Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Computer Aided Geometric Design 24 (2007) 189–199
COMPUTER AIDED GEOMETRIC DESIGN
www.elsevier.com/locate/cagd

B-spline control nets for developable surfaces

L. Fernández-Jambrina

E.T.S.I. Navales, Universidad Politécnica de Madrid, Arco de la Victoria s/n, E-28040-Madrid, Spain

Received 19 December 2005; accepted 4 March 2007

Available online 12 March 2007

Abstract

In this paper a linear algorithm is derived for constructing B-spline control nets for spline developable surfaces of arbitrary degree and number of pieces. Control vertices are written in terms of five free parameters related to the type of developable surface. Aumann's algorithm for constructing Bézier developable surfaces is recovered as a particular case.

Keywords: B-spline surfaces; Developable surfaces

1. Introduction

Developable surfaces are an important family of ruled surfaces for industrial design. Since their total curvature is zero, they model the behavior of steel sheets or pieces of cloth which are deformed just by folding, rolling, cutting, but without stretching. Therefore they are much appreciated, at least as an approximation, in production related with steel, such as naval industry, and also in textile industry.

Qualitatively they are easy to design, since their geometrical characterization in terms of the tangent plane to the rulings of the surface is easy (cfr. for instance (Postnikov, 1979) or (Struik, 1961)). However, they are somehow at odds with the nowadays most common standard for geometric design, NURBS surfaces, since the equations for the control points are non-linear and can be solved exactly only for low degrees.

There have been several approaches to cope with developable surfaces within the NURBS framework, most of them successful to a certain extent, since the seminal paper by Frey and Bindschadler (1993). In (Lang and Röschel, 1992) the conditions for Bézier developable surfaces of low degree are solved. The algorithm based on the de Casteljau algorithm in (Chu and Séquin, 2002) is also useful for low degrees. In (Aumann, 1991) the conditions for interpolating with developable Bézier surfaces are stated.

Resorting to the dual geometry in which points are exchanged with planes (Pottmann and Farin, 1995) simplify the problem of constructing rational Bézier developable surfaces. A first approach to these projective methods may be found in (Bodduluri and Ravani, 1993). These techniques are used in (Pottmann and Wallner, 1999) for approximating developable surfaces.

Approximately developable surfaces useful for industry are shown in (Chalfant and Maekawa, 1998). A method for approximating developable surfaces by spline cones is shown in (Leopoldseder, 2001).

E-mail address: leonardo.fernandez@upm.es.

0167-8396/$ – see front matter
doi:10.1016/j.cagd.2007.03.001

As it is well known, Postnikov (1979), Struik (1961), developable surfaces may be classified into planar, cylindrical, conical and tangent surfaces. Whereas the first three types are easy to implement as Bézier and B-spline surfaces, the latter one is a bit more complex. An algorithm for producing general Bézier developable patches is shown in (Aumann, 2003). This algorithm, based on affine applications of the cells of the control net of the patch is further developed in (Aumann, 2004), where degree elevation is used for modifying the boundary of a patch of a Bézier developable surface, but without altering the surface itself. It would therefore be interesting to know whether developable surfaces are tractable in more general frameworks.

The paper is organized as follows. Next section is devoted to general features of developable surfaces related to the Bézier and B-spline cases, including the classification of developable surfaces. Section 3 shows a way to state the developability condition in terms of blossoms of B-spline curves. This condition is used to rederive in a simple fashion Aumann's algorithm, (Aumann, 2003), and also to derive a linear algorithm for constructing generic B-spline developable surfaces. Applications of this algorithm are shown in Section 4.

2. Developable surfaces

Developable surfaces are a special case of ruled surfaces with null Gaussian curvature. This means that they are intrinsically flat. These are regions of planes that have been folded, rolled or pasted forming cylinders, cones... in three-dimensional space, but without stretching them. They can therefore be extended back onto the plane without stretching them, at most by performing some cuts.

If a surface is parametrized by $c(u,v)$, with a normal vector $n(u,v)$, the null Gaussian curvature condition may be expressed as

$$0 = \begin{vmatrix} c_{uu}\cdot n & c_{uv}\cdot n \\ c_{uv}\cdot n & c_{vv}\cdot n \end{vmatrix}_{(u,v)}, \tag{1}$$

and in the case of a ruled surface generated by two curves, $c(u)$ and $d(u)$, $c(u,v) = (1-v)c(u) + vd(u)$, the condition becomes much simpler,

$$c_{uv}(u,v)\cdot c_u(u,v)\times c_v(u,v) = 0. \tag{2}$$

By either applying it to the curve $c(u)$, $v=0$, or to $d(u)$, $v=1$, the condition is further simplified,

$$d'(u)\cdot c'(u)\times \bigl(d(u)-c(u)\bigr) = 0, \tag{3}$$

and has a direct geometrical meaning, since it states that the generators of the curves at u, $c'(u)$, $d'(u)$ and the vector that links the points $c(u)$, $d(u)$ are coplanary for a developable surface, as it is shown in Fig. 1.

Another way of looking at it arises by taking into account that if $c'(u)$, $d'(u)$, $d(u)-c(u)$ lie on the same plane for u, then $c_u(u,v)$ and $c_v(u,v)$ generate the same plane for u for all v. Therefore the tangent plane to the surface is the same for all points on the same line of the ruling and developable surfaces may be viewed as envelopes of uniparametric families of planes.

Developable surfaces can be classified into several families. Denoting the direction of the line of the ruling at u by $\mathbf{c}(u) := d(u)-c(u)$, $c(u,v) = c(u) + v\mathbf{c}(u)$, the null Gaussian curvature condition may be restated as

$$0 = \mathbf{c}'(u)\cdot c'(u)\times \mathbf{c}(u).$$

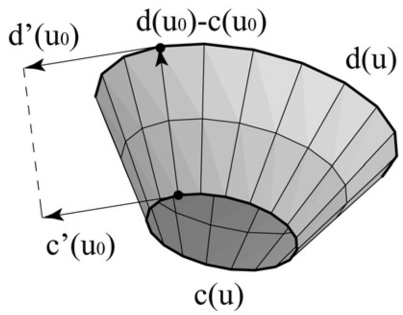

Fig. 1. Developable surface.

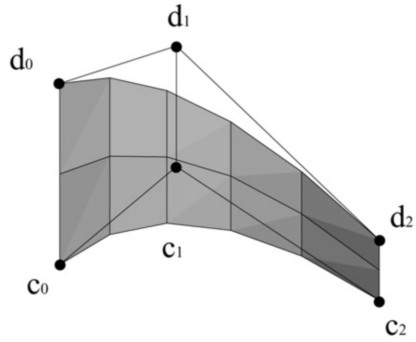

Fig. 2. Cylindrical Bézier surface.

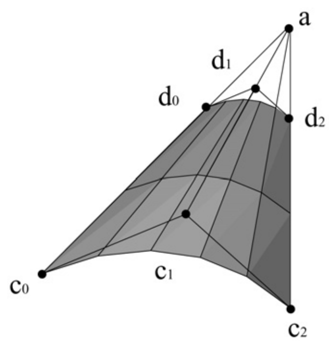

Fig. 3. Conical Bézier surface.

Several possibilities arise from this condition:

(1) Planar surfaces: If all vectors lie on the same constant plane, that is, if $c(u)$ is a plane curve and $\mathbf{c}(u)$ is a vector on the same plane for all u, we are depicting a region of the plane.
Representing planar surfaces is easy, since we just need a control net confined into a plane.

(2) Cylindrical surfaces: If $\mathbf{c}'(u) \parallel \mathbf{c}(u)$, the vector $\mathbf{c}(u)$ is always parallel to a constant direction $\mathbf{v}$,

$$d(u) = c(u) + \lambda(u)\mathbf{v}, \qquad c(u,v) = c(u) + \lambda(u)v\mathbf{v}, \tag{4}$$

that is, all the lines of the ruling are parallel.
This provides an easy way to construct Bézier or spline cylindrical surfaces. If $\{c_0,\dots,c_L\}$, $\{d_0,\dots,d_L\}$ are the control polygons of the curves $c(u)$, $d(u)$, we just require that the vectors $\mathbf{c_0d_0},\dots,\mathbf{c_Ld_L}$ be all parallel to a vector $\mathbf{v}$ in order to have a cylindrical surface, as it is shown in Fig. 2.

(3) Conical surfaces: $c'(u) = k\mathbf{c}'(u)$, that is $c(u) = a + k\mathbf{c}(u)$ and $d(u) = a + (k+1)\mathbf{c}(u)$, where a is a fixed point, the vertex of the cone, and every line of the ruling passes through it,

$$c(u,v) = a + (v+k)\mathbf{c}(u). \tag{5}$$

Since $c(u)$ and $d(u)$ are the same curve but for a translation and a change of scale, the respective sides of their control polygons, if they are Bézier or B-spline curves, must be parallel and proportional, as it is shown in Fig. 3,

$$\alpha\mathbf{c_ic_{i-1}} = \mathbf{d_id_{i-1}}, \quad i = 1,\dots,L,\ \alpha = 1 + 1/k. \tag{6}$$

(4) Tangent surfaces: $\mathbf{c}(u) \parallel c'(u)$: The line of the ruling at $c(u)$ is tangent to the curve at $c(u)$. The surface is therefore spanned by segments of the tangent lines to the curve $c(u)$. An example is shown in Fig. 4.

$$c(u,v) = c(u) + v\lambda(u)c'(u). \tag{7}$$

(5) Generic case: $c'(u)$ is a linear combination of $\mathbf{c}(u)$, $\mathbf{c}'(u)$ for every u,

$$c'(u) = \lambda(u)\mathbf{c}(u) + \mu(u)\mathbf{c}'(u).$$

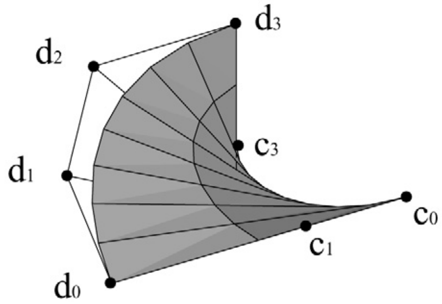

Fig. 4. Bézier tangent surface.

However, if we move $c(u)$ along the lines of the ruling, $\tilde{c}(u) = c(u) - \mu(u)\mathbf{c}(u)$,

$$\tilde{c}'(u) = \tilde{\lambda}(u)\mathbf{c}(u), \qquad \tilde{\lambda}(u) = \lambda(u) - \mu'(u),$$

we get a tangent surface if $\tilde{\lambda}(u) \not\equiv 0$ or a conical surface if $\tilde{\lambda}(u) \equiv 0$.

Therefore, all developable surfaces may be classified into planar, cylindrical, conical or tangent surfaces and the latter is the generic case.

Tangent surfaces expressed in their canonical form of (7) are readily expressed as Bézier or B-spline surfaces. For instance, when $\lambda = 1$ and the curve $c(u)$ is given by its control polygon $\{c_0, \dots, c_n\}$, $c'(u)$ is a vector curve with polygon $\{n\Delta c_0, \dots, n\Delta c_{n-1}\}$ and degree $n-1$, which must be degree-raised by one,

$$n\Delta c_i^1 = (n-i)\Delta c_i + i\Delta c_{i-1} = (n-i)c_{i+1} + (2i-n)c_i - ic_{i-1}$$

in order to write the control net of the tangent surface,

$$\{c_0, c_0 + n\Delta c_0^1; \dots; c_n, c_n + n\Delta c_n^1\}.$$

However, this is a particular solution and the representation of generic developable surfaces is open.

An interesting result is shown in (Aumann, 2003). A Bézier ruled surface defined by two curves $c(u)$ and $d(u)$ with control polygons $\{c_0, \dots, c_n\}$ and $\{d_0, \dots, d_n\}$ is developable if the cells of the control net are related by affine transformations, that is, the barycentric combinations of the vertices of the cells of the control net are the same for all cells,

$$d_i = (1-\lambda-\mu)c_{i-1} + \lambda c_i + \mu d_{i-1}, \quad \lambda, \mu \in \mathbb{R},\ i = 1, \dots, n-1.$$

This result is generic (Aumann, 2004) in the sense that it comprises all non-planar, non-cylindrical Bézier developable surfaces. It would then be interesting to derive another algorithm that would work for B-spline developable surfaces with the same generality of Aumann's result, since B-spline curves are more powerful than Bézier curves for design.

One way to accomplish this would be to cast spline curves into piecewise Bézier form, apply Aumann's algorithm to each of the pieces and then turn them back into B-spline form. For instance, for quadratic B-spline developable surface it may be checked that

$$\begin{aligned} d_i &= (1-\lambda_i-\mu_i)c_{i-1} + \lambda_i c_i + \mu_i d_{i-1}, \quad i = 1, \dots, L, \\ \lambda_i &= \frac{\mu_{i-1}(u_{i-1}-u_{i-2}) + \lambda_{i-1}(u_i - u_{i-2}) - u_{i-1} + u_{i-2}}{u_i - u_{i-1} + \mu_{i-1}(u_{i-1} - u_{i-2})}, \\ \mu_i &= \frac{u_i - u_{i+1} + \mu_{i-1}(u_{i+1} - u_{i-2})}{u_i - u_{i-1} + \mu_{i-1}(u_{i-1} - u_{i-2})}, \quad i = 2, \dots, L, \end{aligned} \tag{8}$$

for B-spline control polygons $\{c_0, \dots, c_L\}$, $\{d_0, \dots, d_L\}$ and sequence of knots $\{u_0, \dots, u_{L+1}\}$ for the spline curves $c(u)$ and $d(u)$ respectively.

This could be done analytically in principle for low degrees, but the results become cumbersome and difficult to generalize to all degrees.

A different approach could be grounded on the fact that Bézier curves are just a special case of B-spline curves and making use of blossoms of curves.

3. Algorithm for B-spline developable surfaces

As it has been pointed out, developable surfaces spanned by two parametrized curves, $c(u)$, $d(u)$, may be characterized by the fact that vectors $c'(u)$, $d'(u)$, $d(u)-c(u)$ must lie on a plane for every value of the parameter u.

Let us assume in order to avoid unnecessary complications in notation that both curves are B-splines of just one segment and degree n, defined by their B-spline control polygons $\{c_0,\dots,c_n\}$, $\{d_0,\dots,d_n\}$ and the same sequence of knots $\{u_0,\dots,u_{2n-1}\}$. Since the derivative of a B-spline curve of degree n at a value u in the knot interval $[u_{n-1},u_n]$ may be written as

$$\frac{\mathrm{d}c(u)}{\mathrm{d}u}=\frac{n}{u_n-u_{n-1}}\big(c_1^{n-1)}(u)-c_0^{n-1)}(u)\big),$$

where $c_0^{n-1)}(u)$ and $c_1^{n-1)}(u)$ are the last but one points obtained by application of the de Boor algorithm (de Boor, 1972; Farin, 2001) for calculating $c(u)$,

$$c_i^{1)}[v_1]:=c[u_{i+1},\dots,u_{i+n-1},v_1]=\frac{u_{i+n}-v_1}{u_{i+n}-u_i}c_i+\frac{v_1-u_i}{u_{i+n}-u_i}c_{i+1},\quad i=0,\dots,n-1,$$

$$c_i^{r)}[v_1,\dots,v_r]:=c[u_{i+r},\dots,u_{i+n-1},v_1,\dots,v_r]$$
$$=\frac{u_{i+n}-v_r}{u_{i+n}-u_{i+r-1}}c_i^{r-1)}[v_1,\dots,v_{r-1}]+\frac{v_r-u_{i+r-1}}{u_{i+n}-u_{i+r-1}}c_{i+1}^{r-1)}[v_1,\dots,v_{r-1}],$$
$$i=0,\dots,n-r,\ r=1,\dots,n,$$

$$c[v_1,\dots,v_n]:=c_0^{n)}[v_1,\dots,v_n]=\frac{u_n-v_n}{u_n-u_{n-1}}c_0^{n-1)}[v_1,\dots,v_{n-1}]+\frac{v_n-u_{n-1}}{u_n-u_{n-1}}c_1^{n-1)}[v_1,\dots,v_{n-1}],$$

$$c(u)=c[u,\dots,u]:=c[u^{\langle n\rangle}],$$

hence the condition of coplanarity for the vectors $c'(u)$, $d'(u)$, $d(u)-c(u)$ is readily translated into a condition of coplanarity for the four intermediate points $c_0^{n-1)}(u)$, $c_1^{n-1)}(u)$, $d_0^{n-1)}(u)$, $d_1^{n-1)}(u)$, as it is shown in Fig. 5.

This of course means that we may write one of the points as a barycentric combination of the other three,

$$d_1^{n-1)}(u)=(1-\lambda-\mu)c_0^{n-1)}(u)+\lambda c_1^{n-1)}(u)+\mu d_0^{n-1)}(u),$$

where the coefficients λ, μ may in principle be functions of u. However, we focus on constant λ, μ, since, as it is shown in (Aumann, 2004) for the Bézier case, the other cases are attained by degree-elevating the constant case.

The coplanarity condition may be expressed in a more symmetric and suitable form for our purposes,

$$(1-\Lambda)c_0^{n-1)}(u)+\Lambda c_1^{n-1)}(u)=(1-M)d_0^{n-1)}(u)+Md_1^{n-1)}(u),$$
$$\Lambda:=\frac{\lambda}{1-\mu},\qquad M:=\frac{1}{1-\mu}.$$

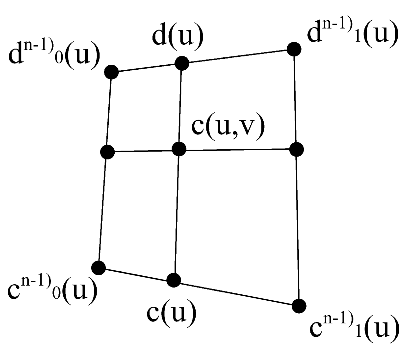

Fig. 5. Coplanarity condition for developability.

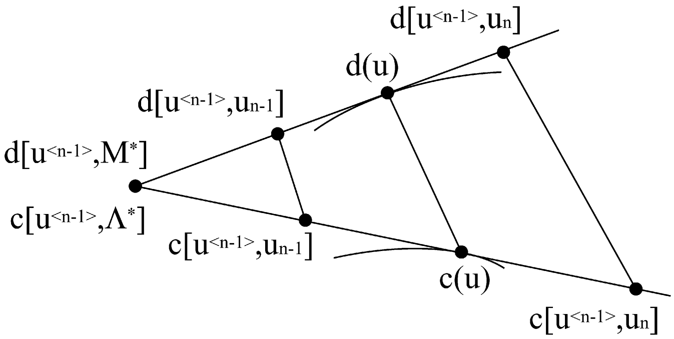

Fig. 6. Interpretation of the developability condition.

This form, however, excludes conical surfaces, since for them $\mu = 1$. On the contrary, cylinders are included, since for them $\lambda = 1$ and then $\Lambda = M$.

Taking into account the definition of the points,

$$c_0^{n-1)}(u) = c[u^{\langle n-1\rangle}, u_{n-1}], \qquad c_1^{n-1)}(u) = c[u^{\langle n-1\rangle}, u_n],$$

we may write down the coplanarity condition in a rather compact form in terms of blossoms instead of parametrizations,

$$c\big[u^{\langle n-1\rangle}, (1-\Lambda)u_{n-1} + \Lambda u_n\big] = d\big[u^{\langle n-1\rangle}, (1-M)u_{n-1} + M u_n\big], \tag{9}$$

since blossoms are multiaffine functions.

This result is rather powerful, since it states that two B-spline curves $c(u)$, $d(u)$ define a developable surface if they have the same blossom when evaluated once respectively on $\Lambda^* = (1-\Lambda)u_{n-1} + \Lambda u_n$, $M^* = (1-M)u_{n-1} + M u_n$.

Another way of looking at it is stating that both curves share the same curve of degree $(n-1)$ defined by (9). Hence, the developability condition implies n conditions on the B-spline polygons of the curves.

We may look at this condition in another way. Since $c[u^{\langle n-1\rangle}, v]$ defines the tangent line to the curve at $c(u)$ parametrized by v, condition (9) merely states that the tangent lines to the curves at $c(u)$ and $d(u)$ intersect at one point $c[u^{\langle n-1\rangle}, \Lambda^*] = d[u^{\langle n-1\rangle}, M^*]$, as it is shown in Fig. 6. This result is obvious since the developability condition implies that both tangent lines lie on the same plane for each value of the parameter u. This explains why the algorithm does not work for conical surfaces (and translational cylindrical surfaces either), since in this case the tangent lines to the curves at $c(u)$ and $d(u)$ are parallel.

If we consider B-spline surfaces with several segments, each segment has its own Λ and M. However, since there are sets of parameters $\{u_1, \dots, u_{n-1}\}$ for which the blossom may be calculated in more than one segment, the different values of Λ and M for each segment should provide globally the same values of Λ^* and M^*, which is therefore common for all segments.

For instance, denoting by Λ_k, M_k the coefficients Λ, M for the segment k,

$$(1-\Lambda_j)u_{n+j-2} + \Lambda_j u_{n+j-1} = \Lambda^* = (1-\Lambda_k)u_{n+k-2} + \Lambda_k u_{n+k-1},$$
$$(1-M_j)u_{n+j-2} + M_j u_{n+j-1} = M^* = (1-M_k)u_{n+k-2} + M_k u_{n+k-1},$$

$$\Lambda_k = \frac{\Delta u_{n+j-2}\Lambda_j + u_{n+j-2} - u_{n+k-2}}{\Delta u_{n+k-2}} = \frac{\Delta u_{n+k-3}}{\Delta u_{n+k-2}}(\Lambda_{k-1} - 1),$$
$$M_k = \frac{\Delta u_{n+j-2}M_j + u_{n+j-2} - u_{n+k-2}}{\Delta u_{n+k-2}} = \frac{\Delta u_{n+k-3}}{\Delta u_{n+k-2}}(M_{k-1} - 1), \tag{10}$$

the expressions for Λ_k and M_k are defined recursively. To the same expression we arrive by matching several developable Bézier surfaces imposing that the final surface is at least C^1.

The expressions relating the coefficients λ, μ of each segment, which we denote by $\lambda_{(k)}$, $\mu_{(k)}$, are more complicated,

$$\mu_{(k)} = 1 + \frac{\Delta u_{n+k-2}}{\Delta u_{n+k-3}}\left(1 - \frac{1}{\mu_{(k-1)}}\right), \qquad \lambda_{(k)} = \frac{\lambda_{(k-1)} + \mu_{(k-1)} - 1}{\mu_{(k-1)}}. \tag{11}$$

It is worth mentioning that cylinders ($\lambda_{(k)} = 1$) can be matched just to cylinders, since if one coefficient $\lambda_{(k)}$ is equal to one, then the other coefficients are also one, regardless of the values of $\mu_{(k)}$.

And cones ($\mu_{(k)} = 1$) can be matched just to cones, since if one coefficient $\mu_{(k)}$ is equal to one, then the other coefficients are also one. All cones share the same value of the coefficient $\mu_{(k)}$ then.

Since the parametrization of a curve unambiguously defines its polar form, we may state the result in terms of blossoms of curves instead of parametrizations:

Theorem 1. *Two B-spline curves $c(u)$, $d(u)$ with the same knot sequence $\{u_0, \ldots, u_K\}$ define a generic developable surface if their polar forms are related by*

$$c[v_1, \ldots, v_{n-1}, \Lambda^*] = d[v_1, \ldots, v_{n-1}, M^*],$$

for some values Λ^*, M^*.

Of course, general planar surfaces fall out of this framework, but also conical surfaces, since they involve vector combinations,

$$\Lambda\big(c[v_1, \ldots, v_{n-1}, u_n] - c[v_1, \ldots, v_{n-1}, u_{n-1}]\big) \\ = d[v_1, \ldots, v_{n-1}, u_n] - d[v_1, \ldots, v_{n-1}, u_{n-1}],$$

but this is not a problem since conical surfaces are easy to implement. Cylindrical surfaces appear when $\Lambda = M$.

A simple consequence of this formulation is that Aumann's algorithm for Bézier developable surfaces is easily recovered. If the curves $c(u)$, $d(u)$ are parametrized over the interval [0, 1], that is, $u_{n-1} = 0$, $u_n = 1$, we get the general condition

$$c[v_1, \ldots, v_{n-1}, \Lambda] = d[v_1, \ldots, v_{n-1}, M],$$

which produces valuable information about the control net of the surface when restricted to sequences of parameters of the form $\{0^{\langle n-i-1\rangle}, 1^{\langle i\rangle}\}$,

$$(1-\Lambda)c[0^{\langle n-i\rangle}, 1^{\langle i\rangle}] + \Lambda c[0^{\langle n-i-1\rangle}, 1^{\langle i+1\rangle}] \\ = (1-M)d[0^{\langle n-i\rangle}, 1^{\langle i\rangle}] + Md[0^{\langle n-i-1\rangle}, 1^{\langle i+1\rangle}], \quad i = 0, \ldots, n-1,$$

which is easily written in terms of control polygon vertices,

$$(1-\Lambda)C_i + \Lambda C_{i+1} = (1-M)D_i + MD_{i+1}, \quad i = 0, \ldots, n-1,$$

which is the formulation of Aumann's algorithm for Bézier developable surfaces (Aumann, 2003):

Corollary 1. *Two Bézier curves $c(u)$, $d(u)$ with respective control polygons $\{C_0, \ldots, C_n\}$, $\{D_0, \ldots, D_n\}$ define a generic developable surface if the cells of the control net of the surface are planar and are defined according to the same barycentric combination,*

$$(1-\Lambda)C_i + \Lambda C_{i+1} = (1-M)D_i + MD_{i+1}, \quad i = 0, \ldots, n-1. \tag{12}$$

The case $\Lambda = M$ corresponds to cylindrical surfaces and conical surfaces require a proportionally condition between the sides of the control polygons,

$$\Lambda(C_{i+1} - C_i) = D_{i+1} - D_i, \quad i = 0, \ldots, n-1.$$

A similar reasoning may be applied for obtaining developability conditions in terms of B-spline control polygons, but using $(n-1)$ sequences of parameters $\{u_{i+1}, \ldots, u_{i+n-1}\}$ instead of sequences of ones and zeros. Since

$$c[u_{i+1}, \ldots, u_{i+n-1}, \Lambda^*] = \frac{u_{i+n} - \Lambda^*}{u_{i+n} - u_i} c[u_i, \ldots, u_{i+n-1}] + \frac{\Lambda^* - u_i}{u_{i+n} - u_i} c[u_{i+1}, \ldots, u_{i+n}] \\ = \frac{u_{i+n} - \Lambda^*}{u_{i+n} - u_i} c_i + \frac{\Lambda^* - u_i}{u_{i+n} - u_i} c_{i+1}, \quad i = 0, \ldots, L-1,$$

we obtain again a simple condition for B-spline surfaces,

$$\frac{u_{i+n}-\Lambda^*}{u_{i+n}-u_i}c_i+\frac{\Lambda^*-u_i}{u_{i+n}-u_i}c_{i+1}=\frac{u_{i+n}-M^*}{u_{i+n}-u_i}d_i+\frac{M^*-u_i}{u_{i+n}-u_i}d_{i+1},$$

for $i=0,\dots,L-1$.

Therefore the developability condition for B-spline surfaces may be summarized as follows:

Corollary 2. *Two B-spline curves $c(u)$, $d(u)$ with respective control polygons $\{c_0,\dots,c_L\}$, $\{d_0,\dots,d_L\}$ and knot sequence $\{u_0,\dots,u_K\}$ define a generic developable surface if the cells of the control net of the surface are planar and are defined according to*

$$\frac{u_{i+n}-\Lambda^*}{u_{i+n}-u_i}c_i+\frac{\Lambda^*-u_i}{u_{i+n}-u_i}c_{i+1}=\frac{u_{i+n}-M^*}{u_{i+n}-u_i}d_i+\frac{M^*-u_i}{u_{i+n}-u_i}d_{i+1}, \tag{13}$$

for $i=0,\dots,L-1$ and some values Λ^*, M^*.

The case $\Lambda=M$ corresponds to cylindrical surfaces and conical surfaces just require a proportionally condition between the sides of the control polygons,

$$\Lambda^*(c_{i+1}-c_i)=d_{i+1}-d_i,\quad i=0,\dots,L-1.$$

The explicit expression of each vertex d_N, $N=1,\dots,L$, in terms of Λ^*, M^*, d_0 and the control polygon of $c(u)$ is easily derived,

$$d_N=\prod_{i=0}^{N-1}\frac{M^*-u_{i+n}}{M^*-u_i}d_0+\frac{u_n-\Lambda^*}{M^*-u_0}\prod_{i=1}^{N-1}\frac{M^*-u_{i+n}}{M^*-u_i}c_0+\frac{\Lambda^*-u_{N-1}}{M^*-u_{N-1}}c_N$$
$$+\frac{M^*-\Lambda^*}{M^*-u_{N-1}}\sum_{i=1}^{N-1}\frac{u_{i+n}-u_{i-1}}{M^*-u_{i-1}}\left(\prod_{j=i}^{N-2}\frac{M^*-u_{n+j+1}}{M^*-u_j}\right)c_i. \tag{14}$$

The vertices can be conveniently sorted,

$$d_N-c_N=\prod_{i=0}^{N-1}\frac{M^*-u_{i+n}}{M^*-u_i}(d_0-c_0)+\frac{\Lambda^*-M^*}{M^*-u_{N-1}}\left\{c_N-\frac{M^*-u_{N-1}}{M^*-u_0}\prod_{i=1}^{N-1}\frac{M^*-u_{i+n}}{M^*-u_i}c_0\right.$$
$$\left.-\sum_{i=1}^{N-1}\frac{u_{i+n}-u_{i-1}}{M^*-u_{i-1}}\left(\prod_{j=i}^{N-2}\frac{M^*-u_{n+j+1}}{M^*-u_j}\right)c_i\right\}, \tag{15}$$

in order to factor the dependence on Λ^* in a (Λ^*-M^*) term. This fact shall be useful for interpolation problems. An example is shown in Fig. 7.

Although it is not directly useful for our purposes, the expression may be written in an even more compact form,

$$d_N-c_N=\prod_{i=0}^{N-1}\frac{M^*-u_{i+n}}{M^*-u_i}(d_0-c_0)+(\Lambda^*-M^*)\sum_{i=0}^{N-1}\left(\prod_{j=i+1}^{N-1}\frac{M^*-u_{n+j}}{M^*-u_j}\right)\frac{\Delta c_i}{M^*-u_i}, \tag{16}$$

in terms of differences of vertices, $\Delta c_i=c_{i+1}-c_i$.

These expressions allow us another interpretation of the parameters Λ^* and M^*: if we perform a restriction on the parameter v from the original interval $[0,1]$ to $[0,\beta]$, that is, preserving $c(u)$ as part of the boundary of the developable surface, denoting by $\tilde{\Lambda}^*$ and $\tilde{M}^*$ the parameters of the restricted surface and by $\{\tilde{d}_0,\dots,\tilde{d}_L\}$ the B-spline control polygon of the new boundary curve $\tilde{d}(u)=c(u,\beta)$,

$$\tilde{d}_i=c_i+\beta(d_i-c_i),\quad i=0,\dots,L,$$

we obtain an interesting relation between the parameters of both surfaces,

$$M^*=\tilde{M}^*,\qquad \tilde{\Lambda}^*=M^*+\beta(\Lambda^*-M^*), \tag{17}$$

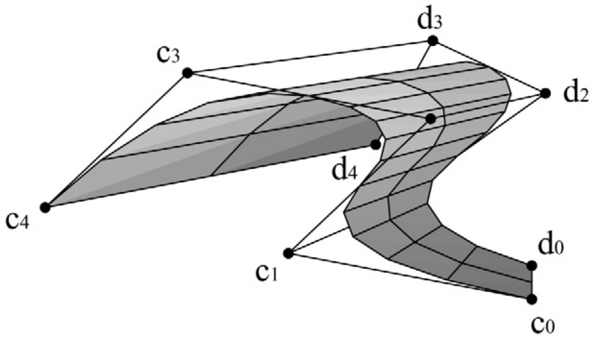

Fig. 7. Regular developable B-spline surface.

which states that M^* defines the developable surface as a whole with boundary on $c(u)$, regardless of the interval of the parametrization, whereas $\Lambda^* - M^*$ controls the piece of surface that is covered by the parametrization, which is larger the larger $|\Lambda^* - M^*|$ is.

If we exchange the roles of $c(u)$ and $d(u)$, Λ^* and M^* exchange theirs either.

This information may be encoded in the coefficients of the barycentric combination of one of the vertices of each cell in terms of the other vertices, extending the result (8) for piecewise quadratic curves to arbitrary degree n,

$$d_i = (1 - \lambda_i - \mu_i)c_{i-1} + \lambda_i c_i + \mu_i d_{i-1}, \quad i = 1, \ldots, L,$$
$$\lambda_i = \frac{\mu_{i-1}(u_{i-1} - u_{i-2}) + \lambda_{i-1}(u_{i+n-2} - u_{i-2}) - u_{i-1} + u_{i-2}}{u_{i+n-2} - u_{i-1} + \mu_{i-1}(u_{i-1} - u_{i-2})},$$
$$\mu_i = \frac{u_{i+n-2} - u_{i+n-1} + \mu_{i-1}(u_{i+n-1} - u_{i-2})}{u_{i+n-2} - u_{i-1} + \mu_{i-1}(u_{i-1} - u_{i-2})}, \quad i = 2, \ldots, L, \tag{18}$$

for B-spline control polygons $\{c_0, \ldots, c_L\}$, $\{d_0, \ldots, d_L\}$ and sequence of knots $\{u_0, \ldots, u_{L+n-1}\}$. Obviously these expressions provide constant values of λ and μ when applied to Bézier curves, according to Aumann's result.

These formulae, though cumbersome, have the advantage of being valid also for conical surfaces.

In fact, as it happened in the Bézier case, it is easy to check that the type of developable surface depends only on the values of λ_i, μ_i in one cell. If $\lambda_i = 1$ for one cell, every λ_i is equal to one for all cells and the surface is cylindrical. If $\mu_i = 1$ for one cell, every μ_i is equal to one for all cells and the surface is conical.

Another interpretation of the coefficients Λ and M arises when we calculate the Bézier control net of a segment of the spline developable surface. For a surface with just one segment, the control net may be obtained by evaluation of the blossom on sequences of the form $\{u_{n-1}^{\langle n-i-1\rangle}, u_n^{\langle i\rangle}\}$,

$$c\big[u_{n-1}^{\langle n-i-1\rangle}, u_n^{\langle i\rangle}, \Lambda^*\big] = (1-\Lambda)c\big[u_{n-1}^{\langle n-i\rangle}, u_n^{\langle i\rangle}\big] + \Lambda c\big[u_{n-1}^{\langle n-i-1\rangle}, u_n^{\langle i+1\rangle}\big] = (1-\Lambda)C_i + \Lambda C_{i+1},$$

and similarly for the other blossom. Therefore, we have

$$(1-\Lambda)C_i + \Lambda C_{i+1} = (1-M)D_i + M D_{i+1}, \tag{19}$$

for the Bézier control polygons $\{C_0, \ldots, C_n\}$, $\{D_0, \ldots, D_n\}$ of the segment of the curves $c(u)$, $d(u)$.

Therefore, generalizing to several segments, the coefficients Λ_i, M_i of the segment i of the surface define the shape of the Bézier control net of that segment. In fact, they are the coefficients arising in Aumann's algorithm (12) for that segment of the surface.

In few words we could say that each segment is a developable surface on its own and has its own coefficients Λ_i, M_i, according to Aumann's algorithm, but when we see them as parts of a C^1 spline developable surface they provide the same Λ^*, M^* in our algorithm.

Finally, the existence of singular points in Bézier developable surfaces has been discussed in (Aumann, 2003). Singular points can be prevented if the first cell of the control net is convex, that is, if $\lambda > 0$, $\mu > 0$, $\lambda + \mu > 1$. Otherwise singular points may appear in some or all rulings. These conditions are easily translated in terms of the coefficients Λ_i and M_i of the segments of the surface:

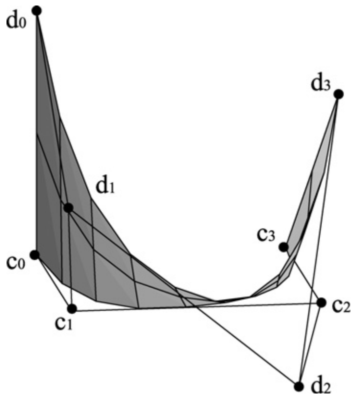

Fig. 8. B-spline quadratic developable surface formed by one regular and one singular segment.

Corollary 3. *A B-spline developable surface does not have singular points if the coefficients Λ_i, M_i of each segment satisfy either $\Lambda_i, M_i < 0$ or $\Lambda_i, M_i > 1$.*

Since crossing from the segment $k-1$ to the segment k involves substractions to the coefficients Λ_{k-1} and M_{k-1}, according to (10), it is clear that a *safe* choice of parameters for *all* sequences of knots would be starting the recursion with negative Λ_1, M_1. Otherwise we could start with regular segments and end up with singular segments. An example of a singular developable surface is shown in Fig. 8.

4. Interpolation

This algorithm for constructing developable B-spline surfaces may be used for solving the following interpolation problem:

Given a spline curve $c(u)$ of degree n, $u \in [a,b]$, and two straight lines l_a and l_b through the endpoints of $c(u)$, construct a developable surface $c(u,v)$ through $c(u)$ ($c(u,0) = c(u)$) with l_a and l_b as respectively first and last rulings of the surface ($l_a: \ c(a,v)$, $l_b: \ c(b,v)$).

The solution to this problem is similar to the one obtained for the Bézier case (Aumann, 2003). The curve $c(u)$ is defined by a B-spline control polygon $\{c_0,\dots,c_L\}$ and a sequence of knots $\{u_0,\dots,u_{L+n-1}\}$ with $u_0 = \cdots = u_{n-1} = a$, $u_L = \cdots = u_{L+n-1} = b$. The curve $d(u) = c(u,1)$ shall have B-spline control polygon $\{d_0,\dots,d_L\}$ and the same sequence of knots.

A developable surface through $c(u)$ has five free design parameters (three corresponding to d_0, $\Lambda_.$, $M_.$). Therefore, imposing a ruling (the line, not the end of the segment) implies using two design parameters. If we impose both first and last rulings, we are left generically with just one free design parameter.

If the rulings l_a and l_b are both parallel to a vector $\mathbf{v}$, we may construct a cylinder through $c(u)$ by choosing $d(u) = c(u) + \lambda(u)\mathbf{v}$ for arbitrary $\lambda(u)$. We just need to choose $\{d_0,\dots,d_L\}$ so that the vectors $\mathbf{c_i d_i}$ are all parallel to $\mathbf{v}$ for $i = 0,\dots,L$.

If the rulings l_a and l_b intersect at one point O, we may choose this point as the vertex of a cone through $c(u)$. We choose each point d_i on the line through O and c_i with fixed proportion α, $\alpha\mathbf{c_i c_{i-1}} = \mathbf{d_i d_{i-1}}$, $i = 1,\dots,L$.

For the generic case of skew rulings l_a and l_b we may solve the problem using the algorithm (15) for $N = L$, expressed as

$$d_L - c_L = \prod_{i=0}^{L-1} \frac{M^* - u_{i+n}}{M^* - u_i}(d_0 - c_0) + \frac{\Lambda^* - M^*}{M^* - u_{L-1}}\big(c_L - a(M^*)\big),$$

$$a(M^*) = \frac{M^* - u_{L-1}}{M^* - u_0} \prod_{i=1}^{L-1} \frac{M^* - u_{i+n}}{M^* - u_i} c_0 + \sum_{i=1}^{L-1} \frac{u_{i+n} - u_{i-1}}{M^* - u_{i-1}} \left(\prod_{j=i}^{L-2} \frac{M^* - u_{n+j+1}}{M^* - u_j} \right) c_i, \tag{20}$$

which relates vectors on the first and last rulings, $d_0 - c_0 = \sigma\mathbf{v}$, $d_L - c_L = \tau\mathbf{w}$, with a third vector, $c_L - a(M^*)$, which must be a linear combination of the other two if the expression is satisfied by a value M_0^*,

$$a(M_0^*) = c_L + \alpha\mathbf{v} + \beta\mathbf{w},$$

where the coefficients of the combination, α, β, may be calculated completing a basis $\{\mathbf{v}, \mathbf{w}, \mathbf{n}\}$ with the vector $\mathbf{n} = \mathbf{v} \times \mathbf{w}$ and solving the linear system,

$$\alpha = \frac{\det(a(M_0^*) - c_L, \mathbf{w}, \mathbf{n})}{\det(\mathbf{v}, \mathbf{w}, \mathbf{n})}, \qquad \beta = \frac{\det(\mathbf{v}, a(M_0^*) - c_L, \mathbf{n})}{\det(\mathbf{v}, \mathbf{w}, \mathbf{n})},$$

which substituted in (20),

$$\tau_0\mathbf{w} = \prod_{i=0}^{L-1} \frac{M_0^* - u_{i+n}}{M_0^* - u_i}\sigma_0\mathbf{v} + \frac{M_0^* - \Lambda^*}{M_0^* - u_{L-1}}(\alpha\mathbf{v} + \beta\mathbf{w}),$$

allow us to determine the coefficients on the rulings l_a and l_b,

$$\sigma_0 = \alpha\frac{\Lambda^* - M_0^*}{M_0^* - u_{L-1}} \prod_{i=0}^{L-1} \frac{M_0^* - u_i}{M_0^* - u_{i+n}}, \qquad \tau_0 = \beta\frac{M_0^* - \Lambda^*}{M_0^* - u_{L-1}} \tag{21}$$

in terms of Λ^* as free parameter, which may be used for imposing the surface to pass through a point on any of the rulings, for instance by choosing either d_0 or d_L, without changing the surface, just the boundary, according to (17). In general, it is not possible to choose both vertices, since we lack another free parameter. This problem is circumvented in the Bézier case by degree elevation of the surface (Aumann, 2004).

5. Conclusions

We have shown a characterization of generic B-spline developable surfaces in terms of the blossoms of the B-spline curves that span them. This provides a simple linear algorithm for constructing control nets of generic B-spline developable surfaces through a given B-spline curve that depends on five additional parameters. For instance the control net may be generated by providing the B-spline control polygon of one of the curves and the first two vertices of the B-spline control polygon of the second curve. This algorithm can be considered a generalization of Aumann's algorithm for Bézier developable surfaces to B-spline developable surfaces. The relation with this algorithm has been explicitly shown. For future work it remains open how to extend these results to the rational case.

This result can be used for solving several problems concerning developable surfaces. For instance, it can be used for interpolating a B-spline developable surface through a spline curve, providing the rulings of the surface at both ends of the curve. The knowledge of the general form of the control net for B-spline developable surfaces could also be useful for producing approximation algorithms based on generic developable surfaces instead of restricting to conical and cylindrical surfaces.

References

Aumann, G., 1991. Interpolation with developable Bézier patches. Computer Aided Geometric Design 8, 409–420.

Aumann, G., 2003. A simple algorithm for designing developable Bézier surfaces. Computer Aided Geometric Design 20, 601–619.

Aumann, G., 2004. Degree elevation and developable Bézier surfaces. Computer Aided Geometric Design 21, 667–670.

Bodduluri, R.M.C., Ravani, B., 1993. Design of developable surfaces using duality between plane and point geometries. Computer-Aided Design 25, 621–632.

de Boor, C., 1972. On calculating with B-splines. J. Approx. Theory 6, 50–62.

Chalfant, J.S., Maekawa, T., 1998. Design for manufacturing using B-spline developable surfaces. J. Ship Production 42, 207–215.

Chu, C.-H., Séquin, C.H., 2002. Developable Bézier patches: properties and design. Computer-Aided Design 34, 511–527.

Farin, G., 2001. Curves and Surfaces for CAGD, fifth ed. Morgan Kaufmann, San Francisco.

Frey, W.H., Bindschadler, D., 1993. Computer aided design of a class of developable Bézier surfaces. GM Research Publication R&D-8057.

Lang, J., Röschel, O., 1992. Developable $(1, n)$-Bézier surfaces. Computer Aided Geometric Design 9, 291–298.

Leopoldseder, S., 2001. Algorithms on cone spline surfaces and spatial osculating arc splines. Computer Aided Geometric Design 18, 505–530.

Postnikov, M., 1979. Lectures in Geometry: Linear Algebra and Differential Geometry. Nauka, Moscow.

Pottmann, H., Farin, G., 1995. Developable rational Bézier and B-spline surfaces. Computer Aided Geometric Design 12, 513–531.

Pottmann, H., Wallner, J., 1999. Approximation algorithms for developable surfaces. Computer Aided Geometric Design 16, 539–556.

Struik, D.J., 1961. Lectures on Classical Differential Geometry. Dover, New York.