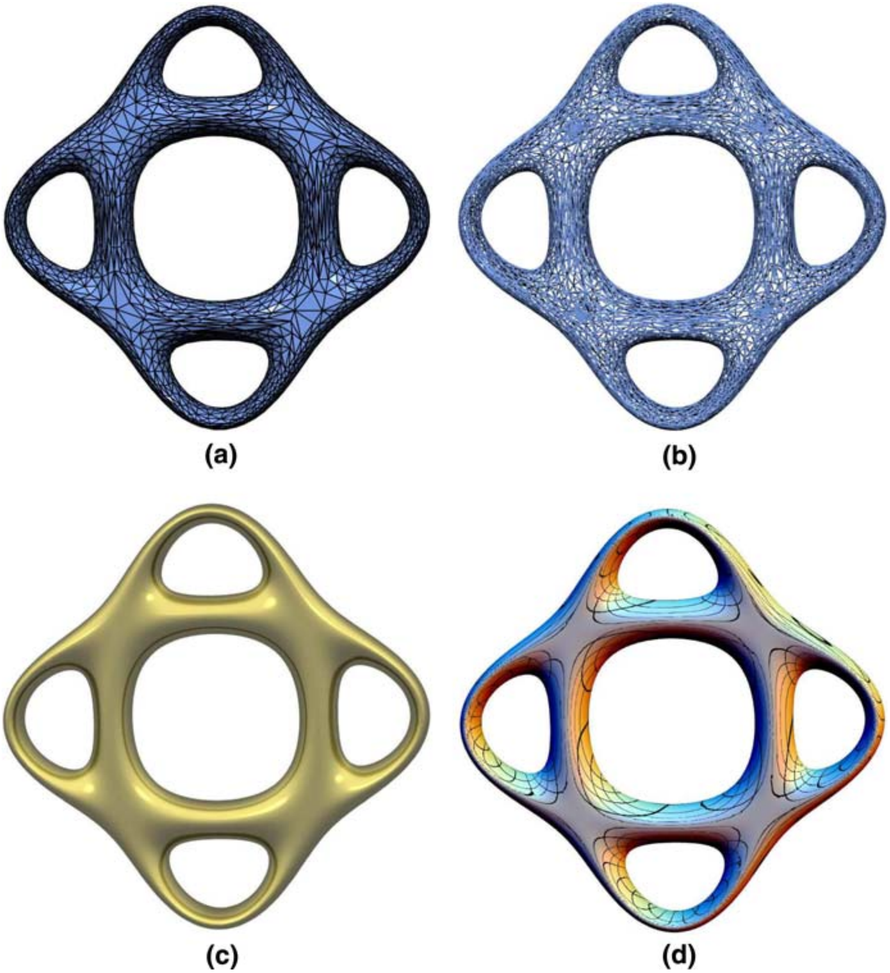


Fig. 8 Approximation of the Bretzel surface: $f(x, y, z) = [(x^2+y^2/4-1)\cdot(x^2/4+y^2-1)]^2+z^2-0.075$. **a** Triangular base mesh, **b** Boundary curves network, **c** G^1 triangular spline surface, **d** Reflection lines

is 5,000. During the construction of the boundary curves network, local subdivisions of the edges were performed. The final surface contains 5,636 triangular Bézier patches. Figure 8c and d confirm the visual quality and smoothness of the final G^1 triangular spline surface. Another example on the approximation of Jack surface is shown in Fig. 9. The number of triangles in the initial base mesh is 4,500, and the number of triangular Bézier patches in the final surface is 5,520.

7 Conclusion and future work

In this paper, we have presented a scheme for constructing G^1 triangular spline surfaces of arbitrary topological type, which can interpolate the given positions, normals,

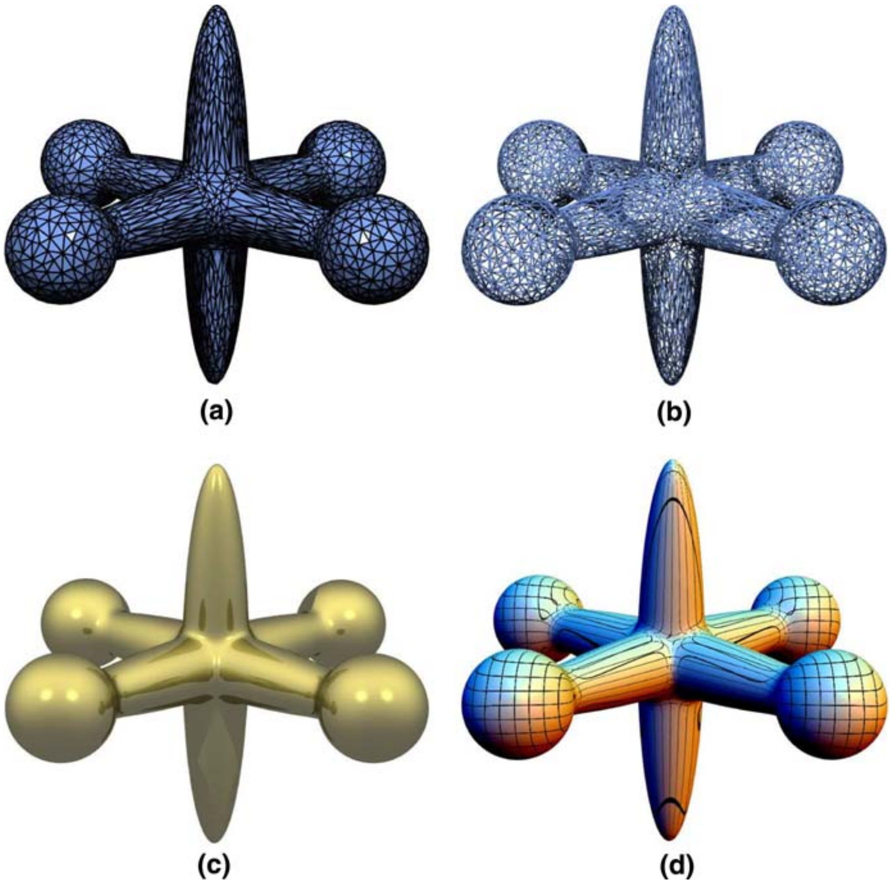


Fig. 9 Approximation of the Jack surface: $f(x, y, z) = \{[x^2/9+4(y^2+z^2)]^{-4}+[y^2/9+4(x^2+z^2)]^{-4}+[z^2/9+4(x^2+y^2)]^{-4}+[(4x/3-4)^2+16(y^2+z^2)/9]^{-4}+[(4x/3+4)^2+16(y^2+z^2)/9]^{-4}+[(4y/3-4)^2+16(x^2+z^2)/9]^{-4}+[(4y/3+4)^2+16(x^2+z^2)/9]^{-4}\}^{-1/4}-1$. **a** Triangular base mesh, **b** Boundary curves network, **c** G^1 triangular spline surface, **d** Reflection lines

and surface curvatures at the vertices of the base mesh. By analyzing the selection of the scalar weight functions and the construction of the boundary curves network, we have shown that the minimum degree required to obtain local and singularity-free scheme is 6. And we have presented a method for constructing boundary curves network which consists of cubic Bézier curves. To eliminate singularities, the base mesh must be locally subdivided and we have proposed an adaptive subdivision strategy for it. The numerical examples demonstrate the visual quality of resulting surfaces.

In the future, we shall try to generalize the above scheme to cover G^2, ε–G^2 and rational forms. We shall also study the problem of shape optimization by adjusting shape parameters such as μ_1, μ_2 and μ_3. Applications of our G^1 triangular spline surface scheme are also worthy of further exploration.

Acknowledgments This work was supported by the Korea Science and Engineering Foundation (KOSEF); Ministry of Education, Science and Technology (MEST); Government of Korea via grant no. 2009-0081219. The work of Wei-hua Tong was partially supported by the National Natural Science Foundation of China under grant No. 10726007. We are strongly indebted to the anonymous referees for reading the manuscript thoroughly and giving valuable comments. Special thanks is due to Malcolm Sabin for reading the draft of this paper and providing many insightful comments and suggestions.

Appendix A: The formula of G^1 continuity condition

Use the notation of Sect. 3 and substitute Eqs. (2) into Eq. (1), we have

$$\begin{aligned}&\left[\sum_{i=0}^{r}\lambda_i B_i^r(t)\right]\cdot\left[\sum_{i=0}^{n-1}\mathbf{p}_i B_i^{n-1}(t)\right]+\left[\sum_{i=0}^{r}\bar{\lambda}_i B_i^r(t)\right]\cdot\left[\sum_{i=0}^{n-1}\mathbf{r}_i B_i^{n-1}(t)\right]\\&=\left[\sum_{i=0}^{s}\mu_i B_i^s(t)\right]\cdot\left[\sum_{i=0}^{m-1}\mathbf{q}_i B_i^{m-1}(t)\right]+\left[\sum_{i=0}^{s}\bar{\mu}_i B_i^s(t)\right]\cdot\left[\sum_{i=0}^{m-1}\mathbf{q}_{i+1} B_i^{m-1}(t)\right].\end{aligned}\tag{15}$$

From the formula of $B_i^m(t)B_j^n(t)=\frac{\binom{m}{i}\binom{n}{j}}{\binom{m+n}{i+j}}B_{i+j}^{m+n}(t)$, we obtain

$$\left[\sum_{i=0}^{m}a_i B_i^m(t)\right]\cdot\left[\sum_{j=0}^{n}b_j B_j^n(t)\right]=\sum_{k=0}^{m+n}\left\{\left[\sum_{i=0}^{k}a_i b_{k-i}\frac{\binom{m}{i}\binom{n}{k-i}}{\binom{m+n}{k}}\right]\cdot B_k^{m+n}(t)\right\}.$$

According to the above equation and the assumption of $r+n=s+m$, we can rewrite Eq. (15) as follows:

$$\begin{aligned}&\sum_{k=0}^{r+n-1}\left[\sum_{i=0}^{k}(\bar{\lambda}_i\mathbf{p}_{k-i}+\lambda_i\mathbf{r}_{k-i})\frac{\binom{r}{i}\binom{n-1}{k-i}}{\binom{r+n-1}{k}}\right]B_k^{r+n-1}(t)\\&=\sum_{k=0}^{r+n-1}\left[\sum_{i=0}^{k}(\bar{\mu}_i\mathbf{q}_{k-i}+\mu_i\mathbf{q}_{k-i+1})\frac{\binom{s}{i}\binom{m-1}{k-i}}{\binom{r+n-1}{k}}\right]B_k^{r+n-1}(t).\end{aligned}\tag{16}$$

Since the Bernstein polynomials $B_k^{r+n-1}(t)$ are linearly independent, we finally arrive at Eq. (3).

References

1. Bloomenthal J (1994) An implicit surface polygonizer. In: Graphics gems IV, Academic Press, San Diego, pp 324–349
2. Boehm W, Prautzsch H, Arner P (1987) On triangular splines. Constr Approx 3:157–167
3. de Boor C, Höllig K (1982) B-splines from parallelepipeds. J d'Anal Math 42:99–115
4. de Boor C, Höllig K, Sabin MA (1987) High accuracy geometric Hermite interpolation. Comp Aided Geom Des 4(4):269–278

5. Cho DY, Lee KY, Kim TW (2006) Interpolating G^1 Bézier surfaces over irregular curve networks for ship hull design. Comp Aided Des 38(6):641–660
6. Cho DY, Lee KY, Kim TW (2007) Analysis and avoidance of singularities for local G^1 surface interpolation of Bézier curve network with 4-valent nodes. Computing 79(2–4):261–279
7. Du WH, Schmitt FJM (1990) On the G^1 continuity of piecewise Bézier surfaces: a review with new results. Comp Aided Des 22(9):556–573
8. Farin G (1985) A modified clough-tocher interpolant. Comp Aided Geom Des 2(1–3):19–27
9. Farin G (2002) Curves and surfaces for CAGD: a practical guide, chap 20, 5th edn. Morgan Kaufmann, San Francisco
10. Fünfzig C, Müller K, Hansford D, Farin G (2008) PNG1 triangles for tangent plane continuous surfaces on the GPU. In: GI'08: Proceedings of graphics interface 2008, pp 219–226
11. Garland M, Heckbert PS (1997) Surface simplification using quadric error metrics. In: Proceedings of ACM SIGGRAPH 1997, computer graphics proceedings. Annual conference series, pp 209–216
12. Garland M, Zhou Y (2005) Quadric-based simplification in any dimension. ACM Trans Graph 24(2):209–239
13. Hagen H, Pottmann H (1989) Curvature continuous triangular interpolants. In: Mathematical methods in computer aided geometric design, Academic Press, San Diego, pp 373–384
14. Hahmann S, Bonneau GP (2000) Triangular G^1 interpolation by 4-splitting domain triangles. Comp Aided Geom Des 17(8):731–757
15. Hoschek J, Lasser D (1993) Fundamentals of computer aided geometric design. AK Peters, Massachusetts
16. Jensen T (1987) Assembling triangular and rectangular patches and multivariate splines. In: Farin G (ed) Geometric modeling: algorithms and new trends, SIAM, Philadelphia, pp 203–220
17. Liu Y, Mann S (2008) Paramteric triangular Bézier surface interpolation with approximate continuity. In: SMP'08: Proceedings of the 2008 ACM symposium on solid and physical modeling, pp 381–387
18. Loop C (1994) A G^1 triangular spline surface of arbitrary topological type. Comp Aided Geom Des 11(3):303–330
19. Mann S, Loop C, Lounsbery M, Meyers D, Painter J, DeRose T, Sloan K (1992) A survey of parametric scattered data fitting using triangular interpolants. In: Hagen H (ed) Curve and surface design, SIAM, Philadelphia, pp 145–172
20. Neamtu M, Pfluger PR (1994) Degenerate polynomial patches of degree 4 and 5 used for geometrically smooth interpolation in $\mathbb{R}^3$. Comp Aided Geom Des 11(4):451–474
21. Nielson G (1987) A transfinite, visually continuous, triangular interpolant. In: Farin G (ed) Geometric modeling: algorithms and new trends, SIAM, Philadelphia, pp 235–246
22. Nocedal J, Wright S (1999) Numerical Optimization. Springer, New York
23. Patrikalakis NM, Maekawa T (2000) Shape interrogation for computer aided design and manufacturing, chap 8. Springer, Berlin
24. Peters J (1991) Smooth interpolation of a mesh of curves. Constr Approx 7(1):221–246
25. Peters J (1995) Biquartic C^1-surface splines over irregular meshes. Comp Aided Des 27(12):895–903
26. Peters J (1995) C^1-surface splines. SIAM J Numer Anal 32(2):645–666
27. Peters J (2002) Geometric continuity. In: Farin G, Hoschek J, Kim MS (eds) Handbook of computer aided geometric design, Elsevier, Amsterdam, pp 193–227
28. Sabin M (1969) Conditions for continuity of surface normal between adjacent parametric surface. Tech. rep. VTO/MS/151, Dynamics and Maths Services Department, British Aircraft Corporation
29. Sabin M (1976) The use of piecewise forms of the numerical representation of shapes. Ph.D. thesis, Hungarian Academy of Sciences, Budapest, Hungary
30. Sabin M (2002) Subdivison surfaces. In: Farin G, Hoschek J, Kim MS (eds) Handbook of computer aided geometric design, Elsevier, Amsterdam, pp 309–326
31. Schaback R (1998) Optimal geometric Hermite interpolation of curves. In: Proceedings of the international conference on mathematical methods for curves and surfaces II, pp 417–428
32. Tong WH, Kim TW (2009) High-order approximation of implicit surfaces by G^1 triangular spline surfaces. Comput Aided Des 41(6):441–445
33. Warren J, Weimer H (2001) Subdivision methods for geometric design: a constructive approach. Morgan Kaufmann, San Francisco
34. Watkins MA (1988) Problems in geometric continuity. Comp Aided Des 20(8):499–502
35. Yvart A, Hahmann S, Bonneau GP (2005) Hierarchical triangular splines. ACM Trans Graph 24(4):1374–1391