Applied Mathematical Modelling 38 (2014) 2398–2413

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

Computing parallel curves on parametric surfaces

Akemi Gálvez [a], Andrés Iglesias [a,b,*], Jaime Puig-Pey [a]

[a] Department of Applied Mathematics and Computational Sciences, University of Cantabria, Avda. de los Castros s/n, 39005 Santander, Spain
[b] Department of Information Sciences, Faculty of Sciences, Toho University, 2-2-1 Miyama, 274-8510 Funabashi, Japan

ARTICLE INFO

Article history:
Received 9 August 2012
Received in revised form 17 July 2013
Accepted 8 October 2013
Available online 1 November 2013

Keywords:
Computer manufacturing
Sculptured machining
Rapid prototyping
Parallel curves
B-spline surfaces
NURBS surfaces

ABSTRACT

Generating parallel curves on parametric surfaces is an important issue in many industrial settings. Given an initial curve (called the base curve or generator) on a parametric surface, the goal is to obtain curves *on* the surface that are parallel to the generator. By *parallel curves* we mean curves that are at a given distance from the generator, where distance is measured point-wise along certain characteristic curves (on the surface) orthogonal to the generator. Except for a few particular cases, computing these parallel curves is a very difficult and challenging problem. In fact, only partial, incomplete solutions have been reported so far in the literature. In this paper we introduce a simple yet efficient method to fill this gap. In clear contrast with other existing techniques, the most important feature of our method is its *generality*: it can be successfully applied to any differentiable parametric surface and to any kind of characteristic curves on surfaces. To evaluate our proposal, some illustrative examples (not addressed with previous methods) for the cases of section, vector-field, and geodesic parallels are discussed. Our experimental results show the excellent performance of the method even for the complex case of NURBS surfaces.

1. Introduction

A classical academic and industrial problem is the determination of parallel curves on parametric surfaces. This problem appears recurrently in many industrial processes. Parallel curves are widely used, for instance, in manufacturing, for toolpath generation in sculptured surface machining. Also in rapid prototyping, to fabricate additively a solid object or assembly from CAD models by using 3D printing technologies such as laser sintering, stereolithography, and laminated object manufacturing. Parallel curves are also intensively used in many other fields, such as metrology and quality control assessment of final manufactured products.

The problem of computing parallel curves on parametric surfaces can be stated as follows: given an initial curve $\mathbf{C}$ (that will be called onwards the *base curve* or *generator*) on a parametric surface $\mathbf{S}$, the goal is to obtain curves on that surface that are parallel to $\mathbf{C}$. By *parallel curves* we mean curves that are at a given distance point-wise from $\mathbf{C}$, where the distance is measured on $\mathbf{S}$ along certain families of curves (called *characteristic curves* henceforth; see Section 2.1 for details) orthogonal to $\mathbf{C}$.

The problem of computing parallel curves is related to that of computing offsets of surfaces of objects represented as polygonal meshes. The latter has been the subject of intensive research, and several approaches can be found in the literature (see, for instance, [1,2]). This is largely due to its remarkable applications to surface reconstruction from sets of input data points in fields such as reverse engineering and the fact that polygonal meshes are the simplest and most common representation of 3D models. In comparison, the problem of computing parallel curves on parametric surfaces has received relatively little attention from the scientific community so far. Notable exceptions are the papers in [3,4], where the authors

[*] Corresponding author at: Department of Applied Mathematics and Computational Sciences, University of Cantabria, Avda. de los Castros s/n, 39005 Santander, Spain. Tel.: +34 942202062; fax: +34 942201703.
E-mail address: iglesias@unican.es (A. Iglesias).

0307-904X/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.apm.2013.10.042

addressed the problem of constructing a particular type of parallel curves, the so-called *geodesic offsets* or *geodesic parallels*, defined as the locus of points at constant distances measured from a curve on a surface along geodesic curves drawn orthogonal to that curve. Unfortunately, both approaches are extremely limited in scope. The work in [3] presents an algorithm for efficient tracking of the geodesic parallels on surfaces. The method works well but is severely restricted to the particular case of surfaces of revolution and is not applicable to other types of parametric surfaces. The authors in [4] provided an algorithm to compute such geodesic offsets on free-form NURBS surfaces. The method performs well but is strictly focused on geodesic curves and no other type of curves are supported.

Since then, to the best of our knowledge, no other attempt has been made to address this issue in the context of geometric processing or manufacturing. In other words, the existing methods reported in the literature so far provide suitable solutions for some particular cases of curves and/or surfaces but never address the problem in all its generality. Consequently, there is a lack of a unified methodology to compute parallel curves for any family of characteristic curves on general parametric surfaces. The present work is aimed at filling this gap.

1.1. Motivation of this work

Our motivation to construct parallel curves on parametric surfaces comes mainly from the field of sculptured surface machining, a point-based CNC (computer numerically controlled)-milling process where a sequence of cutter-contact points are traced by milling cutters by following a pattern of tracing or scanning usually called *tool-path topology* [5]. According to [6], those tool-path patterns can be grouped into four types: serial-pattern, radial-pattern, strip-pattern and contour-pattern. The first two groups, intended for machining an area, consist of trajectories on surfaces that are (either locally or globally) at given distances from a generator. For instance, the serial-pattern include strategies such as BC-parallel and BC-normal (BC stands for boundary curve) that typically require the computation of curves on surfaces that are parallel (or normal) to a prescribed boundary curve. The radial-pattern includes the case of contour-parallel offsets, used for machining an area or a pocket. Other strategies such as strip-parallel topology are used to remove strips of uncut-regions that typically appear in finish-machining with ball-end cutters along the sharp concave fillets [6] by employing small-size cutters along unconnected parallel trajectories on the strip. Another example of application of parallel curves is for tool-path generation procedures for pocketing, the most typical roughing operation for die-cavity machining. In this case, contour-parallel curves for flat-end or round-end cutter milling are used. Such curves can be regarded as parallel curves to the boundary-pocketing curve at given distances measured on the design-surface.

A reason why parallel curves are so often used in CNC-machining is to ensure that the space between adjacent tool-paths is kept constant in either the three-dimensional space (i.e., on the surface) or in the surface parametric domain. As pointed out in [7, pp. 286], geodesic parallels have been applied to tool-path generation in zig-zag finishing with 3-axis machining and ball-end cutter so that the scallop-height (the cusp height of the material removed by the cutter) becomes constant [8,9], thus optimizing the size of the cutter location data and consequently reducing the machining time. Recent papers [10–12] have pointed out, however, that under certain conditions the scallop-height is larger than originally expected and proposed methods to improve the process by achieving fewer and shorter tool-paths. Another recent paper [13] focuses on this problem and introduces a new approach for generating constant cusp height tool-paths by using a new metric, referred to as cusp-metric, defined from the curvature tensors of a workpiece and a tool surface and then constructing geodesic parallels on the resulting Riemannian manifold. All these works emphasized the difficulties of the constant scallop-height problem and the importance of improving current solutions for better performance and efficiency. Some of recent approaches consider the use of parallel curves (mostly geodesic parallels so far) as a suitable tool for further improvement.

A striking remark is that tool-path topology planning has been largely seen as a plain *distance minimization problem*, without taking care of the subtle details of technological "machinery" for CNC milling. From this point of view, geodesic curves are the preferred (often the must-use) mathematical tools as they exhibit the much desired minimum-distance property. But efficient strategies are not that easy; there are always many technological issues to be taken into account (machining time, surface quality, process stability, gouge avoidance, collision prevention and so on) so a balanced trade off among the various factors is usually required. This opens the door to alternative strategies where other kinds of parallel curves could be advantageous for some specialized tasks. An illuminating example can be found in [14], where gradient curves have already been mentioned as useful tools for some manufacturing problems. The present contribution is aimed at providing potential users with a general methodology to generate different collections of parallel curves (including but not restricted to geodesic parallels) on arbitrary smooth parametric surfaces and, therefore, enriching current procedures with new alternative strategies.

1.2. Aim and structure of the paper

In this paper we present a simple yet efficient method to compute parallel curves on parametric surfaces. Our approach relies on geometric-differential arguments to formulate the problem as an initial value problem (IVP) of systems of ordinary differential equations (ODEs), which can be integrated through step-by-step numerical methods [15–17]. A clear advantage of our method is its generality: it can be successfully applied to any type of parametric surfaces, no matter if they are polynomial, rational or other. For a proper formulation of the method, we only require the surfaces to be differentiable. In particular, it can readily be applied to free-form parametric surfaces such as B-splines and NURBS, by far the most important geometric entities in CAD/CAM. Similarly, the method can be applied to any kind of characteristic curves on surfaces, including those with relevance in geometric processing, manufacturing, and rapid prototyping. In fact, the geodesic offsets schemes in [3,4] are actually particular cases of our method, as it will be shown later on. To evaluate our proposal, some illustrative examples for the cases of section, vector-field and geodesic parallels on NURBS surfaces are discussed. Our experimental results show the excellent performance of the method even for the complex case of NURBS surfaces.

The structure of this paper is as follows: some mathematical preliminaries are given in Section 2. In Section 3 we describe our general approach to compute parallel curves on parametric surfaces. The method is discussed for the cases of section, vector-field, and geodesic parallels. Some illustrative examples of these families of parallel curves for the case of NURBS surfaces are described in Section 4. Comparative results for different families of curves parallel to the same base curve on the same surface are also reported in that section. Then, implementation issues are discussed in Section 5. Conclusions and future work close the paper.

2. Mathematical preliminaries

In this paper we will consider differentiable surfaces given in parametric form. Therefore, they are described by a vector-valued function of two variables:

$$\mathbf{S}(u,v) = (x(u,v), y(u,v), z(u,v)), \quad (u,v) \in \Omega \subset \mathbb{R}^2, \tag{1}$$

where u and v are the surface parameters and Ω represents the surface parametric domain. Expression (1) is called a parameterization of the surface **S**. We shall use the notation:

$$\mathbf{S}_u(u,v) = \frac{\partial \mathbf{S}(u,v)}{\partial u}, \quad \mathbf{S}_v(u,v) = \frac{\partial \mathbf{S}(u,v)}{\partial v},$$

to denote the first derivatives of **S**, which depend on the specific parameterization adopted. However, all the differential geometric characteristics of the surface employed in this paper are independent of the chosen parameterization.

For $\{u = u_0, v = v_0\}$, the partial derivatives $\mathbf{S}_u$ and $\mathbf{S}_v$ are vectors on the tangent plane to the surface at the point $\mathbf{S}(u^*, v^*)$, each being tangent to the parametric or coordinate curve $v = v_0$ and $u = u_0$, respectively. The *unit surface normal* **N** of the surface **S** is defined in terms of these parametric derivatives $\mathbf{S}_u$ and $\mathbf{S}_v$ by

$$\mathbf{N} = \frac{\mathbf{S}_u \times \mathbf{S}_v}{||\mathbf{S}_u \times \mathbf{S}_v||_2}, \tag{2}$$

where the symbol "×" is used to indicate the cross product and $||\cdot||_2$ denotes the Euclidean norm.

In the following we assume that $||\mathbf{S}_u(u,v) \times \mathbf{S}_v(u,v)||_2 \neq 0, \forall (u,v) \in \Omega$ (in other words, that the partial derivatives $\mathbf{S}_u$ and $\mathbf{S}_v$ neither become collinear nor vanish). This is called a *regular parameterization*. In this case, from Eq. (2) a unique unit surface normal **N** is defined at each point of the parametric surface **S**. The family of planes containing the normal **N** at the given point intersects the surface in a family of curves passing through that point. Each of these curves is called a *normal section curve*.

Any arbitrary curve on the surface can be described in parametric form on the surface domain Ω by $\{u = u(t), v = v(t)\}$. This expression defines a three-dimensional curve on the surface **S** given by $\mathbf{C}(t) = \mathbf{S}(u(t), v(t))$. Applying the chain rule, the tangent vector of the curve **C** at a point $\mathbf{C}(t)$ becomes:

$$\frac{d\mathbf{C}(t)}{dt} = \mathbf{S}_u \frac{du}{dt} + \mathbf{S}_v \frac{dv}{dt}. \tag{3}$$

It is useful to consider the case in which the curve **C** is parameterized by the arc-length on the surface. Its geometric interpretation is that a constant step traces a constant distance along an arc-length parameterized curve. Therefore, this parameterization is very convenient for surface interrogation issues such as measuring distances on the surface. Furthermore, some industrial operations require an uniform parameterization, so this condition has several practical applications. For example, in computer controlled milling operations, the curve path followed by the milling machine must be parameterized such that the cutter neither speeds up nor slows down along the path. Consequently, the optimal path is that parameterized by the arc-length s on the surface **S**, given by the First Fundamental Form of the surface:

$$E\left(\frac{du}{ds}\right)^2 + 2\,F\,\frac{du}{ds}\,\frac{dv}{ds} + G\left(\frac{dv}{ds}\right)^2 = 1, \tag{4}$$

where the coefficients E, F and G are given by (see [18] for details):

$$E = \mathbf{S}_u \cdot \mathbf{S}_u, \quad F = \mathbf{S}_u \cdot \mathbf{S}_v, \quad G = \mathbf{S}_v \cdot \mathbf{S}_v \tag{5}$$

and "·" is used to indicate the *dot* product. For the sake of clarity, in the following we shall refer exclusively to the parameter s to account for a curve parameterized by the arc-length on the surface. Similarly, the symbol $(.)'$ will denote the derivative operator with respect to s.

2.1. Characteristic curves

Although the method to compute parallel curves presented in this paper performs well for any family of curves orthogonal to the generator, the most interesting cases arise when characteristic curves are used for that purpose. Intuitively, the term *characteristic curve on a surface* usually refers to any curve that reflects some visual or geometric property of the surface. As such, this is an elusive (and subjective at some extent) concept, because there is not a formal definition of what is a characteristic curve on a surface, much less a formal set of mathematical conditions that the curve must meet in order to qualify as a characteristic curve. But researchers in the field have identified a list of classical families of curves that qualify as characteristic curves, simply because they account for some well-known properties of the surface. For the visual properties we can use *reflection lines* [19], *highlight lines* [20], and *isophotes* [21], which help to evaluate the behavior and aesthetics of the surface under illumination models, as it is usually done, for instance, in the automotive and aerospace industries. Geometric properties can be analyzed through *contour lines* [22], *lines of curvature* [22], *geodesic paths* [22,23], *helical curves* [24], *asymptotic lines* [25], *scalar and vector fields* [26], etc. These characteristic curves have been successfully applied to the analysis of workpieces (a field also known as surface interrogation [27–29]) obtained from manufacturing and rapid prototyping and for quality control assessments in industrial environments [6,14,7]. Other applications are in shoe industry [30] (to find surfaces that model the shoe piece with a prescribed characteristic curve called girth, which must be preserved while the designer modifies simultaneously other areas according to aesthetic criteria), in textile manufacturing for garment design [31] and in sail design [32] (to develop mathematical algorithms so that the sail can be properly broken down into small 2D pieces and cut out).

2.1.1. Vector-field curves on surfaces

A *vector-field curve* (or *maximum slope curve with respect to a given direction*) on a surface **S** is a curve which is tangent to the projection on **S** of a certain vector field $\mathbf{\Phi}$ defined on $\mathbb{R}^3$. The concept of vector-field curves is well-known in classical differential geometry and has already been applied to different problems in the context of geometric processing [33,26]. For example, the vector-field curves of the constant vector field $\mathbf{\Phi} = (0,0,1)$ associated with a terrain model surface are the classical maximum slope curves followed by water droplets thrown on **S** under the action of gravity [33]. In this paper, vector-field curves are used to generate a family of parallel curves called vector-field parallels henceforth.

The vector-field curves can be characterized as follows (see Fig. 1): let $\mathbf{P} = (x,y,z)$ be a point on surface **S**, **N** the unit normal vector to **S** at **P** and $\mathbf{\Phi}(x,y,z)$ the vector value at **P** of vector field $\mathbf{\Phi}$ defined in $\mathbb{R}^3$. The vector-field projection curve **C** at **P** (that is, its tangent direction **T**, which has the same direction as the differential arc d**C** of **C**) is tangent to the orthogonal projection of $\mathbf{\Phi}$ onto the tangent plane, Π, to **S** at **P**. If $\mathbf{V} = \mathbf{\Phi} \times \mathbf{N}$, we get:

$$\mathbf{T} = \mathbf{N} \times \mathbf{V} = (\mathbf{N} \cdot \mathbf{N}) \cdot \mathbf{\Phi} - (\mathbf{N} \cdot \mathbf{\Phi}) \cdot \mathbf{N} = \mathbf{\Phi} - (\mathbf{N} \cdot \mathbf{\Phi}) \cdot \mathbf{N}.$$

Note that the differential arc of curve **C** is orthogonal to vector **V**, that is:

$$d\mathbf{C} \cdot \mathbf{V} = 0. \tag{6}$$

When the vector-field is the gradient of a scalar field, as in the gravity field example, those curves are usually called *gradient curves*, because they are projections on the surface of the gradient curves of the considered scalar field.

2.1.2. Geodesic curves on surfaces

Given a curve **C** on a surface **S**, its geodesic curvature at a point $\mathbf{C}(s)$ is the ordinary curvature of the planar curve generated by orthogonal projection of **C** onto the tangent plane of **S** at $\mathbf{C}(s)$. A curve on a surface with identically vanishing geodesic curvature is called a *geodesic curve* (or simply a *geodesic*) of the surface. It can be proved [18] that $\mathbf{G}(s) = \mathbf{S}(u(s), v(s))$ is a geodesic if and only if u, v satisfy the system of two second-order ODEs:

$$\begin{cases} \frac{d^2u}{ds^2} + \Gamma^1_{11}\left(\frac{du}{ds}\right)^2 + 2\Gamma^1_{12}\left(\frac{du}{ds}\right)\left(\frac{dv}{ds}\right) + \Gamma^1_{22}\left(\frac{dv}{ds}\right)^2 = 0, \\ \frac{d^2v}{ds^2} + \Gamma^2_{11}\left(\frac{du}{ds}\right)^2 + 2\Gamma^2_{12}\left(\frac{du}{ds}\right)\left(\frac{dv}{ds}\right) + \Gamma^2_{22}\left(\frac{dv}{ds}\right)^2 = 0, \end{cases} \tag{7}$$

where the coefficients $\Gamma^k_{ij}, i,j,k = 1,2$ (called Christoffel's symbols) are given by:

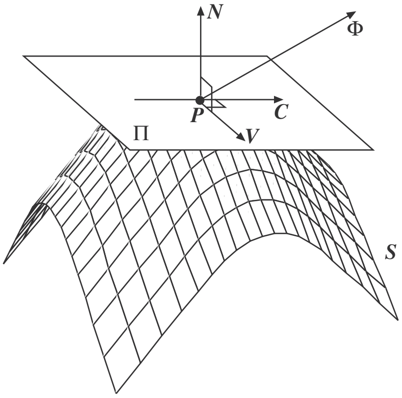

Fig. 1. Geometry of a vector-field curve.

$$\begin{aligned}
\Gamma^1_{11} &= \frac{GE_u - 2FF_u + FE_v}{2(EG - F^2)}, \quad \Gamma^2_{11} = \frac{2EF_u - EE_v + FE_u}{2(EG - F^2)},\\
\Gamma^1_{12} &= \frac{GE_v - FG_u}{2(EG - F^2)}, \quad \Gamma^2_{12} = \frac{EG_u - FE_v}{2(EG - F^2)},\\
\Gamma^1_{22} &= \frac{2GF_v - GG_u + FG_v}{2(EG - F^2)}, \quad \Gamma^2_{22} = \frac{EG_v - 2FF_v + FG_u}{2(EG - F^2)}.
\end{aligned} \tag{8}$$

It is convenient for numerical purposes to transform expression (7) into the system of four first-order ODEs:

$$\begin{cases}
\frac{du}{ds} = u',\\
\frac{dv}{ds} = v',\\
\frac{du'}{ds} = -\Gamma^1_{11}\, u'^2 - 2\Gamma^1_{12}\, u'v' - \Gamma^1_{22}\, v'^2,\\
\frac{dv'}{ds} = -\Gamma^2_{11}\, u'^2 - 2\Gamma^2_{12}\, u'v' - \Gamma^2_{22}\, v'^2.
\end{cases} \tag{9}$$

From (5) and (8) we can see that, since Christoffel's symbols involve second order derivatives of **S**, if surface **S** is C^3-continuous, geodesics are C^1-continuous. Furthermore, if **S** is C^3, then for any point **P** on **S** and for any unit vector **w** on the tangent plane of **S** at **P** there exists a unique geodesic passing through **P** with tangent vector **w**. This property will be used in Section 3.2.3.

3. Computing parallel curves on parametric surfaces

In this section we describe our general method to compute parallel curves on parametric surfaces. Firstly, we formalize the problem and describe the algorithm of the proposed method. Then, a detailed discussion about how to obtain different families of characteristic curves orthogonal to the generator is given.

3.1. The method

The problem addressed in this paper can be stated as follows: *given a base curve* **C** *on a parametric surface* **S**, *compute its parallel curves* $\mathbf{C}^p_1, \mathbf{C}^p_2, \ldots, \mathbf{C}^p_n$ *on* **S** *at given distances* $d_1, d_2, \ldots, d_n$, *respectively, from* **C**. Except for very particular cases, this problem does not have analytical solution, so we must necessarily rely on numerical procedures. A pseudocode-like description of the algorithm is reported below.

Algorithm 1. Parallel curves on parametric surfaces

```
Input:
  $\mathbf{S}(u, v)$                                   /* $\mathbf{S}$: parametric surface */
  $\mathbf{C}(s) = \mathbf{S}(u(s), v(s))$             /* $\mathbf{C}$: base curve on $\mathbf{S}$ */
  $\mathbf{D} = \{d_1, d_2, \ldots, d_n\}$             /* $\mathbf{D}$: set of distances on $\mathbf{S}$ */
Output:
  $\mathbf{C}^p$                                       /* $\mathbf{C}^p$: set of parallel curves to $\mathbf{C}$ at given distances */
Procedure:
  {Initialization}
    $\bar{\mathbf{s}} \leftarrow \{s_1, s_2, \ldots, s_m\}$    /* $\bar{\mathbf{s}}$: set of $m$ parameter values of $\mathbf{C}$ */
  {Main Loop}
    for $j = 1$ to $m$ do
      $\mathbf{P}^{\mathbf{C}}_j \leftarrow \mathbf{C}(s_j)$
      $\mathbf{T}_{\mathbf{P}^{\mathbf{C}}_j} \leftarrow \mathbf{C}'(s_j)$     /* $\mathbf{T}_{\mathbf{P}^{\mathbf{C}}_j}$: tangent vector of $\mathbf{C}$ at $\mathbf{P}^{\mathbf{C}}_j$ */
      $\mathbf{C}^\perp \leftarrow \left\{\mathbf{C}^\perp_j \;/\; (\mathbf{C}^\perp_j)'(s_j).\mathbf{T}_{\mathbf{P}^{\mathbf{C}}_j} = 0\right\}$
    end for
    for $i = 1$ to $n$ do
      for $j = 1$ to $m$ do
        $\mathbf{P}^{\mathbf{S}}_i \leftarrow \left\{\mathbf{P}_{ij} \in \mathbf{C}^\perp_j \;/\; ||\mathbf{P}_{ij} - \mathbf{P}^{\mathbf{C}}_j||_{\mathbf{S}} = d_i\right\}$    /* $||.||_{\mathbf{S}}$: norm on $\mathbf{S}$ */
        $\mathbf{P}^{\Omega}_i \leftarrow \{(u_{ij}, v_{ij}) \in \Omega \;/\; \mathbf{S}(u_{ij}, v_{ij}) = \mathbf{P}_{ij}\}$
      end for
      $\mathbf{C}^{\Omega}_i \leftarrow interp(\mathbf{P}^{\Omega}_i)$
      $\mathbf{C}^p_i \leftarrow \mathbf{S}(\mathbf{C}^{\Omega}_i)$
    end for
    $\mathbf{C}^p \leftarrow \{\mathbf{C}^p_1, \mathbf{C}^p_2, \ldots, \mathbf{C}^p_n\}$
```

Basically, what the procedure does is to compute, for m (as many as desired) points $\{\mathbf{P}_j^{\mathbf{C}}\}_j$ on $\mathbf{C}$, a curve $\mathbf{C}_j^{\perp}$ orthogonal to $\mathbf{C}$ at $\mathbf{P}_j^{\mathbf{C}}$, which belongs to a prescribed family of characteristic curves of $\mathbf{S}$. The orthogonality condition is given by $(\mathbf{C}_j^{\perp})'(s_j).\mathbf{T}_{\mathbf{P}_j^{\mathbf{C}}} = 0$, where $\mathbf{T}_{\mathbf{P}_j^{\mathbf{C}}}$ represents the tangent vector of $\mathbf{C}$ at $\mathbf{P}_j^{\mathbf{C}}$ and s_j is the parameter associated with that point. Then, we compute the points $\mathbf{P}_{ij}$ on $\mathbf{C}_j^{\perp}$ at distances d_i from $\mathbf{P}_j^{\mathbf{C}}$, where the distance is measured *on* the surface (to this aim, the norm $\|\cdot\|_{\mathbf{S}}$ on the surface is considered), along with their corresponding parametric values (u_{ij}, v_{ij}). The latter are then interpolated through cubic spline curves $\mathbf{C}_i^{\Omega}$ in Ω. Of course, other interpolating schemes might be used, but we found that cubic splines are ideal for our purposes: they are simple, fast, and the lowest-order polynomial splines endowed with inflection points. Furthermore, they compare very well with higher-order polynomials in terms of visual quality while the latter are slower and computationally more expensive. Finally, the parallel curve $\mathbf{C}_i^{p}$ can readily be obtained as $\mathbf{S}(\mathbf{C}_i^{\Omega})$. The last step is performed in order to ensure that the parallel curve lies actually on the surface. This strategy is very well suited for computer manufacturing, where interpolation schemes for tool-path generation in CNC-milling are typically described in the surface parametric domain rather than in the objects space, thus preventing potential collisions between the milling cutter and the manufactured surface due to interpolation.

3.2. Constructing characteristic curves orthogonal to the generator

A critical issue of our method is how to construct a series of m characteristic curves $\mathbf{C}_j^{\perp}$ on $\mathbf{S}$ orthogonal to $\mathbf{C}$ at $\mathbf{P}_j^{\mathbf{C}}$. Characteristic curves are generally described in terms of a geometric property. Very often, this property can be represented by equations or systems usually solved by algebraic techniques. This approach works well for academic examples, but it is not well suited for real-world instances. Industrial curves and surfaces are typically described by very complicated nonlinear functions, so algebraic techniques are no longer adequate. In this context, methods based on differential equations solved by numerical integration (often in combination with geometric considerations) have proved to be an appealing alternative [34,27,28,35]. This is also the approach considered in this section. Basically, the problem is formulated as an initial value problem of systems of ordinary differential equations, where the point $\mathbf{P}_j^{\mathbf{C}}$ plays the role of the initial condition for the ODEs.

To the aim of illustrating the method, three families of characteristic curves relevant in geometric processing and manufacturing are analyzed in next paragraphs. In what follows, let us assume we are given a base curve $\mathbf{C}(s)$ on a parametric surface $\mathbf{S}(u,v)$ parameterized by the arc-length on the surface. The unit tangent vector and unit normal vector to $\mathbf{C}(s)$ at a given point $\mathbf{P}^* = \mathbf{S}(u^*, v^*)$, denoted onwards by $\mathbf{T}^*$ and $\mathbf{N}^*$, are given by Eqs. (2) and (3), respectively.

3.2.1. Case 1: section curves

The differential equation of the section curve $\mathbf{R}(s)$ on the surface $\mathbf{S}(u,v)$ and orthogonal to $\mathbf{C}(s)$ at $\mathbf{P}^*$ can be written as:

$$\mathbf{R}'(s)\cdot\mathbf{T}^* = 0. \tag{10}$$

Since $\mathbf{R}(s)$ also lies on $\mathbf{S}(u,v)$, it is given by a parametric curve $\{u = u(s), v = v(s)\}$ on the surface domain, that is, $\mathbf{R}(s) = \mathbf{S}(u(s), v(s))$. From (3) and (10) we obtain:

$$\mathbf{S}_u\cdot\mathbf{T}^*\cdot u'(s) + \mathbf{S}_v\cdot\mathbf{T}^*\cdot v'(s) = 0. \tag{11}$$

Assuming surface arc-length regular parameterization, the corresponding differential equations for $u'(s)$ and $v'(s)$ are obtained from (4) and (11) as a system of first-order ODEs:

$$\begin{cases} \dfrac{du}{ds} = \pm\dfrac{\mathbf{S}_v\cdot\mathbf{T}^*}{\sqrt{E(\mathbf{S}_v\cdot\mathbf{T}^*)^2 - 2F(\mathbf{S}_u.\mathbf{T}^*)(\mathbf{S}_v\cdot\mathbf{T}^*) + G(\mathbf{S}_u\cdot\mathbf{T}^*)^2}}, \\ \dfrac{dv}{ds} = \mp\dfrac{\mathbf{S}_u\cdot\mathbf{T}^*}{\sqrt{E(\mathbf{S}_v\cdot\mathbf{T}^*)^2 - 2F(\mathbf{S}_u.\mathbf{T}^*)(\mathbf{S}_v\cdot\mathbf{T}^*) + G(\mathbf{S}_u\cdot\mathbf{T}^*)^2}}, \end{cases} \tag{12}$$

with initial conditions:

$$u(0) = u^*, \quad v(0) = v^*. \tag{13}$$

Each solution of (12) and (13) defines a section curve $(u(s), v(s))$ in Ω. The corresponding section curve on the surface is given by $\mathbf{S}(u(s), v(s))$, which obviously passes through the point $\mathbf{P}^*$. In addition, since s is a non-negative parameter, only two possible combinations for the signs in (12), namely (+) with $(-)$ or alternatively $(-)$ with (+), are allowed. Each of these feasible choices lead to one of the pieces of the section curve starting from $\mathbf{P}^*$, joined with continuity at this point.

3.2.2. Case 2: vector-field curves

A vector-field curve $\mathbf{F}(s)$ on $\mathbf{S}(u,v)$ orthogonal to $\mathbf{C}(s)$ at $\mathbf{P}^*$ can be constructed by taking as driving vector field $\mathbf{\Phi}^*$ the direction tangent to the surface at $\mathbf{P}^*$ and orthogonal to the tangent vector $\mathbf{T}^*$, i.e.: $\mathbf{\Phi}^* = \mathbf{T}^* \times \mathbf{N}^*$. From Section 2.1.1, taking $\mathbf{V} = \mathbf{\Phi}^* \times \mathbf{N}$ and using Eq. (6) we get:

$$\mathbf{F}'(s)\cdot\mathbf{V} = \mathbf{F}'(s)\cdot((\mathbf{T}^*\times\mathbf{N}^*)\times\mathbf{N}) = 0. \tag{14}$$

Combining (3) and (14) we get the equation:

$$\mathbf{S}_u \cdot ((\mathbf{T}^* \times \mathbf{N}^*) \times \mathbf{N}) \frac{du}{ds} + \mathbf{S}_v \cdot ((\mathbf{T}^* \times \mathbf{N}^*) \times \mathbf{N}) \frac{dv}{ds} = 0. \tag{15}$$

Assuming again that all points on the vector-field curve are regular points, a system of first-order ODEs is obtained from (4) and (15) as:

$$\begin{cases} \dfrac{du}{ds} = \mp \dfrac{\mathbf{S}_v \cdot (\mathbf{\Phi}^* \times \mathbf{N})}{\sqrt{E\,(\mathbf{S}_v \cdot (\mathbf{\Phi}^* \times \mathbf{N}))^2 - 2F\,(\mathbf{S}_u \cdot (\mathbf{\Phi}^* \times \mathbf{N}))(\mathbf{S}_v \cdot (\mathbf{\Phi}^* \times \mathbf{N})) + G\,(\mathbf{S}_u \cdot (\mathbf{\Phi}^* \times \mathbf{N}))^2}}, \\ \dfrac{dv}{ds} = \pm \dfrac{\mathbf{S}_u \cdot (\mathbf{\Phi}^* \times \mathbf{N})}{\sqrt{E\,(\mathbf{S}_v \cdot (\mathbf{\Phi}^* \times \mathbf{N}))^2 - 2F\,(\mathbf{S}_u \cdot (\mathbf{\Phi}^* \times \mathbf{N}))(\mathbf{S}_v \cdot (\mathbf{\Phi}^* \times \mathbf{N})) + G\,(\mathbf{S}_u.(\mathbf{\Phi}^* \times \mathbf{N}))^2}}, \end{cases} \tag{16}$$

with initial conditions:

$$\begin{cases} u(0) = u^*, \\ v(0) = v^*. \end{cases} \tag{17}$$

Once again, the signs of the derivatives in (16) are associated with each of the two spans of the vector-field curve **F** at both sides of $\mathbf{P}^*$. The sign of the dot product $\mathbf{F}'(s).\ \mathbf{\Phi}^*$ provides a criterion to choose the direction: it is positive when the trajectory has the same orientation as $\mathbf{\Phi}^*$ at $\mathbf{P}^*$.

3.2.3. Case 3: geodesic curves

As discussed in Section 2.1.2, geodesic curves can be constructed from Eq. (7) or, alternatively, Eq. (9). In either case, four initial conditions, u, v, u' and v', must be specified. The first two conditions can readily be assigned, as they correspond to the initial point $\mathbf{P}^*$ on the base curve. The last two define the initial direction, so they can be determined by the orthogonality condition between the tangent vectors of the base curve and the geodesic curve. To formalize it, note that the relationship $u'(s) = c\,v'(s), c \in \mathbb{R}$ describes a direction on the tangent plane to **S** at $\mathbf{P}^*$ (since substituting this expression in (3) we can get any direction on the tangent plane to **S** at $\mathbf{P}^*$). Combining this expression with (4) we get:

$$u'(s) = \frac{\pm c}{\sqrt{Ec^2 + 2Fc + G}}, \quad v'(s) = \frac{\pm 1}{\sqrt{Ec^2 + 2Fc + G}}. \tag{18}$$

Under these conditions, we can consider the following initial conditions for u' and v':

$$u'(0) = \left(\frac{du}{ds}\right)_{s=0} = u'^*, \quad v'(0) = \left(\frac{dv}{ds}\right)_{s=0} = v'^*, \tag{19}$$

where u'^*, v'^* come from (18) for the parameter value c that determines the initial direction at $\mathbf{P}^*$.

4. Results

In this section we describe some examples of application of our methodology to the families of curves analyzed in previous paragraphs. In general, the method is valid for any differentiable parametric surface. However, because of their outstanding advantages in industrial environments, their flexibility and the fact that they can represent well a wide variety of shapes, in our examples we focus on NURBS surfaces. Another reason for this choice is that NURBS surfaces are generally more difficult to deal with than most "academic" examples. Consequently, they are an excellent benchmark to evaluate our approach. The reader is referred to Appendix A for the mathematical definition of NURBS surfaces and to Appendix B for the computation of their derivatives. See also [36] for further details about NURBS surfaces.

Table 1 summarizes the basic data of the examples analyzed in this paper. For the sake of clarity, the different figures are arranged in rows. For each row, the following items are reported (in columns): surface degree in (u, v) coordinates, number of control points, number of orthogonal curves (defined by parameter m), number of parallel curves (defined by n), type of characteristic curve on the surface employed to generate the orthogonal curves and CPU runtime (in s). To enrich the discussion, a different degree for u and v has been generally considered. The same applies to other items such as the number of control points and values of parameters m and n. However, we keep the same values in Figs. 6 and 7 for comparative purposes.

Fig. 2 shows three examples of section parallels. Each example is divided into two parts: on the left, the parametric surface with the base curve (in blue), the orthogonal curves (in red),[1] and the section parallels (in black); on the right, the same curves on the surface parametric domain. A similar structure is also applied to all other examples of the paper, with the exception of Figs. 5 and 8. Figs. 3 and 4 show three and two examples of vector-field parallels and geodesic parallels, respectively. The third example of geodesic parallels is shown in Fig. 5, where the right picture of the parametric domain has been replaced by a different view of the surface and the geodesic parallels on it for better illustration.

[1] For interpretation of color in Figs. 2–8, the reader is referred to the web version of this article.

Table 1
Basic data of the examples described in this paper.

Figure	Surface degree	# Control points	# Orthogonal curves	# Parallel curves	Type of parallel curves	Runtime (in s)
2 (top)	(5,4)	7 × 4	30	1	Section	0.1039
2(middle)	(3,4)	4 × 3	60	10		0.3426
2 (bottom)	(5,4)	7 × 5	30	10		0.1921
3 (top)	(5,4)	7 × 5	30	1	Vector-field	0.1193
3 (middle)	(5,4)	3 × 5	30	14		0.2546
3 (bottom)	(5,4)	5 × 5	40	12		0.2832
4 (top)	(5,4)	7 × 5	40	2	Geodesic	0.3886
4 (bottom)	(4,4)	3 × 3	40	10		0.4202
5	(3,4)	4 × 4	40	40	Geodesic	0.7110
6 (top)	(5,4)	7 × 5	15	2	Section	0.1234
6 (middle)					Vector-field	0.1388
6 (bottom)					Geodesic	0.2646
7 (top)	(3,4)	4 × 4	40	10	Section	0.2454
7 (middle)					Vector-field	0.2770
7 (bottom)					Geodesic	0.4382

4.1. Comparative examples

An interesting issue is to compare the behavior of different parallel curves to the base curve **C** as a function of the families of characteristic curves used to compute them according to our approach. Figs. 6 and 7 show two examples of this problem; in each case, we compute the section parallels (top), vector-field parallels (middle) and geodesic parallels (bottom) to the same base curve and at the same distances measured on the same surface. By simple visual inspection, we can see that while in the first example the section, vector-field, and geodesic parallels are very similar, some differences arise for the second example (see, for instance, the curves on the front corner on the surface or, equivalently, the right-bottom corner on the parametric domain). Our experiments show that the degree of dissimilarity among different kinds of parallel curves largely depends on the distance to the generator (the further, the more different) and, especially, on the geometry of the surface those curves lie on. However, further research is still needed to elucidate this issue at full extent.

5. Implementation issues

In general, the initial-value problems of ODE systems in this paper cannot be solved analytically. Fortunately, they can be solved numerically by using methods and computer packages that are fast, reliable and widely available [15–17]. We have implemented all numerical routines of this work in *Matlab* [15]. In our case, the numerical integration has been performed by using the integrator function `ode45` of *Matlab*, based on an adaptive step-by-step technique that combines 4th- and 5th-order Runge–Kutta methods [17]. Although there is no perfect numerical procedure for solving all cases of IVPs of ODEs, in our experience this function has shown a very good behaviour. We also mention the *Numerical Recipes* routines `odeint`, `rkqs`, `rkck` that are available in C/C++, Fortran 77, and Fortran 90 languages [17]. They are adaptive step-size integration procedures based on Runge–Kutta techniques as well. In *Netlib Repository* [16] we can also find a wide variety of tested computer codes for different numerical problems, including ODEs.

In the case of surfaces comprised of several patches, the ODEs are valid inside each patch. Special care is required in the transition from one patch to another, taking into account the continuity between them. If there are no strong discontinuities, one can progress smoothly between patches, because only low-order derivatives appear in the ODEs (this is the case, for instance, in the middle and bottom examples of Fig. 2, with 6 and 12 patches, respectively). Otherwise, the border crossing point must be carefully identified, with the integration of the ODEs starting from it as the initial point for traversing the next patch. Since Runge–Kutta methods are of one-step type and only require evaluations of functions (not of their derivatives), the new integration processes arising when crossing to a new patch can be favorably treated with the above-mentioned routines.

Regarding the runtimes, Table 1 reports the CPU time (in seconds) of all examples in this paper along with their number of orthogonal and parallel curves. All experiments have been carried out in *Matlab* v.2010b on a PC platform, Intel Core 2 Duo processor at 2.66 GHz. with 4 GB of RAM. Expectedly, the runtimes are quite similar for section and vector-field parallels and slightly larger for geodesic parallels. The computation times are amazingly small, taking significantly less than one second (excluding rendering) to be obtained. Similar results were obtained for many other examples not reported here to keep the paper at reasonable length. In general, our method shows a very good performance in terms of computation time for all computed cases.

The existence of singular points on the surface, with zero values for the partial derivatives, may introduce singularities in the ODEs. Although floating point calculations make it very difficult to obtain exact zero values, this scenario could actually happen, leading to numerical instabilities when the integration reaches the neighborhood of such points. In fact, it has been

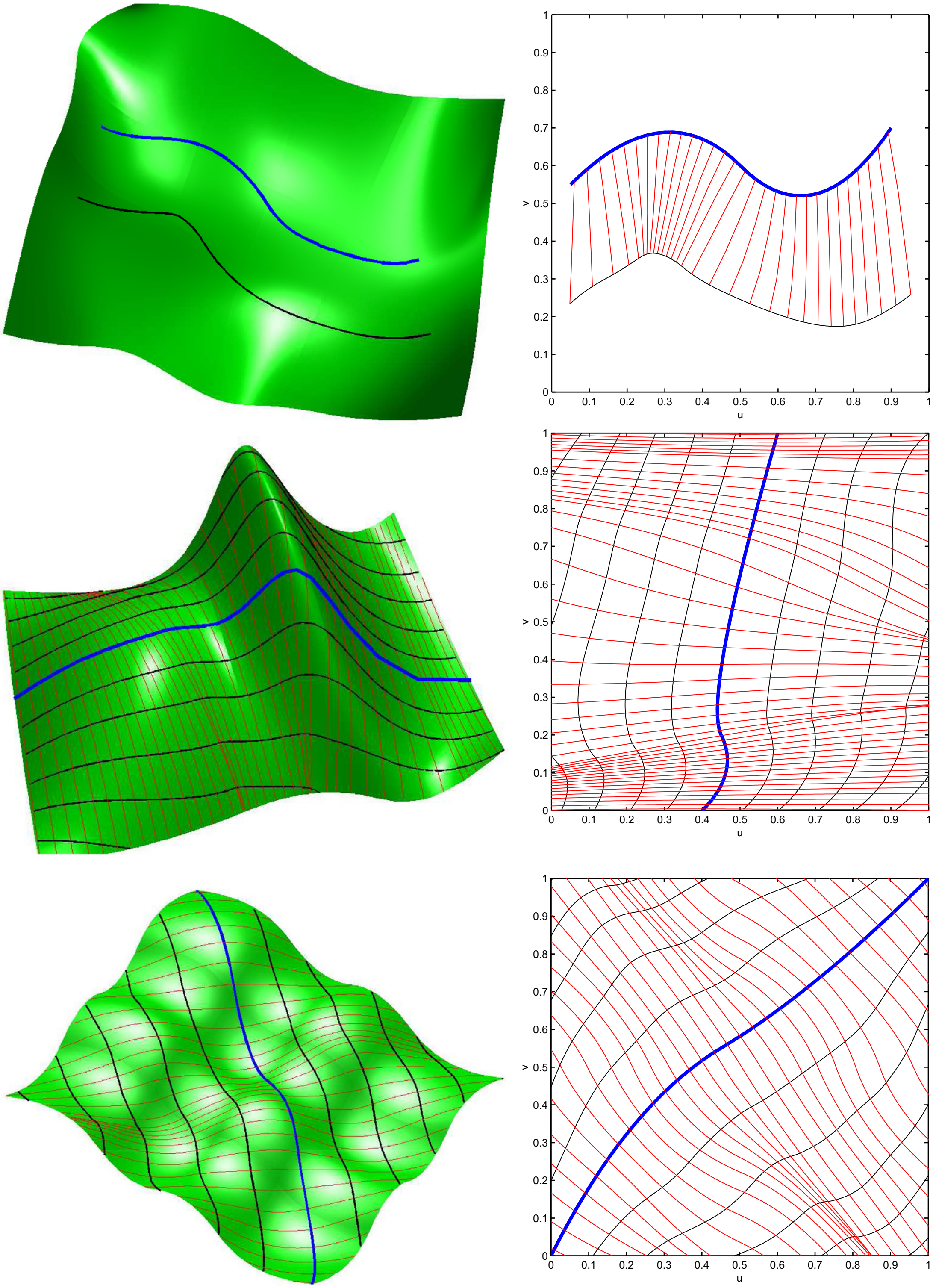

Fig. 2. Examples of section parallels: (left) on the surface; (right) on the surface parametric domain.

pointed out that it is sometimes impossible to cover the entire surface with parallel curves from the same base curve as cusps or self-intersections typically appear [13]. This is a characteristic of parallel curves on surfaces in general, not a limitation of our method specifically. Those problems are evidenced by the presence of swallowtail solutions, as in Fig. 8. A

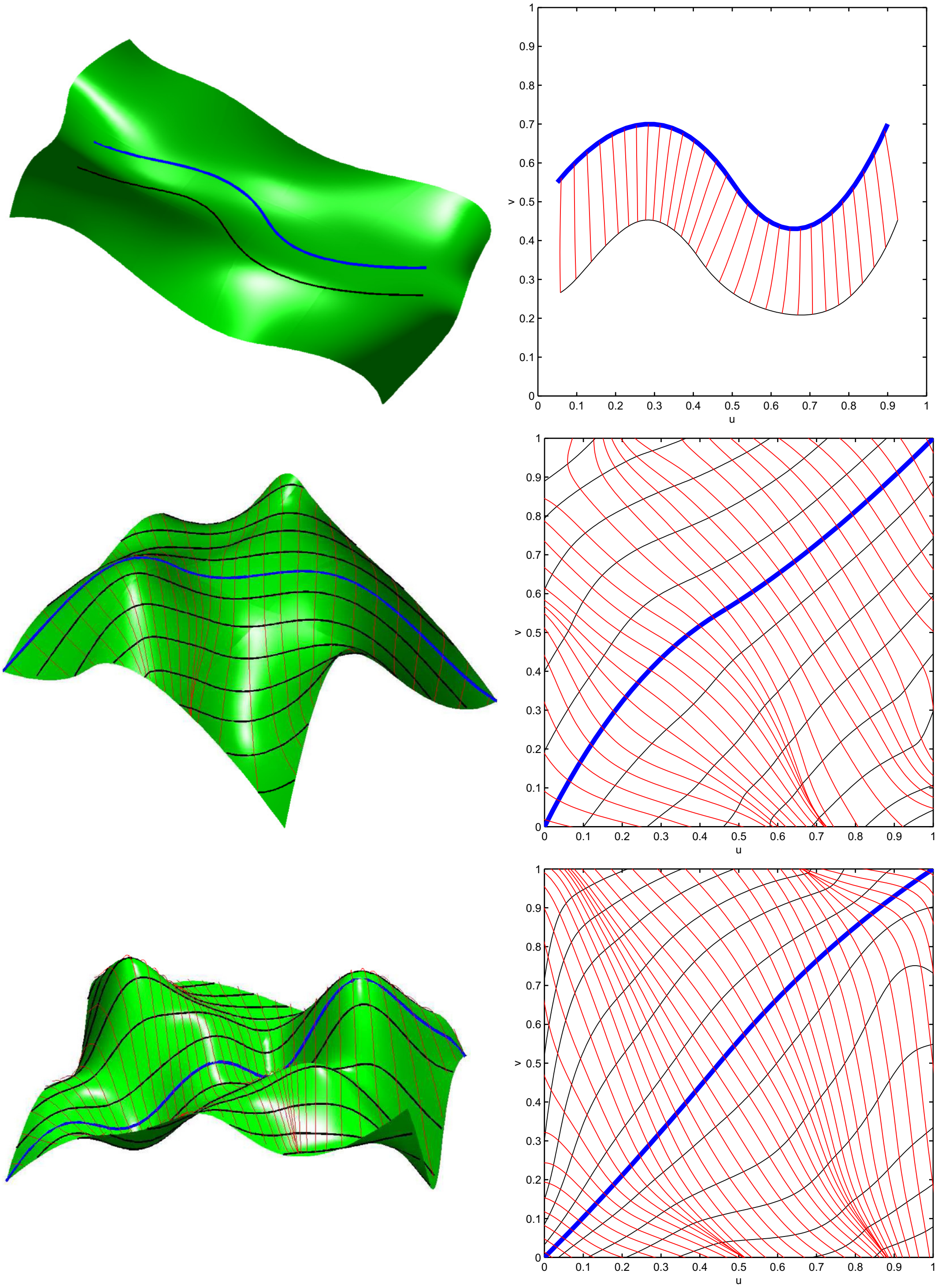

Fig. 3. Examples of vector-field parallels: (left) on the surface; (right) on the surface parametric domain.

crude solution is to confine the base curve to "safe" regions where no swallowtails occur, then consider new generators for the unvisited regions and apply our methodology on such regions, much like the author in [13] did for his constant cusp-height tool-path method. Some alternative approaches that directly remove self-intersections are also referenced in that paper.

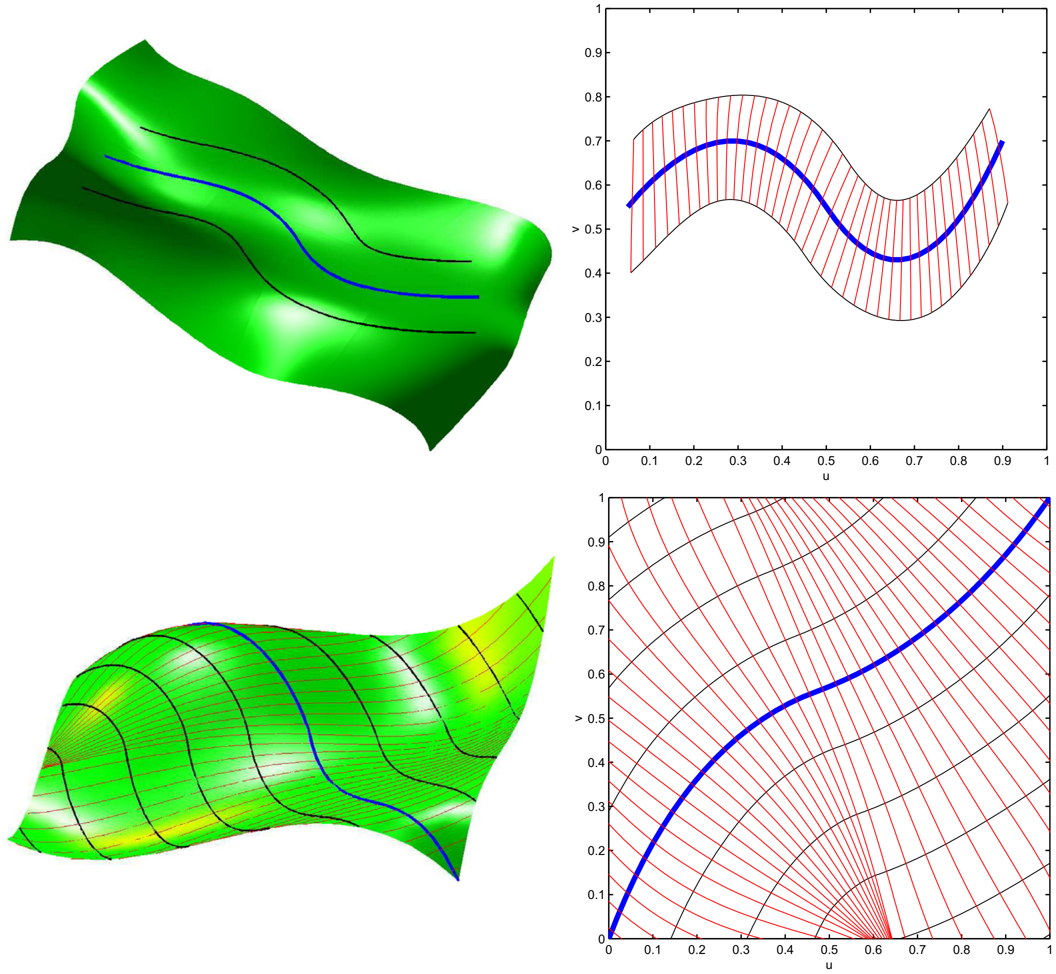

Fig. 4. Examples of geodesic parallels: (left) on the surface; (right) on the surface parametric domain.

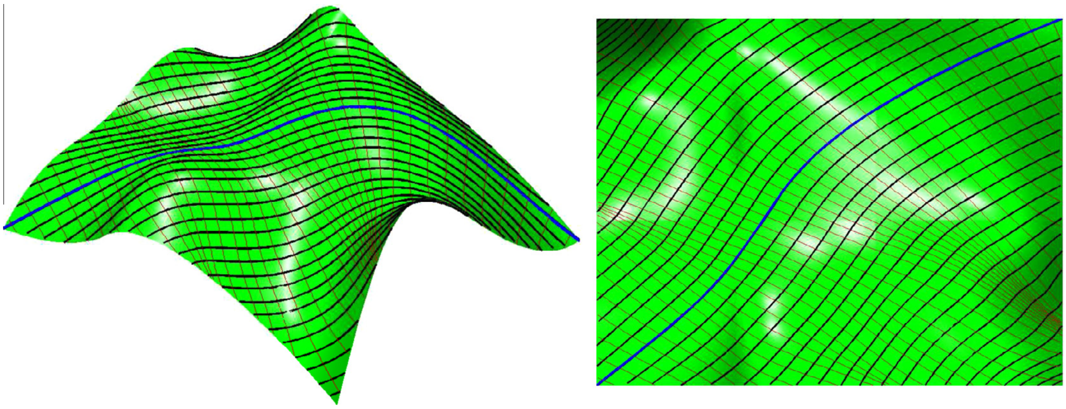

Fig. 5. Two views of a series of geodesic parallels (in black) to the generator (in blue) on a NURBS surface.

An algorithm to determine singularities and cusps for parallel curves is given in [3]. According to this author, how accurately the cusps have to be determined depends on the specific application. In most situations, the cusps appear in self-intersection scenarios, so its detection is only applied to the purpose of modeling the resulting loops correctly. To this aim, it is enough to find a point close to the cusp, not the cusp itself. The same author proposed a visual criterion for their occurrence:

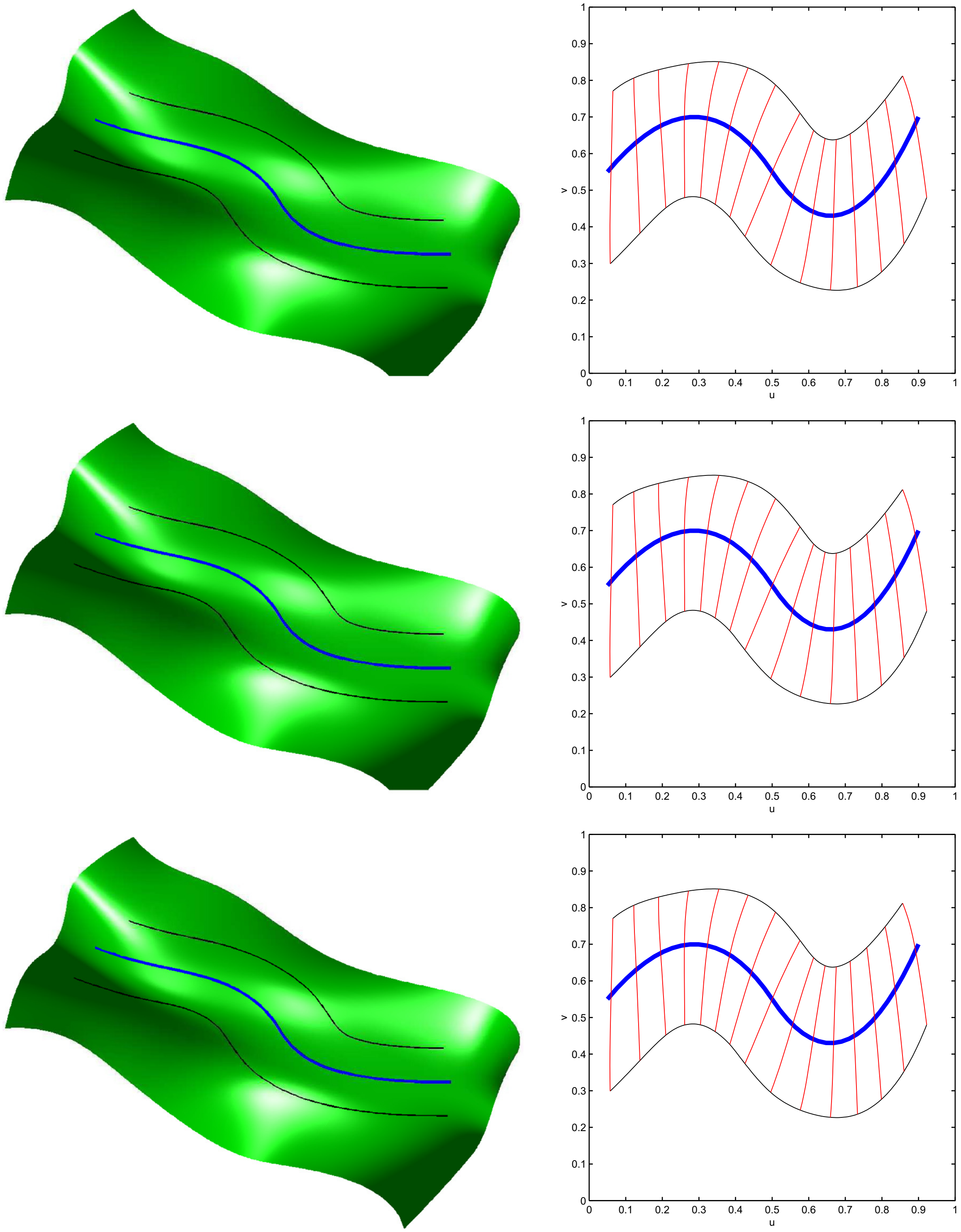

Fig. 6. Example of parallel curves to the generator: (left) on a NURBS surface; (right) on the surface parametric domain: (top) section parallels; (middle) vector-field parallels; (bottom) geodesic parallels.

equally spaced points in the parameter interval of the parallel curve are differently spaced along the parallel curve on the surface according to its parameterization. In the neighborhood of a singularity the points lie very close together. Therefore if a parameter value t^* lies in the neighborhood of the parameter value of the singularity, its corresponding point on the

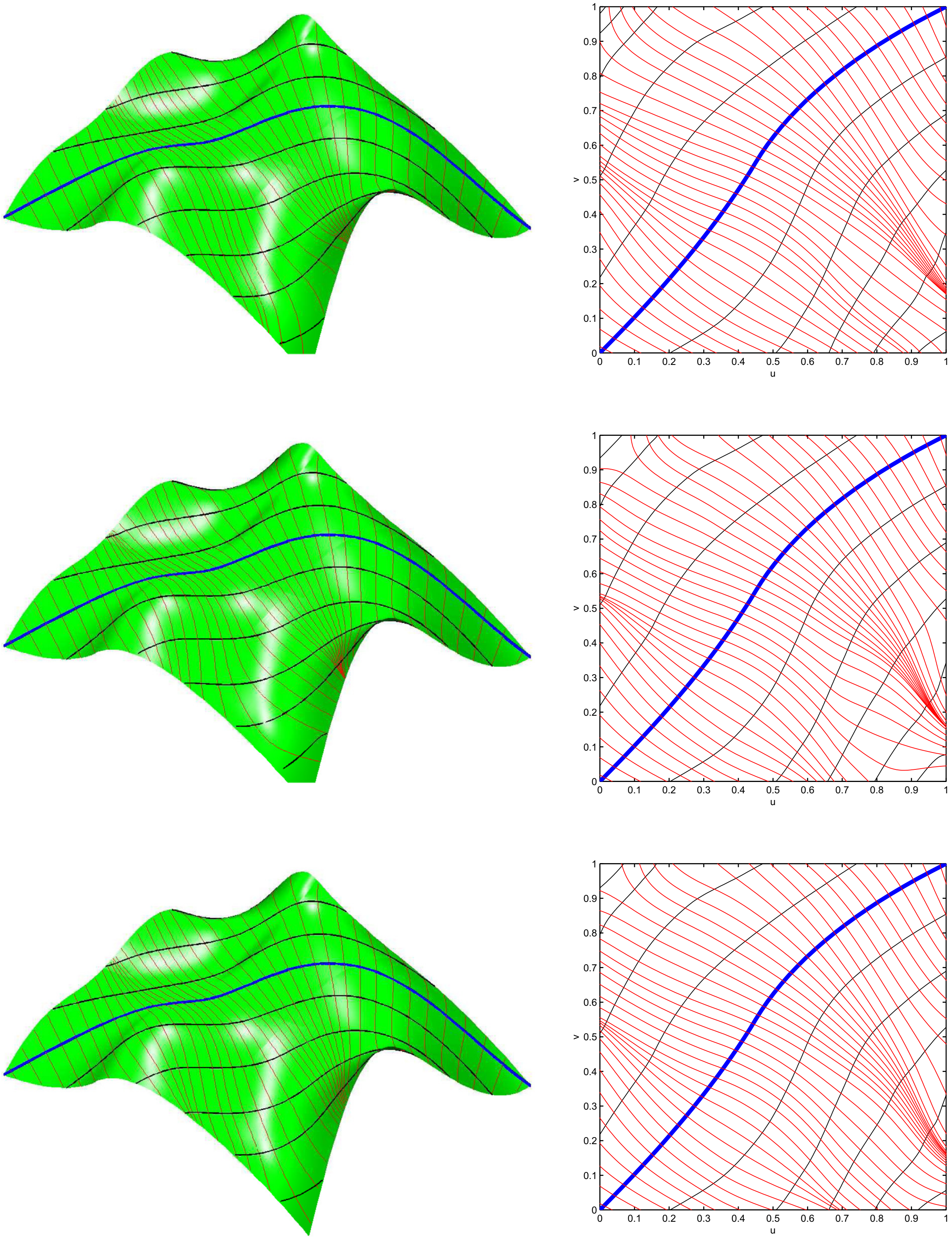

Fig. 7. Example of series of parallel curves to the generator: (left) on a NURBS surface; (right) on the surface parametric domain: (top) section parallels; (middle) vector-field parallels; (bottom) geodesic parallels.

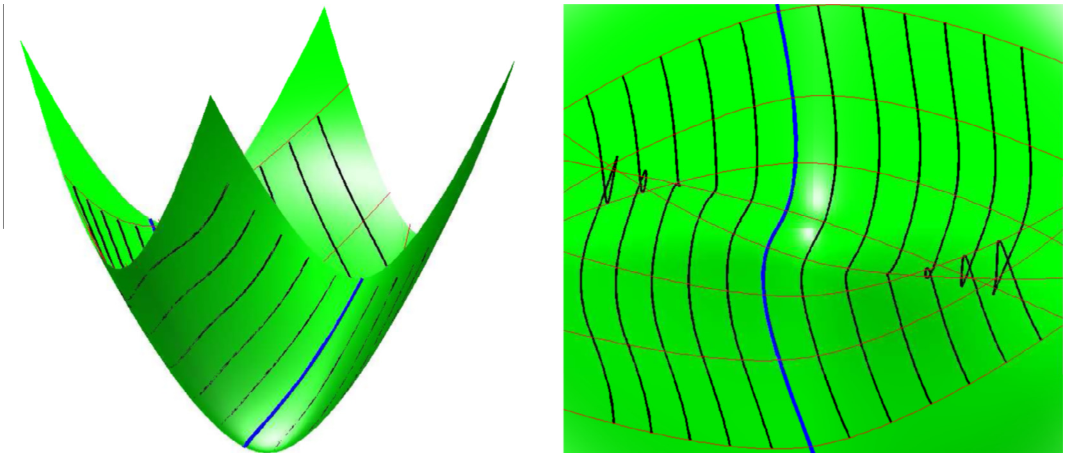

Fig. 8. Two views of a NURBS surface with swallowtail solutions for section parallel curves.

surface, $\mathbf{C}(t^*)$, will lie very close to the singularity itself. In case we need to compute the cusp precisely, a two-pass iterative method can be used. In the first stage, a point close to the singularity is determined; then, an optimization algorithm (such as the steepest descent method) is applied in order to minimize the function $f(t) = (\mathbf{C}(t+h) - \mathbf{C}(t))^2$ for a small displacement $h \in \mathbb{R}$, where $\mathbf{C}(t)$ denotes the parallel curve on the surface.

6. Conclusions and future work

In this paper a general methodology to compute parallel curves on parametric surfaces is introduced. Our approach is based on geometric-differential arguments leading to initial-value problems of systems of ODEs that are numerically integrated. To illustrate our method, the interesting cases of section, vector-field and geodesic parallels are formulated and solved. Several examples showing the good performance of our proposal as well as some computational issues are also briefly discussed. The methodology presented here is very general, as it is valid for any type of differentiable surface, regardless it is polynomial, rational or other. In addition, the method can be applied to any kind of characteristic curves on surfaces. We only need to obtain the corresponding ODEs characterizing the new curves and then proceed in a similar way.

Parallel curves can be seen as a more general concept than the offset of curves on a surface, which are defined as the locus of points that are at constant distance from the generator along the surface normal vector [37,38]. Offset curves are receiving increasing attention during the last years because of their applications to various fields, including manufacturing, robotics, brush stroke representation, feature recognition and many others [6–9,39,40]. However, the offset of a curve on a surface is no longer on that surface. On the contrary, parallel curves do preserve the distance from the generator while still lying on the surface. On the other hand, computation of offsets has revealed to be a very hard problem [7,38,41] while computation of parallel curves can be performed in a straightforward manner by following our approach.

Section, vector-field and geodesic curves (and others) have already been mentioned as useful tools for different processes in manufacturing, particularly in CNC-machining [6,13,37,14,8,9]. These references and others mentioned in our introductory section clearly indicate the usefulness of this contribution to expand the range of techniques for defining tool-paths for CNC-machining and for dealing with other related problems.

Regarding our future work, we plan to extend our approach to the case of implicit surfaces and to other families of characteristic curves on surfaces. We also wish to investigate further in order to determine the range of applicability of those parallel curves to industrial environments and other fields. Finally, a thorough comparison among different families of parallel curves will also be the subject of our future work.

Acknowledgements

This research has been supported by the Computer Science National Program of the Spanish Ministry of Education and Science, Project TIN2012–30768, the University of Cantabria, and Toho University. Special thanks are owe to the Editor and the anonymous reviewers for their encouraging comments and very helpful feedback that allowed us to improve our paper significantly. The authors are particularly grateful to the Department of Information Science of Toho University for all the facilities given to carry out this research work.

Appendix A. NURBS surfaces

Let $\mathcal{U} = \{a = u_0, u_1, u_2, \ldots, u_{r-1}, u_r = b\}$ be a nondecreasing sequence of nonnegative real numbers called *knots*. $\mathcal{U}$ is called the *knot vector*. The *ith B-spline basis function $N_{i,p}(u)$ of degree p* (*or order* $p+1$) is defined by the recurrence relations

$$N_{i,0}(u)=\begin{cases}1 & \text{if } u_i \leqslant u < u_{i+1},\\ 0 & \text{otherwise},\end{cases} \tag{A.1}$$

with $i=0,\dots,r-1$ and

$$N_{i,p}(u)=\frac{u-u_i}{u_{i+p}-u_i}N_{i,p-1}(u)+\frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1}(u) \tag{A.2}$$

for $i=0,\dots,r-p-1$, with $p>0$. In this definition, knots u_i and u_{i+1} must not necessarily be different. In fact, the most common case in industry is the so-called non-periodic knot vector. It consists of repeating the end knots as many times as the order, i.e., $u_0=u_1=\cdots=u_p=a, u_{r-p}=u_{r-p+1}=\cdots=u_r=b$. When necessary, the convention $\frac{0}{0}=0$ in Eq. (A.2) is applied. Without loss of generality, we can assume that $[a,b]=[0,1]$.

Given a set of 3D points called *control points*, $\{\mathbf{Q}_{ij}\}_{i=0,\dots,M;j=0,\dots,N}$, in a bidirectional net and two knot vectors $\mathcal{U}=\{u_0,u_1,u_2,\dots,u_{r-1},u_r\}$ and $\mathcal{V}=\{v_0,v_1,\dots,v_{h-1},v_h\}$, a *NURBS surface* $\mathbf{S}(u,v)$ *of order* (k,l) is a piecewise rational parametric surface given by:

$$\mathbf{S}(u,v)=\frac{\sum_{i=0}^{M}\sum_{j=0}^{N}w_{i,j}\mathbf{Q}_{i,j}N_{i,p}(u)N_{j,q}(v)}{\sum_{i=0}^{M}\sum_{j=0}^{N}w_{i,j}N_{i,p}(u)N_{j,q}(v)}, \tag{A.3}$$

where the $\{N_{i,p}(u)\}_i$ and $\{N_{j,q}(v)\}_j$ are the B-spline basis functions of degree p and q, respectively, defined recursively following (A.1) and (A.2), and $\{w_{i,j}\}_{i,j}$ are weights associated with the control points $\{\mathbf{Q}_{i,j}\}_{i,j}$. For a proper definition in Eq. (A.3), the following relationships must hold: $r=p+M+1, h=q+N+1$ (see [36] for further details).

Appendix B. Derivatives of NURBS surfaces

The kth derivative of a B-spline basis function is given by:

$$N_{i,p}^{(k)}(u)=p\left(\frac{N_{i,p-1}^{(k-1)}(u)}{u_{i+p}-u_i}-\frac{N_{i+1,p-1}^{(k-1)}(u)}{u_{i+p+1}-u_{i+1}}\right). \tag{B.1}$$

To compute the partial derivatives of a NURBS surface, we consider the numerator and denominator of (A.3), denoted as $\mathbf{N}(u,v)$ and $D(u,v)$, respectively. The partial derivatives of either can be obtained by computing the derivatives of the B-spline basis functions according to (B.1). For instance:

$$\frac{\partial^{k+l}\mathbf{N}(u,v)}{\partial^k u\partial^l v}=\sum_{i=0}^{M}\sum_{j=0}^{N}\mathbf{P}_{ij}N_{i,p}^{(k)}(u)N_{j,q}^{(l)}(v) \tag{B.2}$$

and a similar expression is obtained for $D(u,v)$. Now, the partial derivative (k,l) of the NURBS surface can be obtained as:

$$\begin{aligned}\frac{\partial^{k+l}\mathbf{S}(u,v)}{\partial^k u\partial^l v}=\frac{1}{D(u,v)}\Bigg(&\frac{\partial^{k+l}\mathbf{N}(u,v)}{\partial^k u\partial^l v}-\sum_{i=1}^{k}\binom{k}{i}\frac{\partial^i\mathbf{D}(u,v)}{\partial^i u}\frac{\partial^{k+l-i}\mathbf{S}(u,v)}{\partial^{k-i}u\partial^l v}-\sum_{j=1}^{l}\binom{l}{j}\frac{\partial^j\mathbf{D}(u,v)}{\partial^j v}\frac{\partial^{k+l-j}\mathbf{S}(u,v)}{\partial^k u\partial^{l-j}v}\\ &-\sum_{i=1}^{k}\binom{k}{i}\sum_{j=1}^{l}\binom{l}{j}\frac{\partial^{i+j}\mathbf{D}(u,v)}{\partial^i u\partial^j v}\frac{\partial^{k+l-i-j}\mathbf{S}(u,v)}{\partial^{k-i}u\partial^{l-j}v}\Bigg).\end{aligned} \tag{B.3}$$

Appendix C. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.apm.2013.10.042.

References

[1] N. Amenta, S. Choi, R.K. Kolluri, The power crust, in: Proceedings of the 6th ACM Symposium on Solid Modeling, SM'2001, ACM, New York, 2011, pp. 249–260.

[2] N. Amenta, S. Choi, R.K. Kolluri, The power crust, unions of balls, and the medial axis transform, Comput. Geom. 19 (2–3) (2001) 127–153.

[3] G. Brunnett, Geometric modeling of parallel curves on surfaces, Comput. Suppl. 14 (1999) 37–53.

[4] N.M. Patrikalakis, L. Bardis, Offsets of curves on rational B-spline surfaces, Eng. Comput. 5 (1989) 39–46.

[5] S. Marshall, J.G. Griffiths, A new cutter-path topology for milling machines, Comput. Aided Des. 26 (3) (1994) 204–214.

[6] B.K. Choi, R.B. Jerard, Sculptured Surface Machining. Theory and Applications, Kluwer Academic Publishers, Dordrecht, Boston, London, 1998.

[7] N.M. Patrikalakis, T. Maekawa, Shape Interrogation for Computer Aided Design and Manufacturing, Springer Verlag, 2002.

[8] R. Sarma, D. Dutta, The geometry and generation of NC tool paths, J. Mech. Des.: ASME Trans. 119 (1997) 253–258.

[9] K. Suresh, D.C.H. Yang, Constant scallop-height machining of free-form surfaces, J. Eng. Ind.: ASME Trans. 116 (1996) 253–259.

[10] H.Y. Feng, H. Li, Constant scallop-height tool path generation for three-axis sculptured surface machining, Comput. Aided Des. 34 (9) (2002) 647–654.

[11] J.H. Yoon, Fast tool path generation by the iso-scallop height method for ball-end milling of sculptured surfaces, Int. J. Prod. Res. 43 (23) (2005) 4989–4998.

[12] J.H. Yoon, Two-dimensional representation of machining geometry and tool path generation for ball-end milling of sculptured surfaces, Int. J. Prod. Res. 45 (14) (2007) 3151–3164.
[13] T.J. Kim, Constant cusp height tool paths as geodesic parallels on an abstract Riemannian manifold, Comput. Aided Des. 39 (6) (2007) 477–489.
[14] K. Marciniak, Geometric Modeling for Numerically Controlled Machining, Oxford University Press, New York, 1991.
[15] The Mathworks Inc: using Matlab, Natick, MA, 1999.
[16] The netlib repository, Oak Ridge Lab, US Department of Energy. <http://www.netlib.org>.
[17] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes, second ed., Cambridge University Press, Cambridge, 1992.
[18] D.J. Struik, Lectures on Classical Differential Geometry, second ed., Dover Publications, New York, 1988.
[19] R. Klass, Correction of local surface irregularities using reflection lines, Comput. Aided Des. 12 (2) (1980) 73–77.
[20] K.P. Beier, Y.F. Chen, Highlight-line algorithm for realtime surface quality assessment, Comput. Aided Des. 26 (4) (1994) 268–277.
[21] T. Poeschl, Detecting surface irregularities using isophotes, Comput. Aided Geom. Des. 1 (2) (1984) 163–168.
[22] J.M. Beck, R.T. Farouki, J.K. Hinds, Surface analysis methods, IEEE Comput. Graphics Appl. (1986) 18–36.
[23] R.T. Farouki, Graphical methods for surface differential geometry, in: R. Martin (Ed.), Mathematics of Surfaces, IMA Series, 1987, pp. 363–385.
[24] J. Puig-Pey, A. Gálvez, A. Iglesias, Helical curves on surfaces for computer-aided geometric design and manufacturing, Lecture Notes Comput. Sci. 3044 (2004) 771–778.
[25] H. Theisel, G.E. Farin, The curvature of characteristic curves on surfaces, IEEE Comput. Graphics Appl. 17 (6) (1997) 88–96.
[26] J. Puig-Pey, A. Gálvez, A. Iglesias, J. Rodríguez, P. Corcuera, F. Gutiérrez, Some applications of scalar and vector fields to geometric processing of surfaces, Comput. Graphics 29 (5) (2005) 723–729.
[27] H. Hagen, T. Schreiber, E. Gschwind, Methods for surface interrogation, in: Proc. Visualization'90, IEEE Computer Society Press, Los Alamitos, CA, 1990, pp. 187–193.
[28] H. Hagen, S. Hahman, T. Schreiber, Y. Nakajima, B. Wördenweber, P. Hollemann-Grundstedt, Surface interrogation algorithms, IEEE Comput. Graphics Appl. (1992) 53–60.
[29] S. Hahmann, Visualization techniques for surface analysis, in: C. Bajaj (Ed.), Advanced Visualization Technique, John Wiley and Sons, New York, 1999, pp. 53–60.
[30] J. Sánchez-Reyes, R. Dorado, Constrained design of polynomial surfaces from geodesic curves, Comput. Aided Des. 40 (2008) 49–55.
[31] G. Wang, K. Tang, C. Tai, Parametric representation of a surface pencil with a common spatial geodesic, Comput. Aided Des. 36 (2004) 447–459.
[32] R.J. Haw, An application of geodesic curves to sail design, Comput. Graphics Forum 4 (1985) 137–139.
[33] A. Iglesias, A. Gálvez, J. Puig-Pey, Generating drop trajectories on parametric surfaces, in: Q. Peng, W. Li, J. Yu (Eds.), Proc. of CAD/Graphics'2001, International Academic Publ./World Publ. Corp., Beijing, 2001, pp. 350–357.
[34] A. Gálvez, J. Puig-Pey, A. Iglesias, A differential method for parametric surface intersection, Lecture Notes Comput. Sci. 3044 (2004) 651–660.
[35] J. Puig-Pey, A. Gálvez, A. Iglesias, A new differential approach for parametric-implicit surface intersection, Lecture Notes Comput. Sci. 2657 (2003) 897–906.
[36] L. Piegl, W. Tiller, The NURBS Book, Springer Verlag, Berlin, Heidelberg, 1997.
[37] T. Maekawa, An overview of offset curves and surfaces, Comput. Aided Des. 31 (1999) 165–173.
[38] B. Pham, Offset curves and surfaces: a brief survey, Comput. Aided Des. 24 (1992) 223–229.
[39] R.E. Barnhill, Geometric Processing for Design and Manufacturing, SIAM, Philadelphia, 1992.
[40] A. Iglesias, A. Gálvez, J. Puig-Pey, Computational methods for geometric processing. Applications to Industry, Lecture Notes Comput. Sci. 2073 (2001) 698–707.
[41] L. Piegl, W. Tiller, Computing offsets of NURBS curves and surfaces, Comput. Aided Des. 31 (1999) 147–156.