Triangular Bubble Spline Surfaces

Mario Kapl [a,*] and Marek Byrtus [b] and Bert Jüttler [a]

[a] *Institute of Applied Geometry, Johannes Kepler University, Linz, Austria*
[b] *Department of Mathematics, University of West Bohemia, Plzeň, Czech Republic*

Abstract

We present a new method for generating a G^n-surface from a triangular network of compatible surface strips. The compatible surface strips are given by a network of polynomial curves with an associated implicitly defined surface, which fulfill certain compatibility conditions. Our construction is based on a new concept, called bubble patches, to represent the single surface patches. The compatible surface strips provide a simple G^n-condition between two neighboring bubble patches, which are used to construct surface patches, connected with G^n-continuity. For $n \leq 2$, we describe the obtained G^n-condition in detail. It can be generalized to any $n \geq 3$. The construction of a single surface patch is based on Gordon-Coons interpolation for triangles.

Our method is a simple local construction scheme, which works uniformly for vertices of arbitrary valency. The resulting surface is a piecewise rational surface, which interpolates the given network of polynomial curves. Several examples of G^0, G^1 and G^2-surfaces are presented, which have been generated by using our method. The obtained surfaces are visualized with reflection lines to demonstrate the order of smoothness.

Key words: G^n-surface, triangular bubble patch, network of compatible surface strips, Gordon-Coons interpolation for triangles

1. Introduction

The present paper describes a new methodology for generating a G^n-surface from a triangular mesh given by a network of curves with an associated implicitly defined surface, called compatible surface strips. Our method is a multi-patch scheme, which is one of the main approaches for constructing a smooth surface from a triangular or quadrilateral mesh (cf. [4, 12]). The fundamental idea of this approach is the construction of single surface patches, pieced together with G^n-continuity, which define the desired G^n-surface. A survey of this concept is presented in [17]. The amount of existing literature on this topic is large; some references are [6, 7, 13, 16, 18, 19, 20].

Based on the multi-patch approach, several methods for constructing a G^n-surface, which interpolates a given *network of curves*, have been developed (cf. [9, 14, 15, 24]). In general, a mesh of curves has to fulfill certain compatibility conditions, especially at the vertices, to be feasible for G^n-interpolation. For instance, [10, 11] describe geometric constraints on a network of curves which have to be satisfied for the case of G^1 and G^2-smooth surfaces.

A method for generating a smooth triangular (rectangular) surface from given three triangular (four rectangular) surface patches is presented in [21, 22]. The construction is based on a generalization of the Gordon-Coons interpolation.

In [8], Hahn describes an algorithm for filling k-sided polygonal holes with quadrilateral surface patches, meeting at a common vertex. The single surface patches are generated with the help of Cordon-Coons interpolation in such a way that they are joined with G^n-continuity. The main difference to our approach consists in the fact, that Hahn's construction uses an explicitly constructed diffeomorphism between neighboring patches in order to satisfy the G^n-conditions. Consequently, the construction of a single patch depends on the neighboring patches. The choice of the Hermite boundary data for one edge defines already the data for all other edges. Moreover, the construction is not perfectly symmetric, since the cross boundary derivatives are always specified on one side of the edges.

The novelty of our method consists in the use of a *network of compatible surface strips* instead of a network of curves. Starting from a triangular network of compatible surface strips, we generate a triangular spline surface with

* Corresponding author
Email addresses: mario.kapl@jku.at (Mario Kapl), byrtus@kma.zcu.cz (Marek Byrtus), bert.juettler@jku.at (Bert Jüttler).

G^n-continuity, interpolating the given curves of the compatible surface strips. Moreover, we do not need to specify the diffeomorphism between neighboring patches, since the G^n-conditions are specified as contact between implicit and parametric surfaces.

Our method is based on a new type of surface patches, called bubble patches, for representing the individual surfaces. The use of these surface patches provides a simple and natural way to define free-form surfaces over triangular meshes and possesses several advantages. The construction of a single patch is simple and independent of the neighboring patches. In addition, it works uniformly for triangular meshes with vertices of arbitrary valency. Consequently, modifications of the connectivity of the mesh only affect those patches that are actually modified. The resulting surfaces are piecewise rational with G^n-continuity.

The remainder of the paper is organized as follows. In Sections 2 and 3 we introduce some basic definitions. In particular, we introduce the idea of a triangular network of compatible surface strips, which is a network of polynomial curves with an associated implicitly defined surface, fulfilling certain compatibility conditions. A possible construction of a suitable network is presented in Appendix B.

Section 4 describes a new methodology, called bubble patches, for representing triangular surface patches. In Section 5 we use the compatible surface strips to describe a simple G^n-condition between two bubble patches. This provides a simple method to generate Hermite boundary data of the patches, which guarantees G^n-continuity between two neighboring patches.

Section 6 describes the construction of C^n-bubble patches by using Gordon-Coons interpolation for triangles. This interpolation scheme is a well known tool for generating a smooth function, interpolating the given Hermite boundary data and is summarized in Appendix A. Section 7 presents several examples of generated G^0, G^1 and G^2-surfaces and verifies their smoothness with the help of reflection lines. Finally, we conclude the paper.

2. Surface strips

We explain the concept of a surface strip of order n along a curve, which is a standard concept in classical differential geometry.

Definition 1 *Let* $\mathbf{p} : [0,1] \to \mathbb{R}^3$ *a smooth parametric curve. A* ***surface strip*** *of order* n *along the curve* $\mathbf{p}$ *is as an equivalence class of all surfaces through the curve* $\mathbf{p}$ *having a contact of order* n *along this curve.*

More precisely, we describe a surface strip of order n along the curve $\mathbf{p}$, depending on a parametric or implicit representation of the surfaces, as follows.

- *Parametric representation of the surfaces*: A surface strip of order $n \leq 2$ is an equivalence class of all parametric surfaces containing the curve $\mathbf{p}$ ($n = 0$), having the same tangent planes ($n = 1$) and the same normal curvatures ($n = 2$) along the curve $\mathbf{p}$.

  A surface strip of order $n = 0$ along the curve $\mathbf{p}$ is simply described by the curve $\mathbf{p}$. A surface strip of order $n = 1$ is the curve with associated tangent planes. These planes can be described by specifying the first derivative vector of the parametric surfaces across the given curve (the first cross-boundary derivative).

  A surface strip of order $n = 2$ is the curve with associated tangent planes and curvature information. The curvature information can be described by specifying the second derivative vector of the parametric surfaces across the given curve.
- *Implicit representation of the surfaces*: A surface strip of order n is an equivalence of all implicitly defined surfaces containing the curve $\mathbf{p}$ ($n = 0$), having – possibly after a rescaling of the surfaces – the same gradients ($n = 1$) and the same Hessian matrices ($n = 2$) along the curve $\mathbf{p}$.

  A surface strip of order $n = 0$ along the curve $\mathbf{p}$ is simply the curve $\mathbf{p}$. A surface strip of order $n = 1$ is the curve with associated gradient information along it, specifying the tangent plane. A surface strip of order $n = 2$ is the curve with associated gradient information and Hessian matrices along the curve.

For both representations of the surfaces, a surface strip of order n along the curve $\mathbf{p}$ can be seen as the curve $\mathbf{p}$ with the truncated Taylor expansions of order n of the parametric/implicit surfaces along the curve. Clearly, the truncated Taylor expansions have to satisfy certain compatibility conditions for the curve.

In the case of parametric surfaces, the derivatives of the surfaces along the given curve have to agree with the corresponding derivatives of the curve, and the mixed derivatives of the surfaces have to agree with the corresponding derivatives of the previously specified cross-boundary derivatives. Thus, for each order of smoothness, one additional vector field of cross-boundary derivatives can be specified.

Similar compatibility conditions are presented in the case of strips defined by implicitly defined surfaces. For instance, in the case $n = 1$, the specified gradient vectors of the implicitly defined surface have to be orthogonal to the tangents of the curve. Similar conditions can be derived for higher values of n, simply by differentiating the composition of the parametric representations of the curve and the implicitly defined surface through it with respect to the curve parameter.

3. Triangular network of compatible surface strips

Let $n \in \mathbb{Z}_0^+$ and let $\mathcal{M}$ be a *triangular mesh*, given by vertices $\mathbf{v} \in V$ and edges $\mathbf{e} = (\mathbf{v}, \mathbf{w}) \in E$ with $\mathbf{v}, \mathbf{w} \in V$, where V is the vertex set and E is the edge set of the mesh. In addition, we consider a smooth *implicitly defined surface* $F = \{\mathbf{z} \in \mathbb{R}^3 : f(\mathbf{z}) = 0\}$ satisfying $f(\mathbf{v}) = 0$ for $\mathbf{v} \in V$.

For each edge $\mathbf{e} = (\mathbf{v}, \mathbf{w})$ of the mesh, we consider a boundary curve $\mathbf{p_e} : [0,1] \to \mathbb{R}^3$ with $\mathbf{p_e}(0) = \mathbf{v}$ and $\mathbf{p_e}(1) = \mathbf{w}$ of the form

$$\mathbf{p_e}(t) = \tilde{\mathbf{L}}_\mathbf{e}(t) + \tilde{h}_\mathbf{e}(t)\tilde{\mathbf{N}}_\mathbf{e}(t), \quad (1)$$

where $\tilde{\mathbf{L}}_\mathbf{e}$ and $\tilde{\mathbf{N}}_\mathbf{e}$ are the linear interpolants

$$\tilde{\mathbf{L}}_\mathbf{e}(t) = (1-t)\mathbf{v} + t\mathbf{w}$$

of the vertices and

$$\tilde{\mathbf{N}}_\mathbf{e}(t) = (1-t)(\nabla f)(\mathbf{v}) + t(\nabla f)(\mathbf{w}),$$

of the associated normals, respectively. Moreover, the function $\tilde{h}_\mathbf{e} : [0,1] \to [0,1]$, which is assumed to be a polynomial function satisfying $\tilde{h}_\mathbf{e}(0) = \tilde{h}_\mathbf{e}(1) = 0$, specifies the deviation of the mesh from the corresponding edge of the mesh.

Clearly, the curves of this form are fairly special, since they have to be contained in the bi-linear ruled surfaces that are generated by the linear interpolants of points and normals. We consider this special class of curves only, in order to be compatible with the triangular bubble patches which will be introduced in the next section.

We will construct the boundary curves and the implicitly defined surface F such that the surface contains the boundary curve $\mathbf{p_e}$, i.e.

$$f(\mathbf{p_e}(t)) = 0$$

for $t \in [0,1]$. In order to construct a G^n-smooth interpolating bubble spline surface we need to evaluate the first n derivatives of F along the boundary. Consequently, we do not need to have a closed-form representation of f at all points in $\mathbb{R}^3$. Instead, we only specify how to evaluate the derivatives of F along the curve. This can be done in such a way that the surface F is compatible with the network of curves.

The curve $\mathbf{p_e}$ with the implicitly defined surface F – which is represented by its truncated Taylor expansion of order n at all points of the curve – is referred to as a *compatible surface strip* of order n for the edge $\mathbf{e}$, and the network of boundary curves $\mathbf{p_e}$ with the implicitly defined surface is called a *network of compatible surface strips*. More precisely, the network of compatible surface strips for $n \leq 2$ can be interpreted as follows.

- Case $n = 0$: It is simply the network of boundary curves.
- Case $n = 1$: It is the network of boundary curves with the associated gradients of the function f along the curves.
- Case $n = 2$: It is the network of boundary curves with the associated gradients and Hessian matrices of the function f along the curves.

A construction leading to a network of compatible surface strips is described in Appendix B.

4. Triangular bubble patches

We introduce a new concept, called bubble patches, for representing surface patches on triangles with given vertices and normals. This provides a simple and natural way to obtain free-form surfaces from triangular meshes.

Let $\mathbf{T}$ be the standard domain triangle defined by

$$\mathbf{T} = \{(u,v) | (u,v) \in [0,1]^2; u+v \leq 1\}$$

with the vertices $\mathbf{v}_1 = (0,1)$, $\mathbf{v}_2 = (1,0)$ and $\mathbf{v}_3 = (0,0)$.

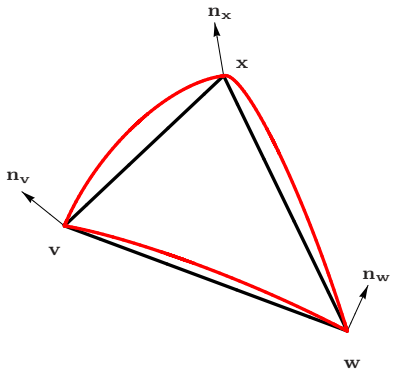

Fig. 1. The boundary curves of a bubble patch on a triangle with given vertices and normals.

Definition 2 *Let $\mathcal{T}$ be a triangle of $\mathcal{M}$ with the vertices $\mathbf{v}, \mathbf{w}, \mathbf{x} \in V$, connected by the edges $(\mathbf{v},\mathbf{w}), (\mathbf{v},\mathbf{x}), (\mathbf{x},\mathbf{w}) \in E$, and the corresponding normals $\mathbf{n_v}$, $\mathbf{n_w}$, $\mathbf{n_x}$; see Fig. 1. We define a surface $\mathbf{B} : \mathbf{T} \to \mathbb{R}^3$ on the triangle $\mathcal{T}$ as follows*

$$\mathbf{B}(u,v) = \mathbf{L}(u,v) + h(u,v)\mathbf{N}(u,v), \quad (2)$$

where $\mathbf{L}$ and $\mathbf{N}$ are linear interpolants, given by

$$\mathbf{L}(u,v) = (1-u-v)\mathbf{v} + u\mathbf{w} + v\mathbf{x}$$

and

$$\mathbf{N}(u,v) = (1-u-v)\mathbf{n_v} + u\mathbf{n_w} + v\mathbf{n_x},$$

*and h is a scalar-valued function. Moreover we require that $\mathbf{B}(0,0) = \mathbf{v}$, $\mathbf{B}(1,0) = \mathbf{w}$, and $\mathbf{B}(0,1) = \mathbf{x}$ which implies that $h(0,0) = h(1,0) = h(0,1) = 0$. The function $\mathbf{B}$ is referred to as a **bubble patch** (on the triangle $\mathcal{T}$) and the function h is called the **bubble function** (on the triangle $\mathcal{T}$).*

In our case, where the construction starts with a network of compatible surface strips, the normals $\mathbf{n_v}$, $\mathbf{n_w}$ and $\mathbf{n_x}$ are given by the gradients of the implicitly defined surface, evaluated at the vertices,

$$\mathbf{n_v} = (\nabla f)(\mathbf{v}),\ \mathbf{n_w} = (\nabla f)(\mathbf{w}) \text{ and } \mathbf{n_x} = (\nabla f)(\mathbf{x}),$$

respectively. An example of a bubble patch on a triangle is shown in Fig. 1.

A bubble patch $\mathbf{B}$ is generated by selecting points on a special 2-parametric family of lines, see Fig. 2. These lines are obtained by combining the linear parametrization of the triangle with the linear interpolation of the normal vectors at the vertices. Consequently, the parametrization of the surface is uniquely determined by this family of lines.

The regularity of a bubble patch can also be guaranteed with the help of this parametrization. If the three boundary normals do not deviate too much from the normal of the triangle, and if the values of the bubble function are not too large, then the bubble patch will be regular.

Bubble patches are in close relation to the underlying triangular mesh. There is a one-to-one correspondence between the points of the mesh and the points on the surface. In addition, it is easy to bound the distance. The distance

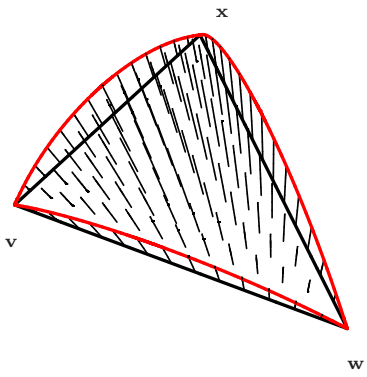

Fig. 2. A 2-parametric family of lines defines the bubble patch.

between a bubble patch $\mathbf{B}$ and the corresponding triangle $\mathcal{T}$ of the mesh $\mathcal{M}$ can be bounded by

$$\max_{(u,v)\in\mathbf{T}} |h(u,v)|\ ||\mathbf{N}(u,v)||.$$

5. Strip-compatible bubble patches

We use a network of compatible surface strips to describe a simple construction for a G^n-smooth spline surface composed of triangular bubble patches. The idea is to construct bubble patches which have a contact of order n with the implicitly defined surface F along the boundary curves. This provides a simple method to generate Hermite boundary data

$$\frac{\partial^i}{\partial v^i} h(u,v)\ \big|_{v=0}\,,\ \frac{\partial^i}{\partial u^i} h(u,v)\ \big|_{u=0} \tag{3}$$

and

$$\frac{\partial^i}{\partial v^i} h(u,v)\ \big|_{v=1-u}\,,\ \frac{\partial^i}{\partial u^i} h(u,v)\ \big|_{u=1-v} \tag{4}$$

for $i \in \{0,\dots,n\}$. By interpolating these data we are then able to guarantee G^n-continuity between neighboring bubble patches.

Our construction of a G^n-smooth spline surface consists of the following two steps. At first we generate for each triangle Hermite boundary data (3) and (4) of the bubble function h. This step is described in the remainder of this section. In the second step we construct the single bubble patches by applying Gordon-Coons interpolation to the generated Hermite boundary data; see Section 6.

Now we describe the first step of our construction. We consider a bubble patch $\mathbf{B}$ and a compatible surface strip given by the curve $\mathbf{p_e}$ and the implicitly defined surface F, such that $\mathbf{p_e}$ is a boundary curve of $\mathbf{B}$. For the sake of simplicity, we consider the case

$$\mathbf{B}(t,0) = \mathbf{p_e}(t). \tag{5}$$

A contact of order n between the patch $\mathbf{B}$ and the surface F along the curve $\mathbf{p_e}$ is described by the contact condition

$$\frac{\partial^{i+j}}{\partial u^i \partial v^j} f(\mathbf{B}(u,v)\ \big|_{v=0} = 0 \tag{6}$$

for $i,j \in \{0,\dots,n\}$ with $i+j \le n$. Since equation (5) is satisfied, we have

$$f(\mathbf{B}(u,0)) = 0 \tag{7}$$

and condition (6) is trivially satisfied for $j=0$. Moreover, we also have

$$h(t,0) = \tilde{h}_{\mathbf{e}}(t).$$

Depending on the order n of contact, the condition

$$\frac{\partial^j}{\partial v^j} f(\mathbf{B}(u,v)\ \big|_{v=0} = 0 \tag{8}$$

for $j \in \{1,\dots,n\}$ provides the Hermite boundary data

$$\frac{\partial^i}{\partial v^i} h(u,v)\ \big|_{v=0}$$

of the bubble function h for $j \in \{1,\dots,n\}$ as follows. For better readability, we denote the first and second partial derivative of a bivariate function $\mathbf{r} : [0,1]^2 \to \mathbb{R}^d$ ($d \in \{1,2,3\}$) with respect to the second argument by $\mathbf{r}_{0,1}$ and $\mathbf{r}_{0,2}$, respectively.

Contact of order 1: The condition

$$\frac{\partial}{\partial v} f(\mathbf{B}(u,v)\ \big|_{v=0} = 0$$

is equivalent to

$$(\nabla f)(\mathbf{B}(u,0))^T \cdot \mathbf{B}_{0,1}(u,0) = 0.$$

By combining this observation with

$$\begin{aligned}\mathbf{B}_{0,1}(u,0) = {}& \mathbf{L}_{0,1}(u,0) + h_{0,1}(u,0)\mathbf{N}(u,0)\\ & + h(u,0)\mathbf{N}_{0,1}(u,0)\end{aligned}$$

we obtain the first partial derivative

$$\begin{aligned}h_{0,1}(u,0) = {}& -\frac{(\nabla f)(\mathbf{B}(u,0))^T \cdot \mathbf{L}_{0,1}(u,0)}{(\nabla f)(\mathbf{B}(u,0))^T \cdot \mathbf{N}(u,0)}\\ & - \frac{h(u,0)(\nabla f)(\mathbf{B}(u,0))^T \cdot \mathbf{N}_{0,1}(u,0)}{(\nabla f)(\mathbf{B}(u,0))^T \cdot \mathbf{N}(u,0)}\end{aligned}$$

of the bubble function.

Contact of order 2: The second partial derivative $h_{0,2}(u,0)$ of the bubble function can be computed similarly. The condition

$$\frac{\partial^2}{\partial v^2} f(\mathbf{B}(u,v)\ \big|_{v=0} = 0$$

can equivalently be reformulated as

$$\begin{aligned}&(\nabla f)(\mathbf{B}(u,0))^T \cdot \mathbf{B}_{0,2}(u,0)\\ &+\mathbf{B}_{0,1}(u,0)^T \cdot \mathrm{Hess}(f)(\mathbf{B}(u,0)) \cdot \mathbf{B}_{0,1}(u,0) = 0.\end{aligned}$$

Since $\mathbf{B}_{0,2}(u,0)$ is given by

$$\mathbf{B}_{0,2}(u,0) = h_{0,2}(u,0)\mathbf{N}(u,0) + 2h_{0,1}(u,0)\mathbf{N}_{0,1}(u,0),$$

we obtain

$$\begin{aligned}h_{0,2}(u,0) = {}& -\frac{\mathbf{B}_{0,1}(u,0)^T \cdot \mathrm{Hess}(f)(\mathbf{B}(u,0)) \cdot \mathbf{B}_{0,1}(u,0)}{(\nabla f)(\mathbf{B}(u,0))^T \cdot \mathbf{N}(u,0)}\\ & - \frac{2h_{0,1}(u,0)(\nabla f)(\mathbf{B}(u,0))^T \cdot \mathbf{N}_{0,1}(u,0)}{(\nabla f)(\mathbf{B}(u,0))^T \cdot \mathbf{N}(u,0)}.\end{aligned}$$

Contact of order $s \geq 3$: We compute the partial derivative

$$\frac{\partial^s}{\partial v^s} h(u,v) \mid_{v=0}$$

by using condition (8) and the previously computed partial derivatives

$$\frac{\partial^j}{\partial v^j} h(u,v) \mid_{v=0}$$

of the bubble function for $j \in \{0, \ldots, s-1\}$.

The fulfillment of the contact condition (6) follows directly from the conditions (7) and (8). Consequently, the bubble patch $\mathbf{B}$ possesses a contact of order n with the implicitly defined surface F along the curve $\mathbf{p_e}$.

By describing analogous G^n-conditions for the boundaries $\mathbf{B}(0,t)$ and $\mathbf{B}(t,1-t)$, we obtain the remaining Hermite boundary data of (3) and (4).

Since the bubble patch $\mathbf{B}$ has a contact of order n with the implicitly defined surface F along the boundary curves which are contained in the surface F, the Hermite boundary data (3) and (4) fulfill the condition

$$\frac{\partial^{i+j}}{\partial u^i \partial v^j} h(u,v) \mid_{(u,v)=\mathbf{v}_k} = \frac{\partial^{i+j}}{\partial v^j \partial u^i} h(u,v) \mid_{(u,v)=\mathbf{v}_k}, \quad (9)$$

for $i,j \in \{1,\ldots,n\}$ at the vertices $\mathbf{v}_k$ with $k \in \{1,2,3\}$. Condition (9) is called the *twist compatibility condition* and is needed for applying Gordon-Coons interpolation to construct a C^n-bubble patch $\mathbf{B}$; see Section 6 and Appendix A.

Condition (6) provides a simple method to construct Hermite boundary data (3) and (4) for an arbitrary value of n, which guarantees G^n-continuity between two neighboring bubble patches. Only a few linear equations need to be solved to get the Hermite boundary data of one patch. Another advantage of this approach is that the generation of the Hermite boundary data is perfectly local, i.e., it is entirely independent of the computation performed for the other patches. In addition, this method works uniformly for vertices of arbitrary valency.

6. Construction of bubble patches

In the previous section we explained how to generate Hermite boundary data (3) and (4) of the bubble function h, which ensures G^n-continuity between two neighboring patches. By using Gordon-Coons interpolation for triangles, we can construct from the generated Hermite boundary data C^n-bubble patches, which are connected with G^n-continuity.

Gordon-Coons interpolation for triangles is a transfinite interpolation scheme which generates from the given Hermite boundary data a smooth function, interpolating this boundary data (cf. [1, 2, 3, 5]). The main concept of this interpolation scheme is similar to the Gordon-Coons method for rectangles (cf. [4, 12]) and can be summarized as follows.

Let $\bar{\mathbf{e}}_1 = (\mathbf{v}_2, \mathbf{v}_3)$, $\bar{\mathbf{e}}_2 = (\mathbf{v}_1, \mathbf{v}_3)$ and $\bar{\mathbf{e}}_3 = (\mathbf{v}_1, \mathbf{v}_2)$ be the edges of the standard triangle $\mathbf{T}$. For $i,j,k \in \{1,2,3\}$ with $i \neq j \neq k \neq i$, we denote by $\mathbf{P}_i$ the operator, which performs Hermite interpolation parallel to the edge $\bar{\mathbf{e}}_i$, using Hermite boundary data, given for the edges $\bar{\mathbf{e}}_j$ and $\bar{\mathbf{e}}_k$; see Fig. 3. By combining the three operators we obtain an interpolation operator, which generates a smooth function, interpolating the given boundary data.

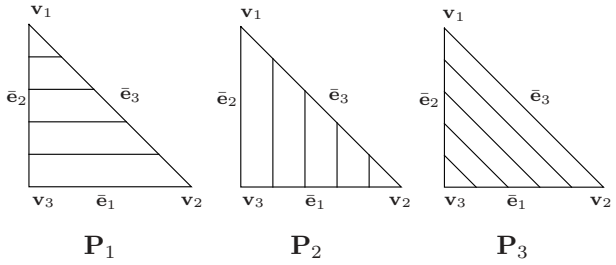

Fig. 3. The three operators $\mathbf{P}_1$, $\mathbf{P}_2$ and $\mathbf{P}_3$ which perform Hermite interpolation parallel to the edges $\bar{\mathbf{e}}_1$, $\bar{\mathbf{e}}_2$ and $\bar{\mathbf{e}}_3$, respectively.

By applying this procedure to the Hermite boundary data (3) and (4) of order n we obtain a C^n-bubble function h, and hence a C^n-bubble patch $\mathbf{B}$. Moreover, the bubble patches are all rational surfaces, which are pieced together with G^n-continuity.

In order to make this paper self-contained, Appendix A summarizes the general concept of the Gordon-Coons method of degree n in detail and present explicit formulas for $n = 0$ and $n = 1$.

The generated bubble surfaces are rational, but their degree is generally so high that an evaluation in closed form does not appear to be of much use. Thus, there are two possibilities to use our construction.

First, one may use the bubble patches as a procedural definition for a surface, where each point of the surface can be evaluated by following the evaluation path determined by the construction. Each step in the construction requires solely standard arithmetic operations, hence the evaluation can be performed easily and derivatives can be obtained, e.g., by using tools such as automatic differentiation.

Second, one may generate low degree approximations (e.g. using spline functions) for the Hermite boundary data. If this approach is taken, then we obtain bubble patches which are spline surfaces of low degree. We have used this approach in our current implementation. The functions specifying the Hermite boundary data along the curves are represented by standard cubic splines. This leads to spline surface patches of degree $(5,5)$, $(7,7)$ and $(9,9)$ for $n = 0$, $n = 1$ and $n = 2$, respectively.

Clearly, the surfaces which are obtained in this way are only approximately G^n-smooth, since an approximation step is performed. However, there exists a precise mathematical model of an exactly G^n-smooth surface. If the level of smoothness turns out to be too low, then one can easily improve the accuracy of the cubic spline approximation used to represent the Hermite boundary data. Only one-dimensional spline fitting procedures are needed for this task.

7. Examples

We present several examples of generated G^0, G^1 and G^2-surfaces, see Fig. 4. These surfaces are interpolating the given curves of the triangular networks of compatible surface strips. A network of compatible surface strips is obtained by using the construction method in Appendix B. Fig. 4 shows only the vertices and edges of the different triangular meshes that were used to generate the networks of compatible surfaces strips. We use reflection lines, which is a well known tool for verifying the resulting geometric continuity (cf. [23]), to demonstrate the smoothness of the surfaces.

We can observe, that all resulting surfaces have the required geometric continuity. In the case of G^1 and G^2, all reflection lines of the surfaces are at least G^0 and G^1, respectively. In the examples of the tube and of the rotated horse shoe, it is clear to see that the resulting G^0 and G^1-surfaces are only G^0 and G^1, respectively, but not smoother. In addition, we can observe that the order of geometric smoothness has nothing to do with the aesthetic appeal of the resulting surfaces. A possible attempt to fair the surfaces could be a different choice of the implicitly defined surfaces at the vertices of the mesh in the first step of the construction of a suitable network of compatible surface strips; see Appendix B.

8. Conclusion

In this paper we have presented a simple method for constructing a G^n-surface from a triangular network of compatible surface strips, which is a network of curves with an associated implicitly defined surface.

Summing up, our construction scheme works as follows. For each triangle we generate a single surface patch, represented by a new methodology, called bubble patch. Thereby a single surface patch is constructed with the help of triangular Coons interpolation in such a way that the patches are connected with G^n-continuity. This is achieved by generating Hermite boundary data of the bubble patch for the Gordon-Coons method, which ensures the desired G^n-continuity between the patches. For this we use the compatible surface strips to describe a simple G^n-condition between the patches, which provides the suitable Hermite boundary data. Our algorithm is explained for $n \leq 2$ in detail and several examples are presented.

The advantages of our method are as follows. Our construction is local and simple and works uniformly for meshes of arbitrary valency. The resulting surfaces are piecewise rational surface patches, which are pieced together with G^n-continuity. By using low degree approximations (e.g. splines) for the Hermite boundary data we obtain (spline) surface patches of low degree. The method can be extended to quadrilateral and mixed triangular and quadrilateral meshes with vertices of arbitrary valency. An example from a quadrilateral mesh is shown in Fig 5.

As a possible topic for future work, we are currently trying to derive a geometric subdivision scheme from our construction, by sampling points and Hermite data from the constructed surface. The use of a globally defined implicit representation could provide a new and promising approach to verify the smoothness of the obtained limit surfaces, which does not require local parametrizations and can deal uniformly with vertices of any valency.

Acknowledgements The second author has been supported by the Austrian Science Fund, project S9202.

Appendix A. Gordon-Coons interpolation for triangles

We give a short overview of a transfinite interpolation scheme for triangles, which is often known as the Gordon-Coons method for triangles. For more details of Gordon-Coons interpolation for triangles we refer to [1, 2, 3, 5].

A.1. *General concept*

For $n \in \mathbb{Z}_0^+$, $k \in \{0,1\}$ and $j \in \{0,\dots n\}$, we denote by $H_{k,j}^{2n+1}$ the classical Hermite polynomials of degree $2n+1$, i.e.

$$H_{k,j}^{2n+1} : [0,1] \to \mathbb{R}$$

satisfying

$$\frac{\partial^i}{\partial t^i} H_{k,j}^{2n+1}(t) \,\big|_{t=l} = \delta_{i,j}\delta_{k,l}$$

for $j \in \{0,\dots,n\}$ and $l \in \{0,1\}$.

A general construction scheme of the Gordon-Coons method of degree n for generating a C^n-function $h : \mathbf{T} \to \mathbb{R}$ is as follows:

(i) Let

$$h_{i,j}(u,v) = \frac{\partial^{i+j}}{\partial r^i \partial s^j} h(r,s) \,\big|_{(r,s)=(u,v)}$$

for $i,j \in \{0,\dots,n\}$. Consider the given Hermite boundary data

$$h_{i,0}(0,v), h_{0,i}(u,0), h_{i,0}(1-v,v) \text{ and } h_{0,i}(u,1-u)$$

for $i \in \{0,\dots,n\}$, satisfying the twist compatibility condition (9).

(ii) For a function $g : \mathbf{T} \to \mathbb{R}$, define the operators $\mathbf{P}_1$, $\mathbf{P}_2$ and $\mathbf{P}_3$ as

$$\begin{aligned}\mathbf{P}_1 &= \sum_{i=0}^{n} H_{0,i}^{2n+1}\Big(\frac{u}{1-v}\Big)(1-v)^i g_{i,0}(0,v) \\ &\quad + \sum_{i=0}^{n} H_{1,i}^{2n+1}\Big(\frac{u}{1-v}\Big)(1-v)^i g_{i,0}(1-v,v), \\ \mathbf{P}_2 &= \sum_{i=0}^{n} H_{0,i}^{2n+1}\Big(\frac{v}{1-u}\Big)(1-u)^i g_{0,i}(u,0) \\ &\quad + \sum_{i=0}^{n} H_{1,i}^{2n+1}\Big(\frac{v}{1-u}\Big)(1-u)^i g_{0,i}(u,1-u),\end{aligned}$$

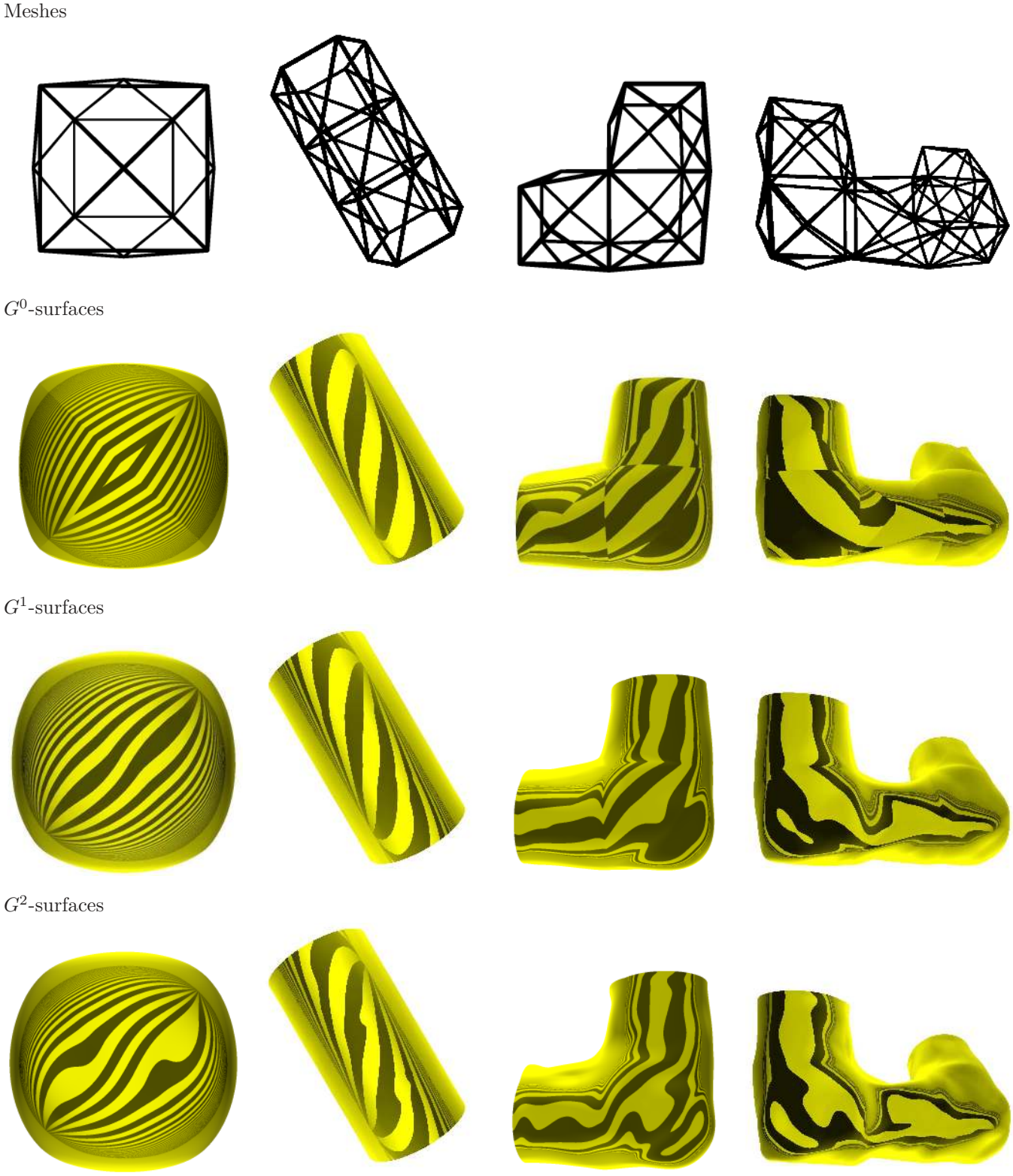

Fig. 4. Examples of different meshes and the resulting G^0, G^1 and G^2-surfaces with reflection lines.

Rotated horse shoe G^1-surface G^2-surface

Fig. 5. Example of a quadrilateral mesh and the resulting G^1 and G^2-surfaces with reflection lines.

and

$$\mathbf{P}_3 = \sum_{i=0}^{n} H_{0,i}^{2n+1}(\frac{u}{u+v})(u+v)^i \mathbf{W}_i(0,u+v) + \sum_{i=0}^{n} H_{1,i}^{2n+1}(\frac{u}{u+v})(u+v)^i \mathbf{W}_i(u+v,0),$$

where the indices for g indicate differentiation and

$$\mathbf{W}_i(u,v) = \sum_{j=0}^{i} \binom{i}{j} (-1)^j g_{i-j,j}(u,v).$$

(iii) Then the function h is given by

$$h(u,v) = (\mathbf{P}_i \oplus \mathbf{P}_j) h(u,v) \tag{A.1}$$

for any choice of $i,j \in \{1,2,3\}$ with $i \neq j$, where the Boolean sum operator $\mathbf{P}_i \oplus \mathbf{P}_j$ is given by

$$\mathbf{P}_i \oplus \mathbf{P}_j = \mathbf{P}_i + \mathbf{P}_j - \mathbf{P}_i \mathbf{P}_j,$$

see [2, Theorem 2.1].

The fulfillment of the twist compatibility condition (9) for the vertex $\mathbf{v}_k$ for $k \in \{1,2,3\}$ with $i \neq j \neq k \neq i$ is sufficient for applying the interpolant (A.1); see [2, Theorem 2.1]. An example of an interpolant (A.1) is

$$h(u,v) = (\mathbf{P}_1 \oplus \mathbf{P}_2) h(u,v), \tag{A.2}$$

for which the explicit formulas for $n = 0$ and $n = 1$ are presented in Subsection A.2 and A.3, respectively (cf. [2]). Clearly, the interpolant (A.1) is not symmetric. According to [2], a more symmetric interpolant can be obtained by averaging interpolants (A.1).

[2, Theorem 2.4] describes similar interpolants, which do not need the twist compatibility condition (9) to be satisfied. Instead, they need some additional assumptions concerning the Hermite boundary data (3) and (4).

A.2. *Gordon-Coons interpolation of degree* 0

The Boolean sum operator $(\mathbf{P}_1 \oplus \mathbf{P}_2) h(u,v)$ is given by

$$(\mathbf{P}_1 \oplus \mathbf{P}_2) h(u,v) = H_{0,0}^1(\frac{v}{1-u}) h(u,0) + H_{1,0}^1(\frac{v}{1-u}) h(u,1-u) + H_{0,0}^1(\frac{u}{1-v})[h(0,v) - \mathbf{Q}_0^{(0)}(0,v)],$$

where

$$\mathbf{Q}_0^{(0)}(0,v) = H_{0,0}^1(v) h(0,0) + H_{1,0}^1(v) h(0,1).$$

A.3. *Gordon-Coons interpolation of degree* 1

The explicit formula of the Boolean sum operator $(\mathbf{P}_1 \oplus \mathbf{P}_2) h(u,v)$ is

$$(\mathbf{P}_1 \oplus \mathbf{P}_2) h(u,v) = \sum_{i=0}^{1} H_{0,i}^3(\frac{v}{1-u})(1-u)^i h_{0,i}(u,0) + \sum_{i=0}^{1} H_{1,i}^3(\frac{v}{1-u})(1-u)^i h_{0,i}(u,1-u) + \sum_{i=0}^{1} H_{0,i}^3(\frac{u}{1-v})(1-v)^i [h_{i,0}(0,v) - \mathbf{Q}_1^{(i)}(0,v)],$$

where

$$\mathbf{Q}_1^{(0)}(0,v) = \sum_{i=0}^{1} H_{i,0}^3(v) h_{0,i}(0,0) + \sum_{i=0}^{1} H_{i,1}^3(v) h_{0,i}(0,1),$$

and

$$\begin{aligned}\mathbf{Q}_1^{(1)}(0,v) = v[&\sum_{i=0}^{1}\frac{\partial^i}{\partial t^i}H_{0,i}^3(t)\mid_{t=v} h_{0,i}(0,0)\\ &+\sum_{i=0}^{1}\frac{\partial^i}{\partial t^i}H_{1,i}^3(t)\mid_{t=v} h_{0,i}(0,1)]\\ &+H_{0,0}^3(v)h_{1,0}(0,0)\\ &+H_{0,1}^3(v)[-h_{0,1}(0,0)+h_{1,1}(0,0)]\\ &+H_{1,0}^3(v)[h_{1,0}(0,1)-h_{0,1}(0,1)]\\ &+H_{0,1}^3(v)[-h_{0,1}(0,1)+h_{1,1}(0,1)].\end{aligned}$$

Appendix B. Construction of a triangular network of compatible surface strips

We explain the construction of a triangular network of compatible surface strips from a triangular mesh of given vertices and associated normals. For this, we consider the triangular mesh $\mathcal{M}$ with vertices $\mathbf{v} \in V$, edges $\mathbf{e} = (\mathbf{v}, \mathbf{w}) \in E$ with $\mathbf{v}, \mathbf{w} \in V$. In addition, we assume that we have for each vertex $\mathbf{v} \in V$ an associated normal $\mathbf{n_v}$. The construction of a network of compatible surface strip of order n consists of three steps, described in the following subsections.

B.1. *Construction of implicitly defined surfaces at the vertices*

At first, we generate for each vertex $v \in V$ an associated implicitly defined surface $F_\mathbf{v} = \{\mathbf{z} \in \mathbb{R}^3 : f_\mathbf{v}(\mathbf{z}) = 0\}$ of degree m, given by its truncated Taylor expansion

$$f_\mathbf{v}(\mathbf{z}) = \mathbf{g}(\mathbf{v})^T \cdot (\mathbf{z}-\mathbf{v}) + \frac{1}{2}(\mathbf{z}-\mathbf{v})^T \cdot \mathbf{H}(\mathbf{v}) \cdot (\mathbf{z}-\mathbf{v}) + \ldots,$$

fulfilling

$$\mathbf{g}(\mathbf{v}) = \mathbf{n_v},$$

where $\mathbf{g}(\mathbf{v})$ is the gradient and $\mathbf{H}(\mathbf{v})$ is the Hessian matrix of the function $f_\mathbf{v}$ at the vertex $\mathbf{v}$. Conceptually, we consider $f_\mathbf{v}$ as the Taylor expansion of a globally implicitly defined surface $F = \{\mathbf{z} \in \mathbb{R}^3 : f(\mathbf{z}) = 0\}$ about the vertex $\mathbf{v}$. Therefore we can refer to $\mathbf{g}(\mathbf{v})$ and $\mathbf{H}(\mathbf{v})$ as local gradient and local Hessian matrix at $\mathbf{v}$, respectively.

A possible construction of the function $f_\mathbf{v}$ is as follows. We generate for $f_\mathbf{v}$ a function of degree m

$$f_\mathbf{v}(x,y,z) = \sum_{r,s,t\in\mathbb{Z}_0^+:\ r+s+t\le m} c_{r,s,t}x^r y^s z^t,$$

satisfying

$$f_\mathbf{v}(\mathbf{v}) = 0 \tag{B.1}$$

and

$$(\bigtriangledown f_\mathbf{v})(\mathbf{v}) = \mathbf{g}(\mathbf{v}) = \mathbf{n_v}, \tag{B.2}$$

with $c_{r,s,t} \in \mathbb{R}$. Now we compute the unknown coefficients $c_{r,s,t}$ by solving the minimization problem

$$\min_{c_{r,s,t}} \sum_{\mathbf{w}\in\Omega^i(\mathbf{v})} \omega_\mathbf{w}(f_\mathbf{v}(\mathbf{w})^2 + ||(\bigtriangledown f_\mathbf{v})(\mathbf{w}) - \mathbf{n_w}||^2)$$

subject to the constraints (B.1) and (B.2), where $\Omega^i(\mathbf{v})$ is the i-ring neighborhood of vertices of $\mathbf{v}$ and $\omega_\mathbf{w}$ is the user specified weight for the vertex $\mathbf{w}$ in the i-ring neighborhood $\Omega^i(\mathbf{v})$.

The implicitly defined surface $F_\mathbf{v}$ will be used in the following subsections to describe the local behavior of the desired compatible surface strips at the vertex $\mathbf{v}$. Therefore, the choice of the implicitly defined surfaces at the vertices influences the shape of the resulting surfaces.

B.2. *Construction of boundary curves*

As the next step, we construct for each edge $\mathbf{e} = (\mathbf{v}, \mathbf{w})$ a curve $\mathbf{p_e}$ of the form (1), having a contact of order $2n$ with the implicitly defined surfaces $F_\mathbf{v}$ and $F_\mathbf{w}$ at the vertices $\mathbf{v}$ and $\mathbf{w}$, respectively, which specifies the function $\tilde{h}_\mathbf{e}$.

In detail, we choose $\tilde{h}_\mathbf{e}$ as a polynomial of degree $4n+1$ in the Bernstein-Bézier representation, i.e.

$$\tilde{h}_\mathbf{e}(t) = \sum_{i=0}^{4n+1} d_i B_i^{4n+1}(t),$$

where B_i^{4n+1} are the Bernstein polynomials of degree $4n+1$ and $d_i \in \mathbb{R}$. To get the function $\tilde{h}_\mathbf{e}$, we compute a boundary curve $\mathbf{p_e}$ which possesses a contact of order $2n$ with the implicitly defined surfaces $F_\mathbf{v}$ and $F_\mathbf{w}$ at the vertices $\mathbf{v}$ and $\mathbf{w}$, respectively, i.e.

$$\frac{\partial^i}{\partial t^i} f_\mathbf{v}(\mathbf{p_e}(t)\mid_{t=0} = 0 \text{ and } \frac{\partial^i}{\partial t^i} f_\mathbf{w}(\mathbf{p_e}(t)\mid_{t=1} = 0 \tag{B.3}$$

for $i \in \{0, \ldots, 2n\}$. Then the contact conditions (B.3) lead to a system of linear equations for the coefficients d_i of the function $\tilde{h}_\mathbf{e}$. Moreover, the contact of order $2n$ of the boundary curve with the implicitly defined surfaces guarantees that the twist compatibility condition (9) is satisfied.

B.3. *Construction of compatible implicitly defined surfaces along the boundary curves*

Finally, we generate a family of implicitly defined surfaces $F_\mathbf{q} = \{\mathbf{z} \in \mathbb{R}^3 : f_\mathbf{q}(\mathbf{z}) = 0\}$ of degree n along each boundary curve $\mathbf{p_e}$. For a point $\mathbf{q} = \mathbf{p_e}(t)$ on the boundary curve, the function $f_\mathbf{q}$ is given by

$$f_\mathbf{q}(\mathbf{z}) = \underbrace{\mathbf{g}(\mathbf{q})^T \cdot (\mathbf{z}-\mathbf{q})}_{\text{only for } n\ge 1} + \underbrace{\frac{1}{2}(\mathbf{z}-\mathbf{q})^T \cdot \mathbf{H}(\mathbf{q}) \cdot (\mathbf{z}-\mathbf{q})}_{\text{only for } n\ge 2} + \ldots,$$

where $\mathbf{g}(\mathbf{q})$ is the local gradient and $\mathbf{H}(\mathbf{q})$ is the local Hessian matrix of the function $f_\mathbf{q}$ at the point $\mathbf{q}$. This family of surface is generated in such a way that it is compatible with the curve $\mathbf{p_e}$ and with the implicitly defined surfaces $F_\mathbf{v}$ and $F_\mathbf{w}$ at the vertices $\mathbf{v}$ and $\mathbf{w}$, respectively. This is achieved by simply projecting the Taylor expansions obtained by Hermite interpolation into the linear subspaces which are defined by the compatibility conditions. We explain this construction step for $n \le 2$ in detail, but it can be generalized to any $n \ge 3$, which is beyond the scope of this paper.

The function $f_{\mathbf{q}}$ is determined by zero for $n = 0$, by the local gradient $\mathbf{g}(\mathbf{q})$ for $n = 1$, and by the local gradient $\mathbf{g}(\mathbf{q})$ and the local Hessian matrix $\mathbf{H}(\mathbf{q})$ for $n = 2$. As already described before, the local gradients (and the local Hessian matrices) have to be compatible with the boundary curve $\mathbf{p_e}$, which means that the boundary curve needs to have a first (and second order contact) with the function $f_{\mathbf{q}}$ along the boundary curve, i.e.

$$\frac{\partial}{\partial s} f_{\mathbf{q}}(\mathbf{p_e}(s)) \mid_{s=t} = 0 \tag{B.4}$$

(and

$$\frac{\partial^2}{\partial s^2} f_{\mathbf{q}}(\mathbf{p_e}(s)) \mid_{s=t} = 0 \;) \tag{B.5}$$

for $\mathbf{q} = \mathbf{p_e}(t)$. The conditions (B.4) and (B.5) lead to the following conditions

$$\mathbf{g}(\mathbf{p_e}(t))^T \cdot \frac{\partial}{\partial t} \mathbf{p_e}(t) = 0 \tag{B.6}$$

and

$$\frac{\partial}{\partial t} \mathbf{g}(\mathbf{p_e}(t)) = \mathbf{H}(\mathbf{p_e}(t)) \cdot \frac{\partial}{\partial t} \mathbf{p_e}(t) \tag{B.7}$$

for the local gradients and for the local Hessian matrices, respectively.

The computation of the local gradients for $n \in \{1, 2\}$ and the local Hessian matrices for $n = 2$ works as follows.

Computation of the local gradients: We first generate an initial field of pseudo-gradient vectors $\hat{\mathbf{g}}(\mathbf{p_e})$ by Hermite interpolation of suitable boundary data at the vertices $\mathbf{v}$ and $\mathbf{w}$ of the edge $\mathbf{e}$, i.e.

$$\hat{\mathbf{g}}(\mathbf{p_e}(t)) = \sum_{j=0}^{2n-1} H_{0,j}^{4n-1}(t) \hat{\mathbf{g}}_{\mathbf{v}}^j + H_{1,j}^{4n-1}(t) \hat{\mathbf{g}}_{\mathbf{w}}^j,$$

where

$$\hat{\mathbf{g}}_{\mathbf{v}}^j = \frac{\partial^j}{\partial t^j} (\bigtriangledown f_{\mathbf{v}})(\mathbf{p_e}(t)) \mid_{t=0}$$

and

$$\hat{\mathbf{g}}_{\mathbf{w}}^j = \frac{\partial^j}{\partial t^j} (\bigtriangledown f_{\mathbf{w}})(\mathbf{p_e}(t,0)) \mid_{t=1},$$

for $j \in \{0, \ldots, 2n-1\}$.

This initial field of vectors $\hat{\mathbf{g}}(\mathbf{p_e})$, however, is not guaranteed to fulfill condition (B.6). A valid field of gradient vectors $\mathbf{g}(\mathbf{p_e})$ is then obtained by solving the minimization problem

$$\mathbf{g}(\mathbf{p_e}(t)) = \arg\min_{\bar{\mathbf{g}}} ||\bar{\mathbf{g}} - \hat{\mathbf{g}}(\mathbf{p_e}(t))||^2$$

subject to the constraint (B.6). Its solution is given in explicit form by

$$\mathbf{g}(\mathbf{p_e}(t)) = \hat{\mathbf{g}}(\mathbf{p_e}(t)) - \frac{\hat{\mathbf{g}}(\mathbf{p_e}(t))^T \cdot \frac{\partial}{\partial t} \mathbf{p_e}(t)}{\frac{\partial}{\partial t}(\mathbf{p_e}(t))^T \cdot \frac{\partial}{\partial t} \mathbf{p_e}(t)} (\frac{\partial}{\partial t}(\mathbf{p_e}(t))).$$

This possesses a simple geometric interpretation. The gradients $\mathbf{g}(\mathbf{p_e})$ are obtained as the projections of the gradients $\hat{\mathbf{g}}(\mathbf{p_e})$ into the normal plane of the curve $\mathbf{p_e}$. Moreover, this design specification of the local gradients ensures that the twist compatibility condition (9) is satisfied.

Computation of the local Hessian matrices: The construction of the local Hessian matrices works similar to the case of the local gradients. Again, we start with the construction of interpolants $\hat{\mathbf{H}}(\mathbf{p_e})$, given by

$$\hat{\mathbf{H}}(\mathbf{p_e}(t)) = \sum_{j=0}^{2} H_{0,j}^5(t) \hat{\mathbf{H}}_{\mathbf{v}}^j + H_{1,j}^5(t) \hat{\mathbf{H}}_{\mathbf{w}}^j,$$

where

$$\hat{\mathbf{H}}_{\mathbf{v}}^j = \frac{\partial^j}{\partial t^j} \mathrm{Hess}(f_{\mathbf{v}})(\mathbf{p_e}(t)) \mid_{t=0}$$

and

$$\hat{\mathbf{H}}_{\mathbf{w}}^j = \frac{\partial^j}{\partial t^j} \mathrm{Hess}(f_{\mathbf{w}})(\mathbf{p_e}(t)) \mid_{t=1},$$

for $j \in \{0, 1, 2\}$. By solving the minimization problem

$$\mathbf{H}(\mathbf{p_e}(t)) = \arg\min_{\bar{\mathbf{H}}} ||\bar{\mathbf{H}} - \hat{\mathbf{H}}(\mathbf{p_e}(t))||^2 \tag{B.8}$$

subject to the constraint (B.7), we obtain the local Hessian matrices which satisfy condition (B.7) and guarantee the fulfillment of the twist compatibility condition (9). Moreover, the construction of the Hessian matrices $\mathbf{H}(\mathbf{p_e})$ is invariant with respect to the choice of a coordinate system.

Since the functions $f_{\mathbf{q}}$ can be considered as the Taylor expansions of a globally implicitly defined surface $F = \{\mathbf{z} \in \mathbb{R}^3 : f(\mathbf{z}) = 0\}$ about the vertex $\mathbf{v}$, a triangular network of compatible surface strips of order n is given by this surface F and the boundary curves $\mathbf{p_e}$ for the edges $\mathbf{e}$ of the triangular mesh $\mathcal{M}$.

References

[1] R. E. Barnhill, G. Birkhoff, and W. J. Gordon. Smooth interpolation in triangles. *J. Approximation Theory*, 8:114–128, 1973.

[2] R. E. Barnhill and J. A. Gregory. Compatible smooth interpolation in triangles. *J. Approximation Theory*, 15(3):214–225, 1975.

[3] R. E. Barnhill and J. A. Gregory. Polynomial interpolation to boundary data on triangles. *Math. Comp.*, 29:726–735, 1975.

[4] G. Farin. *Curves and surfaces for computer-aided geometric design*. Academic Press, 1997.

[5] J. A. Gregory. Smooth interpolation without twist constraints. In *Computer aided geometric design (Proc. Conf., Univ. Utah, Salt Lake City, Utah, 1974)*, pages 71–87. Academic Press, New York, 1974.

[6] S. Hahmann and G.-P. Bonneau. Triangular G^1 interpolation by 4-splitting domain triangles. *Comput. Aided Geom. Design*, 17(8):731–757, 2000.

[7] S. Hahmann, G.-P. Bonneau, and B. Caramiaux. Bicubic G^1 interpolation of irregular quad meshes using a 4-split. In *Advances in geometric modeling and processing*, volume 4975 of *Lecture Notes in Comput. Sci.*, pages 17–32. Springer, Berlin, 2008.

[8] J. Hahn. Filling polygonal holes with rectangular patches. In *Theory and practice of geometric model-*

ing (Blaubeuren, 1988), pages 81–91. Springer, Berlin, 1989.

[9] T. Hermann. G^2 interpolation of free form curve networks by biquintic Gregory patches. *Comput. Aided Geom. Design*, 13(9):873–893, 1996. In memory of John Gregory.

[10] T. Hermann, J. Peters, and T. Strotman. A geometric criterion for smooth interpolation of curve networks. In J. Keyser, editor, *SPM 2009: 2009 SIAM/ACM Joint Conferences on Geometric and Physical Modeling*, pages 169–173. ACM, New York, 2009.

[11] T. Hermann, J. Peters, and T. Strotman. Constraints on curve networks suitable for G^2 interpolation. In B. Mourrain, S. Schaefer, and G. Xu, editors, *GMP 2010, LNCS 6130*, pages 77–87. Springer-Verlag Berlin Heidelberg, 2010.

[12] J. Hoschek and D. Lasser. *Fundamentals of computer aided geometric design*. A K Peters Ltd., Wellesley, MA, 1993.

[13] K. Karčiauskas and J. Peters. Guided spline surfaces. *Comput. Aided Geom. Design*, 26(1):105–116, 2009.

[14] Q. Liu and T. Sun. G1 interpolation of mesh curves. *Computer-Aided Design*, 26(4):259–267, 1994.

[15] J. Peters. Smooth interpolation of a mesh of curves. *Constr. Approx.*, 7(2):221–246, 1991.

[16] J. Peters. C^2 free-form surfaces of degree $(3, 5)$. *Comput. Aided Geom. Design*, 19(2):113–126, 2002.

[17] J. Peters. Geometric continuity. In *Handbook of computer aided geometric design*, pages 193–227. North-Holland, Amsterdam, 2002.

[18] H. Prautzsch. Freeform splines. *Comput. Aided Geom. Design*, 14(3):201–206, 1997.

[19] U. Reif. Biquadratic G-spline surfaces. *Comput. Aided Geom. Design*, 12(2):193–205, 1995.

[20] U. Reif. TURBS—topologically unrestricted rational B-splines. *Constr. Approx.*, 14(1):57–77, 1998.

[21] M. Szilvási-Nagy and I. Szabó. C^1-continuous Coons-type blending of triangular patches. *KoG*, 9:29–34, 2005.

[22] M. Szilvási-Nagy and I. Szabó. Generalization of Coons construction. *Computers and Graphics*, 30:588–597, 2006.

[23] H. Theisel. Are isophotes and reflection lines the same? *Comput. Aided Geom. Design*, 18(7):711–722, 2001.

[24] X. Ye. Curvature continuous interpolation of curve meshes. *Comput. Aided Geom. Design*, 14(2):169–190, 1997.