A direct and local method for computing polynomial Pythagorean-normal patches with global continuity

Bizzarri, Michal; Lávička, Miroslav; Vršek, Jan; Kosinka, Jiří

Published in:
Computer-Aided design

DOI:
10.1016/j.cad.2018.04.013

Publication date:
2018

A direct and local method for computing polynomial Pythagorean-normal patches with global G^1 continuity

Michal Bizzarri*[,b], Miroslav Lávička[a,b], Jan Vršek[a,b], Jiří Kosinka[c]

[a] *Department of Mathematics, Faculty of Applied Sciences, University of West Bohemia, Univerzitní 8, 306 14 Plzeň, Czech Republic*
[b] *NTIS – New Technologies for the Information Society, Faculty of Applied Sciences, University of West Bohemia, Univerzitní 8, 306 14 Plzeň, Czech Republic*
[c] *Bernoulli Institute, University of Groningen, Nijenborgh 9, 9747 AG Groningen, The Netherlands*

Abstract

We present a direct and local construction for polynomial G^1 spline surfaces with a piece-wise Pythagorean normal (PN) vector field. A key advantage of our method is that the constructed splines possess exact piece-wise rational offsets without any need for reparametrisations, which in turn means that no trimming procedure in the parameter domain is necessary. The spline surface consists of locally constructed triangular PN macro-elements, each of which is completely local and capable of matching boundary data consisting of three points with associated normal vectors. The collection of the macro-elements forms a G^1-continuous spline surface. The designed method is demonstrated on several examples.

Key words: Hermite interpolation, piece-wise polynomial PN surfaces, rational offsets, macro-elements

1. Introduction

Curves and surfaces satisfying a certain *Pythagorean* property of their tangent or normal vector fields have become an intensive research topic in recent years. Investigating their properties and applications significantly influenced research in related theoretical as well as applied disciplines, and nowadays one can find a large number of papers and other contributions related to this interesting concept [1, 2].

This paper is devoted to surfaces in 3-space whose normal vectors satisfy the Pythagorean property, the so-called PN surfaces. Rational PN surfaces were defined in [3] as a surface counterpart of Pythagorean hodograph (PH) curves [4]. It holds that PH curves in the plane and PN surfaces in 3-space share some common properties, for instance they both yield rational offsets. This property is highly appreciated in technical practice since for general free-form NURBS curves and surfaces an exact (piece-wise) rational parametric representation of their offsets is not available, and approximate techniques for computing and interrogating their offsets are thus needed.

Nonetheless, when considering only the rationality of offsets as a main feature of PH curves or PN surfaces then other useful properties might be overlooked. In the curve case, another very important practical application is based on the fact that the parametric speed (or the length element), and thus also the arc length, of polynomial PH curves is also polynomial. This is important, for instance, when formulating efficient real time interpolator algorithms for CNC machines. The area element and the surface area are then the analogues in the surface case: they are both polynomial for polynomial PN surfaces. This feature is useful for instance in CNC painting. This shows the prominent role of polynomial PH curves and PN surfaces within their rational families.

Despite the fact that both PH curves and PN surfaces belong among hypersurfaces with a Pythagorean property, one can find important differences between these two classes. For instance, PH curves were first introduced as planar *polynomial* shapes, including a compact formula for their description based on Pythagorean polynomial triples, whereas a description of *rational* PN surfaces using their duals was first revealed in [3]. This has clear consequences for formulating interpolation/approximation algorithms with these shapes. There exist many Hermite interpolation results for polynomial PH curves [1, 2], but there are not many algorithms for PN surface interpolation. Moreover, only select few are *direct* PN surface algorithms and the majority of those use rational PN surfaces. A direct PN algorithm is a construction of the object together with its PN parametrisation (i.e., no reparametrisation is required).

In contrast, results of *indirect* PN algorithms are surfaces which become PN only after a suitable rational reparametrisation, i.e., one does not obtain a polynomial PN surface but a rational one. For instance, in [5] a method for the construction of exact offsets of quadratic triangular Bézier surface patches was designed. These

*Corresponding author
Email addresses: bizzarri@ntis.zcu.cz (Michal Bizzarri), lavicka@kma.zcu.cz (Miroslav Lávička), vrsekjan@kma.zcu.cz (Jan Vršek), j.kosinka@rug.nl (Jiří Kosinka)

patches are in fact PN surfaces but their PN parametrisations were obtained only via a certain reparametrisation. A nice approach also based on reparametrisations was formulated in [6], using surfaces with linear normals [7].

As for direct methods, a scheme with triangular patches on parabolic Dupin cyclides was designed in [8], interpolation of triangular data using the support function was studied in [9], and using bicubic Coons patches in the isotropic model for the construction of smooth PN surfaces was investigated in [10].

The key advantage of direct PN interpolation techniques is obvious: as no reparametrisation is required, one does not need to apply trimming in parameter space. Nevertheless, as in the case of indirect approaches, all abovementioned direct methods yield rational PN surfaces, and thus cannot be used when polynomial parametrisations are required. Only recently, the first method solving the Hermite problem directly, and thus yielding polynomial PN parametrisations, was formulated in [11]. However, the method is global and requires solving a global linear system; the locality of e.g. the (rational) method presented in [10] is lost.

In the present paper, based on reformulating the approaches taken in [6] and [11], we solve the challenging problem of designing a PN Hermite interpolation method which

- is local, i.e., a PN macro-element is computed only from vertex and normal data of one triangle at a time;
- is direct, i.e., it yields polynomial PN macro-elements with no need for reparametrisations;
- yields globally G^1-continuous PN spline surfaces.

We describe our algorithm in Section 3, present examples in Section 4, and conclude the paper in Section 5. But before all that, we recall some preliminary notions and set notation in the following section.

2. Preliminaries

In this section we recall some fundamental facts about surfaces with rational offsets and rational curves on them.

2.1. PN surfaces and PSN curves

For the sake of completeness, we first recall the definition of PN surfaces.

Definition 2.1. Let $\mathcal{X}$ be a rational surface for which there exists a parametrisation $\mathbf{x}(u,v):\mathbb{R}^2\to\mathbb{R}^3$ satisfying the condition

$$\|\mathbf{x}_u\times\mathbf{x}_v\|^2=\sigma^2, \tag{1}$$

where $\|\cdot\|$ denotes the Euclidean norm, $\sigma(u,v)$ is a rational function, and $\mathbf{x}_u$ and $\mathbf{x}_v$ are partial derivatives of $\mathbf{x}$ with respect to u and v, respectively. Then $\mathcal{X}$ is called a *surface with a Pythagorean normal vector field* (or a *PN surface*) and condition (1) is referred to as *PN condition* or *PN property*. A parametrisation satisfying the PN condition is called a *PN parametrisation*. If every parametrisation of $\mathcal{X}$ is PN, we call $\mathcal{X}$ a *proper* PN surface. If there exist both PN and non-PN parametrisations of $\mathcal{X}$ then we speak about a *non-proper* PN surface.

A distinguishing property of PN surfaces is that they admit two-sided rational *δ-offset surfaces*

$$\mathbf{x}_\delta=\mathbf{x}\pm\delta\frac{\mathbf{N}}{\|\mathbf{N}\|}=\mathbf{x}\pm\delta\,\frac{\mathbf{x}_u\times\mathbf{x}_v}{\sigma}, \tag{2}$$

where $\mathbf{x}(u,v)$ is a PN parametrisation of $\mathcal{X}$ and $\mathbf{N}(u,v)$ is a normal vector (at regular points of $\mathcal{X}$).

Moreover, as it holds

$$\begin{vmatrix}\mathbf{x}_u\cdot\mathbf{x}_u & \mathbf{x}_u\cdot\mathbf{x}_v\\ \mathbf{x}_u\cdot\mathbf{x}_v & \mathbf{x}_v\cdot\mathbf{x}_v\end{vmatrix}=EG-F^2=\|\mathbf{x}_u\times\mathbf{x}_v\|^2 \tag{3}$$

with E, F, G the coefficients of the first fundamental form, and the squared *area element* has the form

$$\mathrm{d}A^2=(EG-F^2)\,\mathrm{d}u^2\mathrm{d}v^2, \tag{4}$$

then PN surfaces are simultaneously *surfaces with a rational area element* in $\mathbb{R}^3$. In addition, all polynomial PN surfaces (with polynomial area element) possess piece-wise polynomial surface area

$$A(u,v)=\iint\sqrt{EG-F^2}\,\mathrm{d}u\mathrm{d}v=\iint|\sigma|\,\mathrm{d}u\mathrm{d}v.$$

Let $\mathcal{X}$ be a rational surface and $\mathcal{C}\subset\mathcal{X}$ be a rational curve on it given by the parametrisation $\mathbf{c}(t)=\mathbf{x}(u(t),v(t))$ for some rational functions $u(t)$ an $v(t)$. The normal vector field of the surface along $\mathcal{C}$ is expressed as

$$\mathbf{N}(u(t),v(t))=\mathbf{x}_u(u(t),v(t))\times\mathbf{x}_v(u(t),v(t)). \tag{5}$$

The *δ-offset of the given surface along its curve* is then defined by

$$\mathbf{x}(u(t),v(t))\pm\delta\frac{\mathbf{N}(u(t),v(t))}{\|\mathbf{N}(u(t),v(t))\|}. \tag{6}$$

Of course, the curve (6) is not rational, in general. Indeed, the formula gives a rational mapping if and only if there exists a rational function $\sigma(t)$ such that the following (Pythagorean) condition is satisfied:

$$[\mathbf{x}_u(u(t),v(t))\times\mathbf{x}_v(u(t),v(t))]^2=\sigma^2(u(t),v(t)). \tag{7}$$

Then we say that the parametrisation $\mathbf{x}(u(t),v(t))$ satisfying (7) *admits Pythagorean surface normals with respect to* $\mathcal{X}$, and is shortly called a *PSN parametrisation*. A curve $\mathcal{C}\subset\mathcal{X}$ admitting a PSN parametrisation is called a *PSN curve*; see [12].

The PSN condition (7) can be satisfied for some curves despite the fact that the PN condition (1) does not hold

for the given surface parametrisation. On the other hand, when the parametrisation $\mathbf{x}(u,v)$ of the surface $\mathcal{X}$ is PN, then any parametrisation $\mathbf{x}(u(t),v(t))$ of the curve $\mathcal{C}\subset\mathcal{X}$ is PSN. Nevertheless, we emphasise that not every rational curve on a PN surface is PSN; this can happen when the surface is a non-proper PN surface.

2.2. Polynomial PN triangles

Our goal is to construct a smooth piece-wise polynomial PN surface interpolating given G^1 data, i.e., points and normals. We assume that the input data are organised in a triangular manifold mesh (with or without boundary). Before proceeding, we first reformulate the expressions involved in the PN property for Bézier triangular patches.

With $\mathbf{i}=(i,j,k)$, $|\mathbf{i}|=i+j+k$ and $i,j,k\geq 0$, a triangular Bézier surface patch of degree n is defined as

$$\mathbf{x}(\mathbf{u})=\sum_{|\mathbf{i}|=n}\frac{n!}{i!j!k!}\mathbf{x}_{\mathbf{i}}u^iv^jw^k \tag{8}$$

with barycentric coordinates $\mathbf{u}=(u,v,w)$, $u,v,w\geq 0$, $u+v+w=1$, and control points $\mathbf{x}_{\mathbf{i}}\in\mathbb{R}^3$; see [13]. The domain of the patch is a triangle $\triangle\subset\mathbb{R}^2$.

The first directional derivatives with directions parallel to the edges of $\triangle$ are

$$\begin{aligned}\mathbf{x}_u(\mathbf{u}) &= n\sum_{|\mathbf{i}|=n-1}\Delta_u\mathbf{x}_{\mathbf{i}}u^iv^jw^k,\\ \mathbf{x}_v(\mathbf{u}) &= n\sum_{|\mathbf{i}|=n-1}\Delta_v\mathbf{x}_{\mathbf{i}}u^iv^jw^k,\\ \mathbf{x}_w(\mathbf{u}) &= n\sum_{|\mathbf{i}|=n-1}\Delta_w\mathbf{x}_{\mathbf{i}}u^iv^jw^k,\end{aligned} \tag{9}$$

where

$$\begin{aligned}\Delta_u\mathbf{x}_{\mathbf{i}} &= \mathbf{x}_{ijk+1}-\mathbf{x}_{ij+1k},\\ \Delta_v\mathbf{x}_{\mathbf{i}} &= \mathbf{x}_{i+1jk}-\mathbf{x}_{ijk+1},\\ \Delta_w\mathbf{x}_{\mathbf{i}} &= \mathbf{x}_{ij+1k}-\mathbf{x}_{i+1jk},\end{aligned} \tag{10}$$

and it holds that

$$\mathbf{x}_u(\mathbf{u})+\mathbf{x}_v(\mathbf{u})+\mathbf{x}_w(\mathbf{u})\equiv\mathbf{0}. \tag{11}$$

A possible field of normal vectors can then be computed as the cross product of any two directional derivatives, e.g.

$$\mathbf{N}(\mathbf{u})=\mathbf{x}_u(\mathbf{u})\times\mathbf{x}_v(\mathbf{u}). \tag{12}$$

If there exists a polynomial $\sigma(\mathbf{u})$ such that

$$\|\mathbf{N}(\mathbf{u})\|^2=\sigma(\mathbf{u})^2 \tag{13}$$

is satisfied, we arrive at a polynomial PN triangle. Analogously, we can consider PSN curves on triangular surface patches; cf. (7).

Remark 2.2. Note that the term 'PN triangle' exists in the literature in a different sense from ours: curved Point-Normal triangles, or often simply just PN triangles, defined in [14]. These are cubic triangles matching point and normal information at the vertices of an input (flat) triangle, and globally form a G^0 surface primarily aimed for computer graphics applications. In our paper, we use the term PN triangle solely in connection with the Pythagorean-normal (PN) property.

3. Hermite interpolation with PN macro-elements

We now focus on the problem of turning a given triangular manifold mesh with associated vertex normals into a G^1 PN surface spline. As our construction is local, we consider only one triangle at a time.

Given three points $\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3\in\mathbb{R}^3$ and three corresponding unit normal vectors $\mathbf{n}_1,\mathbf{n}_2,\mathbf{n}_3$ on the unit sphere $\mathcal{S}^2$, we construct an interpolating triangular PN macro-element. To achieve global G^1 continuity while maintaining locality of the method, each boundary curve of the macro-element and also the surface normals along it must depend only on two vertices and their normals.

This approach is an extension of the method presented in [6] for surfaces with LN normals (LN surfaces). One of the main and necessary modifications is the construction of a suitable polynomial vector field with the Pythagorean property. To this end, we build upon the results of [11].

Our method consists of three steps:

1. a polynomial vector field satisfying the PN property and interpolating the directions $\mathbf{n}_1,\mathbf{n}_2,\mathbf{n}_3$ is constructed (Section 3.1);
2. three boundary PSN curves of the sought-after patch joining $\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3$ and respecting the normal field constructed in Step 1 are determined (Section 3.2);
3. a polynomial PN macro-element respecting the normal field of Step 1 and the PSN boundary curves of Step 2 is computed (Section 3.3).

Remark 3.1. In [6], the authors assume that the input data are sampled from a (part of) smooth surface with solely elliptic or hyperbolic points. The reason is that a whole line of parabolic points may correspond to a singular point of the normal vector field, which causes problems in their method. This occurs when the surface normal along a parabolic line is constant. In our approach, we allow surfaces with all types of points but assume that the input triangulation with normals respects parabolic points: parabolic points can occur only at one vertex or along one side of a triangle. Hence, the interiors of triangles correspond to solely elliptic or hyperbolic points.

We now address each of the three steps of our method in detail.

3.1. Polynomial PN normal vector fields

Consider $\mathbf{n}_1,\mathbf{n}_2,\mathbf{n}_3$ as points on the unit sphere $\mathcal{S}^2$. We construct a spherical triangular patch having these points as its vertices, and arcs of great circles connecting these points as its boundaries. Then by omitting the least common denominator of such constructed parametrisation, we obtain the sought-after polynomial PN normal vector field.

A great circular arc on $\mathcal{S}^2$ connecting $\mathbf{n}_i,\mathbf{n}_j$, $i,j\in\{1,2,3\}$, $i\neq j$, can be parametrised as follows:

$$\mathbf{a}(t)=\frac{\sum_{i=0}^2\binom{2}{i}\mathbf{a}_i\alpha_it^i(1-t)^{2-i}}{\sum_{i=0}^2\binom{2}{i}\alpha_it^i(1-t)^{2-i}},\quad t\in[0,1], \tag{14}$$

where

$$\begin{aligned} \mathbf{a}_0 &= \mathbf{n}_i, \quad \mathbf{a}_1 = \frac{\mathbf{n}_i + \mathbf{n}_j}{1 + \mathbf{n}_i \cdot \mathbf{n}_j}, \quad \mathbf{a}_2 = \mathbf{n}_j, \\ \alpha_0 &= \alpha_2 = 1, \quad \alpha_1 = \sqrt{\frac{1 + \mathbf{n}_i \cdot \mathbf{n}_j}{2}}. \end{aligned} \tag{15}$$

According to (15), we set

$$\begin{aligned} &\mathbf{c}_{002} = \mathbf{n}_1, \quad \mathbf{c}_{020} = \mathbf{n}_2, \quad \mathbf{c}_{200} = \mathbf{n}_3, \\ &\gamma_{002} = \gamma_{020} = \gamma_{200} = 1, \\ &\mathbf{c}_{011} = \frac{\mathbf{n}_1 + \mathbf{n}_2}{1 + \mathbf{n}_1 \cdot \mathbf{n}_2}, \quad \gamma_{011} = \sqrt{\frac{1 + \mathbf{n}_1 \cdot \mathbf{n}_2}{2}}, \\ &\mathbf{c}_{101} = \frac{\mathbf{n}_1 + \mathbf{n}_3}{1 + \mathbf{n}_1 \cdot \mathbf{n}_3}, \quad \gamma_{101} = \sqrt{\frac{1 + \mathbf{n}_1 \cdot \mathbf{n}_3}{2}}, \\ &\mathbf{c}_{110} = \frac{\mathbf{n}_2 + \mathbf{n}_3}{1 + \mathbf{n}_2 \cdot \mathbf{n}_3}, \quad \gamma_{110} = \sqrt{\frac{1 + \mathbf{n}_2 \cdot \mathbf{n}_3}{2}}. \end{aligned} \tag{16}$$

Next, employing the stereographic projection

$$\pi : \mathcal{S}^2 \setminus \{\mathbf{w}\} \to \mathcal{P}_\mathbf{w} : \mathbf{w} \cdot \mathbf{x} = 0, \quad \mathbf{x} \mapsto \frac{\mathbf{x} - (\mathbf{x} \cdot \mathbf{w})\mathbf{w}}{1 - \mathbf{x} \cdot \mathbf{w}} \tag{17}$$

with the centre $\mathbf{w} \in \mathcal{S}^2$ we arrive at three circles in the plane $\mathcal{P}_\mathbf{w}$, which determine the planar quadratic patch

$$\mathbf{p}(\mathbf{u}) = \frac{\displaystyle\sum_{|\mathbf{i}|=2} \frac{2!}{i!j!k!} \mathbf{b}_\mathbf{i} \beta_\mathbf{i} u^i v^j w^k}{\displaystyle\sum_{|\mathbf{i}|=2} \frac{2!}{i!j!k!} \beta_\mathbf{i} u^i v^j w^k} \tag{18}$$

with

$$\mathbf{b}_\mathbf{i} = \frac{\mathbf{c}_\mathbf{i} - (\mathbf{c}_\mathbf{i} \cdot \mathbf{w})\mathbf{w}}{1 - \mathbf{c}_\mathbf{i} \cdot \mathbf{w}}, \quad \beta_\mathbf{i} = \gamma_\mathbf{i}(1 - \mathbf{c}_\mathbf{i} \cdot \mathbf{w}). \tag{19}$$

Lifting (18) back onto the sphere using

$$\pi^{-1} : \mathcal{P}_\mathbf{w} \to \mathcal{S}^2 \setminus \{\mathbf{w}\}, \quad \mathbf{x} \mapsto \frac{(\mathbf{x} \cdot \mathbf{x} - 1)\mathbf{w} + 2\mathbf{x}}{1 + \mathbf{x} \cdot \mathbf{x}}, \tag{20}$$

we arrive at a parametrised spherical patch, denoted by $\mathbf{n}(\mathbf{u})$ in Fig. 1. Finally, considering only the numerator of this parametrisation we get the sought-after quartic PN normal vector field

$$\mathbf{N}(\mathbf{u}) = \sum_{|\mathbf{i}|=4} \frac{4!}{i!j!k!} \mathbf{N}_\mathbf{i} u^i v^j w^k, \tag{21}$$

with control points

$$\begin{aligned} &\mathbf{N}_{004} = \mathbf{H}^{002}_{002}, \quad \mathbf{N}_{040} = \mathbf{H}^{020}_{020}, \quad \mathbf{N}_{400} = \mathbf{H}^{200}_{200}, \\ &\mathbf{N}_{013} = \mathbf{H}^{002}_{011}, \quad \mathbf{N}_{103} = \mathbf{H}^{002}_{101}, \quad \mathbf{N}_{130} = \mathbf{H}^{020}_{110}, \\ &\mathbf{N}_{031} = \mathbf{H}^{011}_{020}, \quad \mathbf{N}_{301} = \mathbf{H}^{101}_{200}, \quad \mathbf{N}_{310} = \mathbf{H}^{110}_{200}, \\ &\mathbf{N}_{022} = \tfrac{1}{3}\mathbf{H}^{002}_{020} + \tfrac{2}{3}\mathbf{H}^{011}_{011}, \quad \mathbf{N}_{202} = \tfrac{1}{3}\mathbf{H}^{002}_{200} + \tfrac{2}{3}\mathbf{H}^{101}_{101}, \\ &\mathbf{N}_{220} = \tfrac{1}{3}\mathbf{H}^{020}_{200} + \tfrac{2}{3}\mathbf{H}^{110}_{110}, \quad \mathbf{N}_{112} = \tfrac{1}{3}\mathbf{H}^{002}_{110} + \tfrac{2}{3}\mathbf{H}^{011}_{101}, \\ &\mathbf{N}_{121} = \tfrac{1}{3}\mathbf{H}^{020}_{101} + \tfrac{2}{3}\mathbf{H}^{011}_{110}, \quad \mathbf{N}_{211} = \tfrac{1}{3}\mathbf{H}^{011}_{200} + \tfrac{2}{3}\mathbf{H}^{101}_{110}, \end{aligned} \tag{22}$$

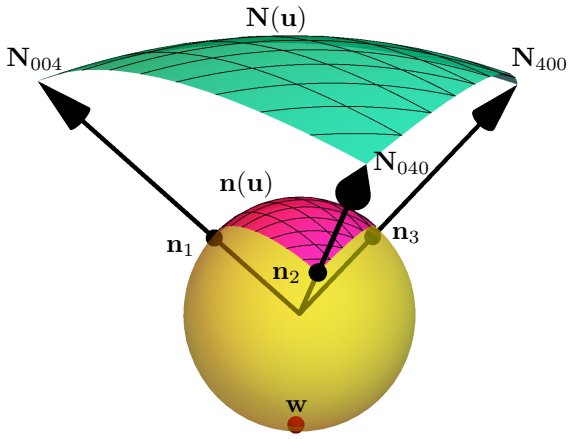

Figure 1: A spherical patch $\mathbf{n}(\mathbf{u})$ (magenta) with corners at $\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3$ and boundaries as the great circular arcs connecting these points. The corresponding polynomial patch $\mathbf{N}(\mathbf{u})$ with a polynomial area element is shown in green.

where

$$\begin{aligned} \mathbf{H}^\mathbf{i}_\mathbf{j} = \gamma_\mathbf{i}\gamma_\mathbf{j} \, (&\mathbf{c}_\mathbf{i} + \mathbf{c}_\mathbf{j} - \mathbf{w} + \\ &\mathbf{w}(\mathbf{c}_\mathbf{i} \cdot \mathbf{c}_\mathbf{j}) - \mathbf{c}_\mathbf{i}(\mathbf{w} \cdot \mathbf{c}_\mathbf{j}) - \mathbf{c}_\mathbf{j}(\mathbf{w} \cdot \mathbf{c}_\mathbf{i})) \, . \end{aligned} \tag{23}$$

We now show that this construction indeed leads to the desired normal field.

Proposition 3.2. *The constructed normal vector field* $\mathbf{N}(\mathbf{u})$ *of* (21) *interpolates positive multiples of* $\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3$ *and satisfies the PN property.*

Proof. Our construction ensures that

$$\begin{aligned} \mathbf{N}_{004} &= \mathbf{N}(0,0,1) = 2(1 - \mathbf{w} \cdot \mathbf{n}_1)\mathbf{n}_1, \\ \mathbf{N}_{040} &= \mathbf{N}(0,1,0) = 2(1 - \mathbf{w} \cdot \mathbf{n}_2)\mathbf{n}_2, \\ \mathbf{N}_{400} &= \mathbf{N}(1,0,0) = 2(1 - \mathbf{w} \cdot \mathbf{n}_3)\mathbf{n}_3. \end{aligned} \tag{24}$$

The PN property follows from the construction of $\mathbf{N}(\mathbf{u})$ obtained as the numerator of $\mathbf{n}(\mathbf{u})$, which lives on $\mathcal{S}^2$ and thus satisfies $\mathbf{n}(\mathbf{u}) \cdot \mathbf{n}(\mathbf{u}) \equiv 1$. □

Remark 3.3. The centre of the stereographic projection $\mathbf{w}$ does not affect the shape of the patch on the unit sphere. However, it has influence on its parametrisation and thus the shape of the normal vector patch $\mathbf{N}(\mathbf{u})$. In all our experiments the choice $\mathbf{w} = -(\mathbf{n}_1 + \mathbf{n}_2 + \mathbf{n}_3)/\|\mathbf{n}_1 + \mathbf{n}_2 + \mathbf{n}_3\|$ worked well and is thus used as the default setting.

Turning back to Remark 3.1, if one of the triangle's edges corresponds to a parabolic line, the two normals there may coincide (e.g. $\mathbf{n}_1 = \mathbf{n}_2$). In that case the spherical patch $\mathbf{n}(\mathbf{u})$ degenerates to a great circular arc (given e.g. by $\mathbf{n}_2$ and $\mathbf{n}_3$), and thus also the final normal vector field $\mathbf{N}(\mathbf{u})$ is geometrically degenerated into a curve segment. Nevertheless, they are still parametrised as triangular patches and therefore pose no problems in further steps, as shown and discussed below.

3.2. Polynomial PSN boundaries of the PN patch

We want to construct the PSN boundary curves of a certain (lowest possible) degree d of the sought-after PN macro-element with the normal vector field $\mathbf{N}(\mathbf{u})$ in the form:

$$\begin{aligned}\mathbf{h}_1(t) &= \textstyle\sum_{i=0}^{d} \binom{d}{i} \mathbf{h}_{i,d-i,0} t^i (1-t)^{d-i},\\ \mathbf{h}_2(t) &= \textstyle\sum_{i=0}^{d} \binom{d}{i} \mathbf{h}_{d-i,0,i} t^i (1-t)^{d-i},\\ \mathbf{h}_3(t) &= \textstyle\sum_{i=0}^{d} \binom{d}{i} \mathbf{h}_{0,i,d-i} t^i (1-t)^{d-i}.\end{aligned} \tag{25}$$

These boundary curves have to interpolate $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ and respect $\mathbf{N}(\mathbf{u})$. The former condition is trivially satisfied by setting

$$\mathbf{h}_{00d} = \mathbf{p}_1, \quad \mathbf{h}_{0d0} = \mathbf{p}_2, \quad \mathbf{h}_{d00} = \mathbf{p}_3. \tag{26}$$

The latter condition for the edge corresponding to $\mathbf{h}_1$ can be expressed as

$$\mathbf{h}_1'(t) \cdot \mathbf{N}(t, 1-t, 0) \equiv 0 \tag{27}$$

and similarly for the other two edges. Turning back to (16) and defining

$$\begin{aligned}\mathbf{c}_1(t) &= \textstyle\sum_{i=0}^{2} \binom{2}{i} \gamma_{i,2-i,0} \mathbf{c}_{i,2-i,0} t^i (1-t)^{2-i},\\ \mathbf{c}_2(t) &= \textstyle\sum_{i=0}^{2} \binom{2}{i} \gamma_{2-i,0,i} \mathbf{c}_{2-i,0,i} t^i (1-t)^{2-i},\\ \mathbf{c}_3(t) &= \textstyle\sum_{i=0}^{2} \binom{2}{i} \gamma_{0,i,2-i} \mathbf{c}_{0,i,2-i} t^i (1-t)^{2-i},\end{aligned}$$

we observe that $\mathbf{N}(t, 1-t, 0) \,||\, \mathbf{c}_1(t)$ for all $t \in [0,1]$ (see Fig. 1) as both fields correspond to the same arc on $\mathcal{S}^2$ and agree in parameter. All combined, the orthogonality conditions read

$$\mathbf{h}_i'(t) \cdot \mathbf{c}_i(t) = 0, \quad i = 1, 2, 3. \tag{28}$$

The lowest degree d which ensures a solution of (28) exists is four. Moreover, for $d = 4$, system (28) contains six independent linear equations and nine variables and hence it yields a 3-parametric solution.

For two points $\mathbf{p}_i, \mathbf{p}_j$ and their normal vectors $\mathbf{n}_i, \mathbf{n}_j \in \mathcal{S}^2$, $i, j \in \{1, 2, 3\}$, $i \neq j$, we construct

$$\mathbf{h}(t) = \sum_{i=0}^{4} \binom{4}{i} \mathbf{g}_i t^i (1-t)^{4-i}, \tag{29}$$

with $\mathbf{g}_0 = \mathbf{p}_1$ and $\mathbf{g}_4 = \mathbf{p}_2$, and we arrive at the following equations:

$$\begin{aligned}&\mathbf{n}_1 \cdot (\mathbf{g}_1 - \mathbf{p}_1) &&= 0,\\ &\sqrt{2}(\mathbf{n}_1 + \mathbf{n}_2) \cdot (\mathbf{g}_1 - \mathbf{p}_1) + 3\chi \mathbf{n}_1 \cdot (\mathbf{g}_2 - \mathbf{g}_1) &&= 0,\\ &3\mathbf{n}_1 \cdot \big(\sqrt{2}(\mathbf{g}_2 - \mathbf{g}_1) + \chi(\mathbf{g}_3 - \mathbf{g}_2)\big) +\\ &\qquad + \mathbf{n}_2 \cdot \big(3\sqrt{2}(\mathbf{g}_2 - \mathbf{g}_1) + \chi(\mathbf{g}_1 - \mathbf{p}_1)\big) &&= 0,\\ &3\mathbf{n}_2 \cdot \big(\sqrt{2}(\mathbf{g}_3 - \mathbf{g}_2) + \chi(\mathbf{g}_2 - \mathbf{g}_1)\big) +\\ &\qquad + \mathbf{n}_1 \cdot \big(3\sqrt{2}(\mathbf{g}_3 - \mathbf{g}_2) - \chi(\mathbf{g}_3 - \mathbf{p}_2)\big) &&= 0,\\ &\sqrt{2}(\mathbf{n}_1 + \mathbf{n}_2) \cdot (\mathbf{p}_2 - \mathbf{g}_3) + 3\chi \mathbf{n}_1 \cdot (\mathbf{g}_3 - \mathbf{g}_2) &&= 0,\\ &\mathbf{n}_2 \cdot (\mathbf{g}_3 - \mathbf{p}_2) &&= 0,\end{aligned} \tag{30}$$

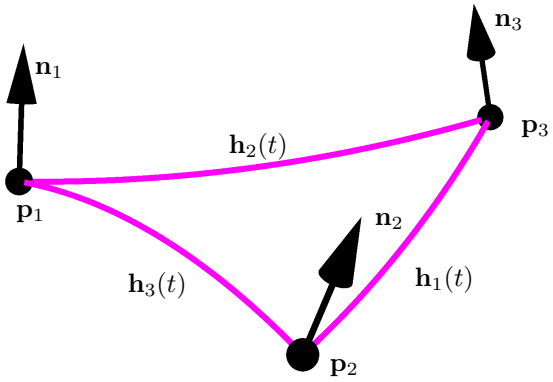

Figure 2: The uniquely determined boundary curves (magenta) having minimal squared lengths of their respective control polygons.

where $\chi = \sqrt{\mathbf{n}_1 \cdot \mathbf{n}_2 + 1}$. The free parameters are then chosen to minimise a suitable objective function with a unique minimum. We follow the approach taken in [6] and minimise the squared length of the control polygon, i.e.,

$$\sum_{i=0}^{3} \|\mathbf{g}_i - \mathbf{g}_{i+1}\|^2 \to \min; \tag{31}$$

see Fig. 2. This procedure is applied to all three boundary edges.

Lemma 3.4. *Let* $\mathbf{n}_1$ *and* $\mathbf{n}_2$ *be linearly independent. Then the linear system* (30) *with the constraint* (31) *has a solution and the boundary curve* $\mathbf{h}(t)$ *of* (29) *is determined uniquely.*

Proof. Consider two linearly independent unit normal vectors $\mathbf{n}_1$ and $\mathbf{n}_2$. The Lagrangian multiplier technique leads to a 15×15 linear system. The determinant of its coefficient matrix is equal to

$$82944\, \chi_+^{-6} \chi_-^4 \, \|\mathbf{n}_1 \times \mathbf{n}_2\|^6 \left(2\chi_+^2 + 2\sqrt{2}\chi_+ + 1\right), \tag{32}$$

where $\chi_\pm = \sqrt{\mathbf{n}_1 \cdot \mathbf{n}_2 \pm 1}$. The function $2x + 2\sqrt{2}\sqrt{x} + 1$ is positive for all $x \in \mathbb{R}^+$ and the expressions $\mathbf{n}_1 \cdot \mathbf{n}_2 \pm 1$ and $\|\mathbf{n}_1 \times \mathbf{n}_2\|$ are equal to zero if and only if $\mathbf{n}_1 = \pm \mathbf{n}_2$. □

Remark 3.5. When parabolic points $\mathbf{p}_1$ and $\mathbf{p}_2$ (sampled from the same parabolic line) are vertices of the input triangle and $\mathbf{n}_1 = \mathbf{n}_2$, then it holds

$$\mathbf{n}_1 \cdot (\mathbf{p}_1 - \mathbf{p}_2) = 0.$$

Therefore, the boundary curve of the resulting patch is the straight line-segment between $\mathbf{p}_1$ and $\mathbf{p}_2$. Thus, as long as the input triangulation respects parabolic lines (as detailed in Remark 3.1), our method produces correct results.

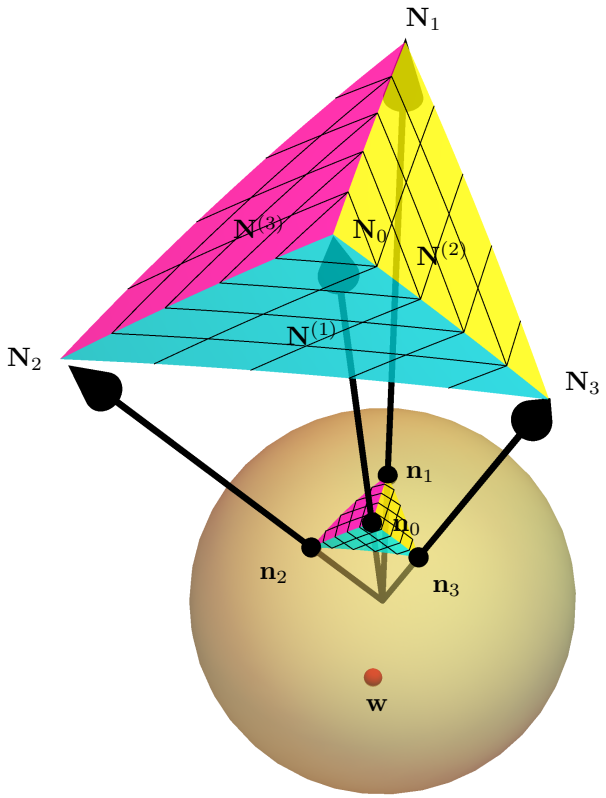

Figure 3: The normal vector field of the constructed PN macro-element consisting of three micro-triangles.

3.3. Polynomial PN macro-elements

Our goal is to construct a polynomial triangular PN patch of degree n (defined in (8)) which interpolates boundary curves $\mathbf{h}_i(t)$, see Lemma 3.4, and its associated tangent space is orthogonal to $\mathbf{N}(\mathbf{u})$ given in (21).

We first elevate the degree of $\mathbf{h}_i(t)$ from 4 to n and set

$$\begin{aligned} \mathbf{x}_{i,n-i,0} &= \mathbf{h}_{i,n-i,0} \\ \mathbf{x}_{n-i,0,i} &= \mathbf{h}_{n-i,0,i} \quad , \quad i = 0, ..., n. \\ \mathbf{x}_{0,i,n-i} &= \mathbf{h}_{0,i,n-i} \end{aligned} \tag{33}$$

Next, consider the directional derivatives in (9). Then normal field condition can be translated to the language of linear equations in the coefficients of $\mathbf{x}$ in the following way:

$$\mathbf{x}_u(\mathbf{u}) \cdot \mathbf{N}(\mathbf{u}) = 0 \quad \text{and} \quad \mathbf{x}_v(\mathbf{u}) \cdot \mathbf{N}(\mathbf{u}) = 0. \tag{34}$$

Of course, any two of the three directional derivatives can be considered due to (11).

Unfortunately, (34) does not, in general, yield a solution, therefore we employ the approach taken in [6] based on macro-elements. We split the triangle into three micro-triangles and obtain a macro-element. That is, we construct three PN patches $\mathbf{x}^{(i)}(\mathbf{u})$ forming a G^1 continuous macro-element interpolating the prescribed triangular Hermite data, as shown in Fig. 4.

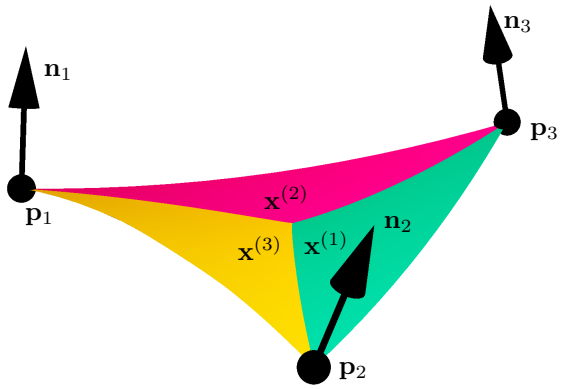

Figure 4: A polynomial PN macro-element composed of three micro-triangles $\mathbf{x}^{(i)}$, $i = 1, 2, 3$.

Given the three input normals $\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3$, we set

$$\mathbf{n}_0 = (\mathbf{n}_1 + \mathbf{n}_2 + \mathbf{n}_3)/\|\mathbf{n}_1 + \mathbf{n}_2 + \mathbf{n}_3\|$$

and define

$$\begin{aligned} \mathbf{n}_1^{(1)} = \mathbf{n}_2,\ \mathbf{n}_2^{(1)} = \mathbf{n}_3,\ \mathbf{n}_3^{(1)} = \mathbf{n}_0, \\ \mathbf{n}_1^{(2)} = \mathbf{n}_3,\ \mathbf{n}_2^{(2)} = \mathbf{n}_1,\ \mathbf{n}_3^{(2)} = \mathbf{n}_0, \\ \mathbf{n}_1^{(3)} = \mathbf{n}_1,\ \mathbf{n}_2^{(3)} = \mathbf{n}_2,\ \mathbf{n}_3^{(3)} = \mathbf{n}_0. \end{aligned} \tag{35}$$

Step 1 (Section 3.1) is then applied to each of the resulting micro-triangles; see Fig. 3. It yields $\mathbf{n}^{(i)}$ and $\mathbf{N}^{(i)}$ for $i = 1, 2, 3$.

Next, we employ Step 2 (Section 3.2) to model the three boundary PSN curves of the macro-triangle; see Fig. 2. The interpolation conditions then read

$$\mathbf{x}^{(i)}(0, t, 1-t) = \mathbf{h}_i(t), \quad i = 1, 2, 3, \tag{36}$$

one per micro-patch, or equivalently

$$\begin{aligned} \mathbf{x}^{(1)}_{0,j,n-j} &= \mathbf{h}_{i,n-i,0} \\ \mathbf{x}^{(2)}_{0,j,n-j} &= \mathbf{h}_{n-i,0,i} \quad , i = 1, 2, 3;\ j = 0, ..., n. \\ \mathbf{x}^{(3)}_{0,j,n-j} &= \mathbf{h}_{0,i,n-i} \end{aligned} \tag{37}$$

Moreover, the micro-triangles have to meet with C^0 continuity along their pair-wise shared micro-edges:

$$\mathbf{x}^{(i)}(t, 1-t, 0) = \mathbf{x}^{(i+1)}(t, 0, 1-t),\ i = 1, 2, 3, \tag{38}$$

where the index i is treated cyclically, which leads to the following conditions for control points:

$$\mathbf{x}^{(i)}_{j,n-j,0} = \mathbf{x}^{(i+1)}_{j,0,n-j}. \tag{39}$$

As the micro-patch normal vector fields form a continuous normal field macro-element, the macro-element composed of $\mathbf{x}^{(i)}$ is G^1 continuous.

Algorithm 1 Polynomial PN G^1 spline construction

Input: A triangular manifold mesh $\mathcal{M}$ respecting parabolic lines in the sense of Remark 3.1 with normals defined at vertices.

Step 1: For each triangle in $\mathcal{M}$, construct PN micro-triangle vector fields $\mathbf{N}^{(i)}(\mathbf{u})$, $i = 1, 2, 3$, given in (21) and interpolating the normals defined in (35).

Step 2: For each edge in $\mathcal{M}$, compute the unique boundary PSN quartic $\mathbf{h}(t)$ interpolating its end-points via (29). This is achieved by solving the PSN condition (28) and minimising the squared length (31) of the control polygon of $\mathbf{h}(t)$. Elevate the degree of $\mathbf{h}(t)$ to 9.

Step 3: For each triangle in $\mathcal{M}$, construct micro-elements $\mathbf{x}^{(i)}(\mathbf{u})$, $i = 1, 2, 3$, of degree 9 satisfying (36) (computed in Step 2), the continuity condition (39), and respecting the PN normal vector fields constructed in Step 1 by solving (40). A unique solution can be chosen by minimising the bending energy of the whole macro-element.

Output: A polynomial PN spline of degree 9 with global G^1 continuity interpolating the vertices and normals of $\mathcal{M}$.

Finally, we gather the system of linear equations for the free coefficients of $\mathbf{x}^{(i)}$ that ensures the PN condition (cf. (34))

$$\begin{aligned} \mathbf{x}_u^{(i)}(\mathbf{u}) \cdot \mathbf{N}^{(i)}(\mathbf{u}) &= 0 \\ \mathbf{x}_v^{(i)}(\mathbf{u}) \cdot \mathbf{N}^{(i)}(\mathbf{u}) &= 0 \end{aligned}, \quad i = 1, 2, 3. \tag{40}$$

According to our computational experiments, the resulting system is solvable for data in general position. In particular, we obtain a 3-parametric solution for polynomial PN macro elements of degree 9. A suitable solution can be chosen by, for instance, minimising the *bending energy* of the whole macro-element, yielding a unique solution; see Fig. 4.

A potential drawback of the relatively high degree of the macro-patch is outweighed by the fact that the obtained parametrisations are polynomial and directly PN, without any need for rational reparametrisations. Consequently, the rational offsets (see (2)) of the PN macro-elements are readily available, as shown in Fig 5. This concludes Step 3 for a single triangle.

Given an input triangulation, Step 1 is applied to each triangle, then Step 2 to each edge, and finally Step 3 is again applied to every triangle. The method is summarised in Algorithm 1.

We now show that not only a single PN macro-triangle is G^1 continuous as discussed above, but also the whole spline of macro-triangles constructed from an input triangulation is G^1 as well; see Fig. 6.

Lemma 3.6. *The constructed PN spline is globally G^1 continuous.*

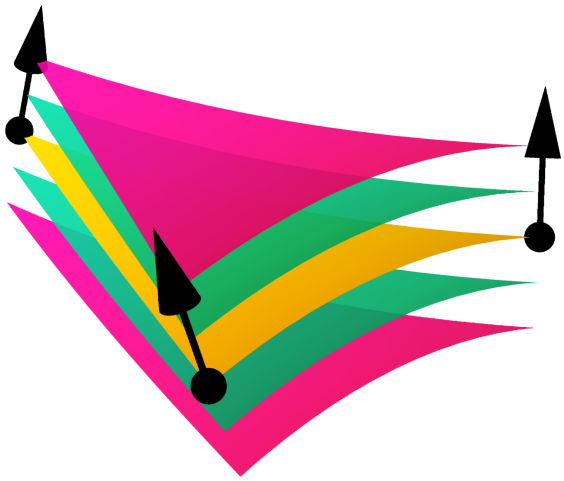

Figure 5: A polynomial PN macro element (yellow) and its rational two-sided offsets (magenta and cyan).

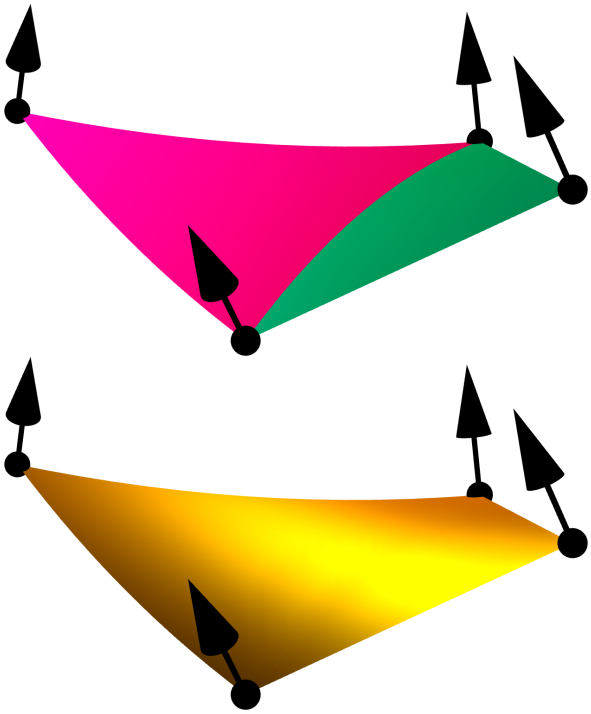

Figure 6: Two neighbouring PN macro-elements. Top: Differently coloured macro-elements. Bottom: A shaded surface indicating that the two macro-elements, and thus six micro-triangles, meet smoothly.

Proof. First, by construction it follows that the boundary edge between two neighbouring normal (micro-)patches is shared, ensuring C^0 continuity. Second, the boundary edges of the normal patches are uniquely determined and also shared by neighbouring normal field (micro-)patches, meaning that the normal field is globally G^0, which in turn implies that the PN spline is G^1 continuous over the whole input triangulation. □

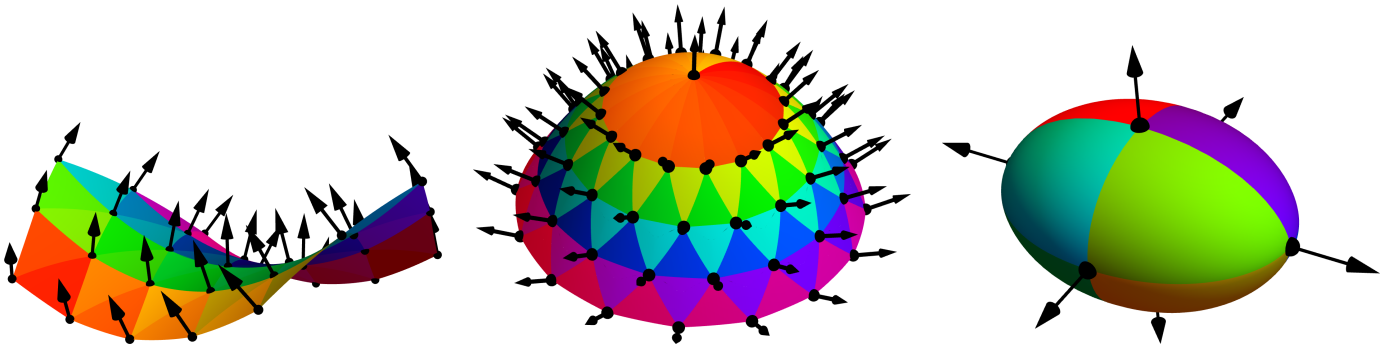

Figure 7: PN G^1 splines approximating parts of quadrics. From left to right: A hyperbolic paraboloid, an elliptic paraboloid, and an ellipsoid. Each of the surfaces has been triangulated. The interpolated normals of the underlying surfaces are shown in black.

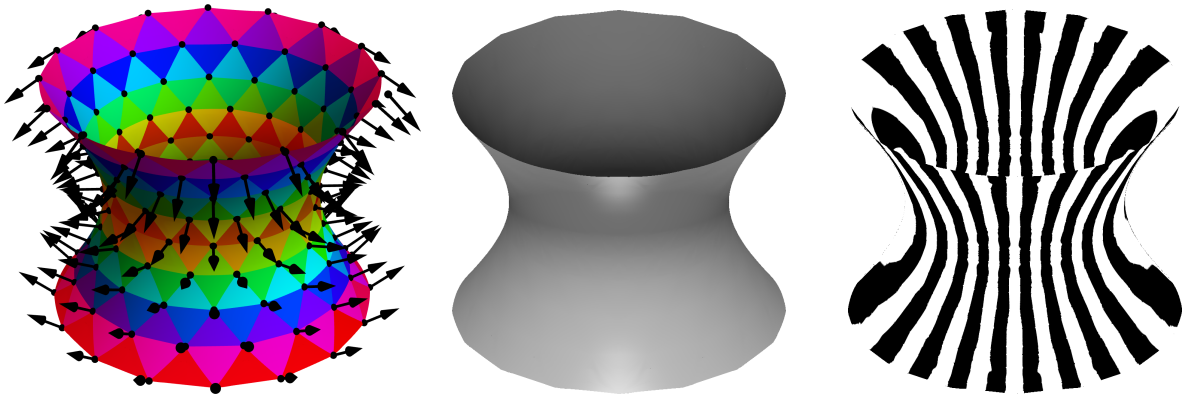

Figure 8: A PN G^1 spline constructed from a hyperboloid of one sheet. From left to right: Macro-patches and input points and normals. A rendering of the PN spline surface using Phong shading. Reflection lines on the spline surface are continuous (but not smooth), indicating that the PN spline surface is G^1 continuous.

Remark 3.7. Our method yields polynomial PN parametrisations such that

$$\mathbf{x}_u^{(i)}(\mathbf{u}) \times \mathbf{x}_v^{(i)}(\mathbf{u}) = f^{(i)}(\mathbf{u})\, \mathbf{N}^{(i)}(\mathbf{u}), \tag{41}$$

where $f^{(i)}(\mathbf{u})$ are extra factors suitably relating the degrees of $\mathbf{N}^{(i)}(\mathbf{u})$ and $\mathbf{x}^{(i)}x(\mathbf{u})$. A non-constant factor $f^{(i)}(\mathbf{u})$ indicates the existence of a curve on the surface $\mathbf{x}^{(i)}(\mathbf{u})$ where the normal field vanishes. This is analogous to what happens for non-primitive PH curves which can have cusps. However, the situation in the surface case is more complicated and all existing methods, including those of [11, 6], also result in parametrisations with non-constant extra factors. Finding surfaces without these extra factors is still an open problem and a challenging avenue for future research.

In our experience, these extra factors do not cause problems provided that the input mesh respects parabolic lines and the normals of each triangle are sufficiently close to being perpendicular to its plane. This condition can be achieved by sampling an input surface more densely when required.

4. Examples

In this section, we demonstrate the functionality of the presented method on several examples, both with and without boundary, and of different topologies. Triangulated data with normals were sampled from various quadrics: a hyperbolic paraboloid, an elliptic paraboloid, and an ellipsoid, and these were then turned into polynomial PN G^1 splines shown in Fig. 7 using Algorithm 1.

Another example is shown in Fig. 8, where the input surface is a hyperboloid of one sheet. We also include renderings using shading and reflection lines to demonstrate the smoothness of the final result.

All the above examples contained no parabolic points on the input surfaces. The fact that our algorithm handles correctly also surfaces with parabolic lines (as long as they are respected in the triangulation; see Remark 3.1) is demonstrated in Fig. 9, where the input torus contains two circles of parabolic points. Observe that each of the light green and light blue macro-elements contains two parabolic points as their vertices. Consequently, each micro-triangle containing such two parabolic vertices is a developable surface patch with a straight line-segment for one of its boundary edges, cf. Remark 3.5.

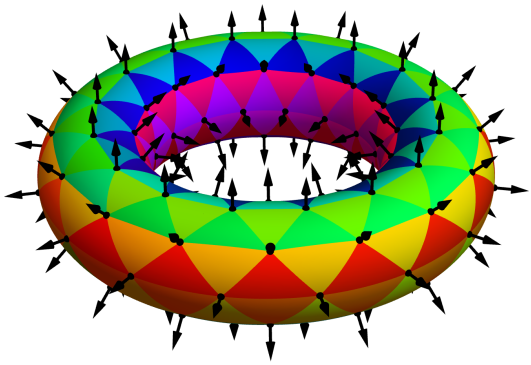

Figure 9: A torus approximated by a polynomial PN G^1 spline. The input surface contains two circles of parabolic points. Our algorithm handles that correctly as the triangular mesh respects those parabolic loci.

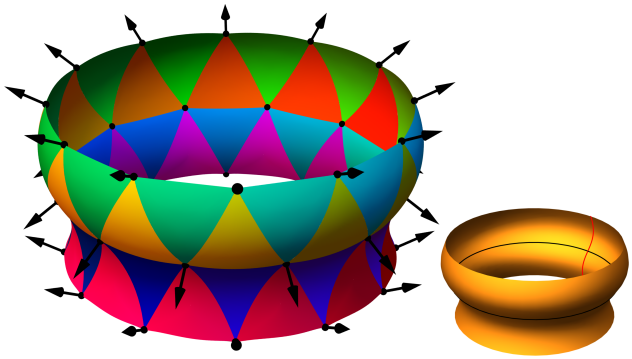

Figure 10: A surface of revolution (bottom right) with a line of parabolic points (black) generated by a cubic curve (red) with an inflection point has been approximated by a polynomial PN G^1 spline (left). The surface contains an elliptic (top half) and a hyperbolic (bottom half) region separated by a line of parabolic points with a non-constant surface normal field along it.

Yet another example with a parabolic line and hyperbolic and elliptic regions is shown in Fig. 10. In this case, the surface normal along the parabolic line is not constant.

5. Conclusion

We have presented a method for interpolating G^1 Hermite data, i.e., points and normals, organised in a triangular mesh of arbitrary manifold topology by a polynomial Pythagorean normal (PN) G^1 spline. Ours is the first such method that is purely local (each macro-elements depends only on data of a single triangle), is direct (the resulting parametrisations are PN and thus do not require any rational reparametrisation), and produces globally G^1 splines.

Our method handles parabolic points, which are known to cause issues in similar constructions, as long as the input triangulation respects, i.e., does not cross but is aligned with, parabolic curves on the input surface. This includes the special case when the surface normal is constant along a parabolic line.

To further improve the method, we plan to investigate possibilities for lowering the degree of the resulting patches.

Acknowledgements

Michal Bizzarri, Miroslav Lávička, and Jan Vršek were supported by the project LO1506 of the Czech Ministry of Education, Youth and Sports.

References

[1] R.T. Farouki. *Pythagorean-Hodograph Curves: Algebra and Geometry Inseparable*. Springer, 2008.

[2] J. Kosinka and M. Lávička. Pythagorean hodograph curves: A survey of recent advances. *Journal for Geometry and Graphics*, 18(1):23–43, 2014.

[3] Helmut Pottmann. Rational curves and surfaces with rational offsets. *Computer Aided Geometric Design*, 12(2):175–192, 1995.

[4] R.T. Farouki and T. Sakkalis. Pythagorean hodographs. *IBM Journal of Research and Development*, 34(5):736–752, 1990.

[5] B. Bastl, B. Jüttler, J. Kosinka, and M. Lávička. Computing exact rational offsets of quadratic triangular Bézier surface patches. *Computer-Aided Design*, 40:197–209, 2008.

[6] B. Jüttler and M.L. Sampoli. Hermite interpolation by piecewise polynomial surfaces with rational offsets. *Computer Aided Geometric Design*, 17:361–385, 2000.

[7] B. Jüttler. Triangular Bézier surface patches with linear normal vector field. In R. Cripps, editor, *The Mathematics of Surfaces VIII. Information Geometers*, pages 431–446, 1998.

[8] M. Peternell and H. Pottmann. Designing rational surfaces with rational offsets. In F. Fontanella, K. Jetter, and P.J. Laurent, editors, *Advanced Topics in Multivariate Approximation*, pages 275–286. World Scientific, 1996.

[9] J. Gravesen. Surfaces parametrized by the normals. *Computing*, 79(2):175–183, 2007.

[10] M. Lávička, Z. Šír, and J. Vršek. Smooth surface interpolation using patches with rational offsets. *Computer Aided Geometric Design*, 48:75–85, 2016.

[11] Michal Bizzarri, Miroslav Lávička, Zbyněk Šír, and Jan Vršek. Hermite interpolation by piecewise polynomial surfaces with polynomial area element. *Computer Aided Geometric Design*, 51:30–47, 2017.

[12] Jan Vršek and Miroslav Lávička. Surfaces with Pythagorean normals along rational curves. *Computer Aided Geometric Design*, 31(7–8):451–463, 2014.

[13] Gerald Farin. *Curves and surfaces for CAGD: A practical guide*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[14] Alex Vlachos, Jörg Peters, Chas Boyd, and Jason L. Mitchell. Curved PN triangles. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, I3D '01, pages 159–166, 2001.