Approximation Algorithms for Developable Surfaces

Helmut Pottmann and Johannes Wallner
Institut für Geometrie, Technische Universität Wien
Wiedner Hauptstraße 8–10, A–1040 Wien, Austria

Abstract

By its dual representation, a developable surface can be viewed as a curve of dual projective 3-space. After introducing an appropriate metric in the dual space and restricting ourselves to special parametrizations of the surfaces involved, we derive linear approximation algorithms for developable NURBS surfaces, including multiscale approximations. Special attention is paid to controlling the curve of regression.

Keywords: computer aided geometric design, surface approximation, developable surface, dual representation, NURBS

1 Introduction

A *developable surface* is a surface which can be unfolded (*developed*) into a plane without stretching or tearing. Mathematically speaking, there is a mapping of the surface into the Euclidean plane which is isometric, at least locally. Because of this property, developable surfaces possess a variety of applications in manufacturing with materials that are not amenable to stretching. These include the formation of aircraft skins, ship hulls, ducts and automobile parts such as upholstery, body panels and windshields (see e.g. [8]).

Since current CAD/CAM systems are using rational B-splines (NURBS) as standard for curve and surface representations [7, 18], there is a demand for efficient computing with developable NURBS surfaces. There are basically two approaches to dealing with rational developable surfaces. On the one hand, one can express such a surface as a tensor product surface of degree $(1, n)$ and solve the nonlinear side conditions expressing the developability [1, 2, 5, 14]. On the other hand, we can view the surface as envelope of its one parameter set of tangent planes and thus treat it as a curve in dual projective space [3, 4, 11, 12, 19, 20]. Based on the latter approach, some interpolation and approximation algorithms as well as initial solutions to special applications have been presented recently [11, 12, 13, 20, 22, 24]

In the present paper we develop further the use of the dual representation for the solution of fundamental approximation problems with developable surfaces. The new algorithms are based on appropriate metrics in dual space as well as on limitation to special surface classes. Thus most of them are of a linear nature. Only pushing out the line of regression from the area of interest requires the solution of convex programming problem.

2 The dual representation of developable surfaces

Developable surfaces can be isometrically mapped (*developed*) into the plane, at least locally. When sufficient differentiability is assumed, they are characterized by the vanishing of their Gaussian curvature. A non-flat developable surface is the envelope of its one parameter family of tangent planes. Such a developable surface locally is either a conical surface, a cylindrical surface, or the tangent surface of a twisted curve. Globally, of course, it can be a rather complicated composition of these three surface types. Thus, developable surfaces are ruled surfaces, but with the special property that they possess the same tangent plane at all points of the same generator (=*ruling*).

We will do our calculations in the projective extension P^3 of real Euclidean 3-space E^3 and use homogeneous Cartesian coordinates (x_0, x_1, x_2, x_3) for points. For points not at infinity, i.e., $x_0 \neq 0$, the corresponding inhomogeneous Cartesian coordinates will be denoted by

$$x = \frac{x_1}{x_0}, \quad y = \frac{x_2}{x_0}, \quad z = \frac{x_3}{x_0};$$

we write $X = (x, y, z)$.

A *plane* with equation $u_0x_0 + u_1x_1 + u_2x_2 + u_3x_3 = 0$, or, equivalently, $u_0 + u_1x + u_2y + u_3z = 0$ can be represented by its *homogeneous plane coordinates* $U = (u_0, u_1, u_2, u_3)$.

Because in all points of a generator line the tangent plane is the same, we can identify a developable surface with the one-parameter family of its tangent planes $U(t)$, or in other words, with a certain curve in dual projective space. If this curve is a NURBS curve

$$U(t) = \sum_{i=0}^{n} U_i N_i^k(t), \qquad (1)$$

we call the original surface a *developable NURBS surface.* Here the N_i^k denote normalized B-spline basis functions of degree k over a given knot vector. The word ‘normalized’ means that the sum of the basis functions is the constant function 1. The symbol U_i denotes a coordinate quadruple of the i-th *control plane* U_i. Of course the coordinate quadruple contains more information than just the plane as a point set, but for simplicity we just speak of the coordinates of the plane.

There is no mathematical reason why we restrict ourselves to the B-splines. They have, however, a lot of properties which make them easy to deal with. A theorem is easily verified to hold for developable surfaces which are modeled by a different spline space as well, if this spline space enjoys all properties of the B-spline space which are used in the proof of this theorem.

It is well known that the plane $U(t)$ touches the envelope of the family $U(t)$ along the generator line

$$U(t) \cap \dot{U}(t).$$

In particular, the rulings which correspond to parameter values which are $(k+1)$-fold knots (usually t_0 and t_n), can easily be expressed in terms of the control planes.

The cuspoidal edge or *line of regression* of the surface is obtained as the intersection

$$U(t) \cap \dot{U}(t) \cap \ddot{U}(t).$$

In general it is a rational B-spline curve of degree $3k-6$.

Recently, algorithms for the computation with the dual representation, the conversion to the standard tensor product representation and the solution of interpolation and some approximation algorithms have been developed [11, 12, 20]. In this paper we explore further approximation of and with developable surfaces. This is not a straightforward application of duality, as might be expected, since duality does not extend to the Euclidean metric and, moreover, Euclidean geometry does not contain deviation measures between planes that would be useful in the present context.

3 A special class of developable NURBS surfaces

3.1 Definitions and elementary properties

For the approximation algorithms discussed in this paper, we will restrict the class of developable surfaces we are working with: We only consider surfaces whose family of tangent planes is of the form

$$U(t) = (u_0(t), u_1(t), u_2(t), -1). \tag{2}$$

For NURBS surfaces this is equivalent to the choice of control planes $U_i = (u_{0,i}, u_{1,i}, u_{2,i}, u_{3,i})$ such that always $u_{3,i} = -1$. This means that for all possible planes

U we no longer allow to choose an arbitrary coordinate quadruple describing U, but we restrict ourselves to the unique one whose last coordinate equals -1. This is not possible if the last coordinate is zero, so we have to exclude all surfaces with tangent planes parallel to the z-axis. In most cases this requirement is easily fulfilled by choosing an appropriate coordinate system.

We can also use other basis functions (instead of the normalized B-splines), which do not sum up to 1. The only difference is that we have to set $u_3(t)$ to -1 and to ignore the third coordinate of planes when computing $u_i(t)$. This makes the formulae more clumsy, but is no essential restriction.

Dual projective space with the bundle of planes $(u_0, u_1, u_2, 0)$ removed is an affine space and (u_0, u_1, u_2), describing the plane $(u_0, u_1, u_2, -1)$, are affine coordinates in it. The surfaces (2) become ordinary piecewise polynomial B-spline curves.

For most applications it is convenient to restrict the class of surfaces even further by prescribing the parametrization:

$$U(t) = (u_0(t), u_1(t), t, -1). \tag{3}$$

Its generators $g(t)$ lie in the first derivative planes, which now have the form

$$\dot{U}(t) = (\dot{u}_0, \dot{u}_1, 1, 0) \tag{4}$$

As it was with the surfaces of type (2), it does not matter whether or not the spline space contains the identity function $t \mapsto t$. The set of parametrizations $(u_0(t), u_1(t), t, -1)$ with u_0 and u_1 from our spline space still is a well-defined set of functions whose properties can be studied, and which metrics of an ambient space can be restricted to (see later).

The next lemma shows some limitations of the class (3):

Lemma 1 *The surfaces (3) do not possess inflection generators and generators parallel to the plane $x = 0$.*

Proof: The generator $g(t_0)$ is an inflection generator if and only if $U(t)$ has a singularity as a curve in projective space. For surfaces of type (2) this happens if and only if $\dot{U}(t) = 0$, and it never happens for surfaces of type (3).

Further, direct calculation gives the intersection point of the three planes $U(t)$, $\dot{U}(t)$ and $x = 0$, which never is situated at infinity. □

Because inflection generators are singularities in dual space, they need a special treatment anyway. In approximation problems we will have to cut the surface which we want to approximate into pieces which do not contain inflection generators. For a detailed study of inflection generators of developable surfaces in connection with applications in manufacturing, we refer the reader to [6].

For more insight into the geometry of the special developable surfaces of type (3), let us intersect the surface with the plane $x = c$.

Lemma 2 *The intersection curve C_c of a NURBS surface (3) with the plane $x = c$ is a polynomial B-spline curve of degree k and knot multiplicities one greater than the knot multiplicities of (3).*

Proof: The curve C_c is the envelope of the lines $z = u_0 + cu_1 + ty$. Thus the geometric meaning of the parameter t is the *tangent slope* of the intersection curves C_c. An elementary calculation gives the parametric representation of C_c:

$$x = c,\ y = -\dot{h}_c(t),\ z = h_c(t) - t\dot{h}_c(t),\ \text{with}\ h_c(t) := u_0(t) + cu_1(t). \qquad (5)$$

We see that these are *polynomial B-spline curves.* The function $t\dot{h}_c$ is of polynomial degree less or equal k and its differentiability class at the knot values is one less then the differentiability class of the u_i. This implies the statement about the multiplicities. □

Corollary 3 *A developable NURBS surface (3) can also be written as a polynomial tensor product B-spline surface of degree* $(1, k)$*,*

$$S = (1 - u)C_a(t) + uC_b(t),$$

where C_a and C_b are the intersection curves in planes $x = a$ and $x = b$.

Developable tensor product B-spline surfaces with boundary curves in parallel planes have been investigated in several papers [1, 2, 5], but the computational simplicity of our special subclass (3) remained unobserved so far.

4 Approximation algorithms

Our treatment of approximation problems is based on two ingredients. First, we are limiting our candidate surfaces which we would like to use for approximation to special subclasses discussed in the previous section. Second, we are using appropriate error measures, which will be discussed now.

4.1 Distance functions between planes

In order to approximate a given set of planes by another one, it is necessary to introduce an appropriate *distance* between two planes. Euclidean geometry does not directly provide such a distance function. All invariants are expressed in terms of the angle between planes and are inappropriate for our purposes, because we are only interested in the distances of points of the two planes which are near some region of interest, and this distance can become arbitrarily large with the angle getting arbitrarily close to zero at the same time.

In order to keep certain algorithms linear, we approach the problem in the following way. When designing a developable surface, we do it in pieces for which there exists a vector $e \in \mathbb{R}^3$ such that the angle between the surface normals and e does not exceed some angle $\gamma_0 < \pi/2$. Then, e is taken as third unit vector of a Cartesian coordinate system. Now all tangent planes of the surface can be written as graph of a linear function in x and y as $z = u_0 + u_1 x + u_2 y$.

For a positive measure μ in $\mathbb{R}^2$ we define the distance d_μ between planes $U_i = (u_{0,i}, u_{1,i}, u_{2,i}, -1)$ as

$$d_\mu(U_1, U_2) = \|(u_{0,1} - u_{0,2}) + (u_{1,1} - u_{1,2})x + (u_{2,1} - u_{2,2})y\|_{L^2(\mu)}, \tag{6}$$

i.e., the $L^2(\mu)$-distance of the linear functions whose graphs are U_1 and U_2. This, of course, makes sense only if the linear function which represents the difference between the two planes is in $L^2(\mu)$. We will always assume that the measure μ is such that all linear and quadratic functions possess finite integral.

A useful choice for μ is Lebesgue measure $dxdy$ times the characteristic function of a *region of interest*. If $\mu = dxdy\chi_D$, we have

$$d_\mu(U_1, U_2)^2 = \int_D ((u_{0,1} - u_{0,2}) + (u_{1,1} - u_{1,2})x + (u_{2,1} - u_{2,2})y)^2 dxdy. \tag{7}$$

We write $d_D(U_1, U_2)$ instead of $d_\mu(U_1, U_2)$.

Another possibility is that μ equals the sum of several point masses at points (x_i, y_i), see [11]. In this case we have

$$d_\mu(U_1, U_2)^2 = \sum_j ((u_{0,1} - u_{0,2}) + (u_{1,1} - u_{1,2})x_j + (u_{2,1} - u_{2,2})y_j)^2. \tag{8}$$

Lemma 4 *The distance d_μ defines a Euclidean metric in the set of planes of type (2), if and only if μ is not concentrated in a straight line.*

Proof: The coefficients of the planes enter (7) in a bilinear way. Symmetry and positive semi-definiteness follow from the respective properties of the L^2 scalar product. The positive definiteness is also seen easily:

Suppose the zero set of the nonzero linear function $f(x, y)$ the line g. μ is not concentrated on g, so there is a measurable set E with $\mu(E \setminus g) > 0$. Let $A_n = \{P \in \mathbb{R}^2 | 2^{-(n+1)} \leq \overline{Pg} < 2^{-n}\}$. Then $\mu(E \setminus g) = \sum_{n\in\mathbb{Z}} \mu(E \cap A_n)$, so there is an i such that $\mu(E \cap A_i) > 0$. In A_i the function f^2 is bounded from below by $c > 0$, so $\|f\|^2_{L^2(\mu)} \geq \mu(A_i \cap E) \cdot c > 0$ and the metric is positive definite. The converse is obvious. □

Because the space of symmetric bilinear forms in $\mathbb{R}^3$ is six-dimensional, the variety of distance functions between planes is not as great as it may seem. For example, the problem, given a metric, to determine three points such that (8) reproduces this metric up to a scalar factor, is quadratic in the six unknown coordinates.

4.2 Approximation of tangent planes

Consider the following approximation problem. Given m planes $V_1, \dots, V_m$ and corresponding parameter values v_i, approximate these planes by a developable surface $U(t)$, such that $U(v_i)$ is close to the given plane V_i within an associated area of interest, where i ranges from 1 to m.

The meaning of 'close' is the following: There is a Cartesian coordinate system fixed in space such that all planes are graphs of linear functions of the xy-plane. Its third unit vector may be found as solution of a regression problem to the given plane normals. For all i there is a region of interest D_i, or, more generally, a measure μ_i, in the xy-plane. We want to minimize

$$F_1 := \sum_{i=1}^{m} d_{\mu_i}(V_i, U(v_i))^2, \tag{9}$$

for an unknown developable surface $U(t)$. If $U(t)$ is a NURBS surface of type (2), F_1 is a quadratic function in the unknown coordinates of the control planes U_i. These can then be found by solving a linear system of equations.

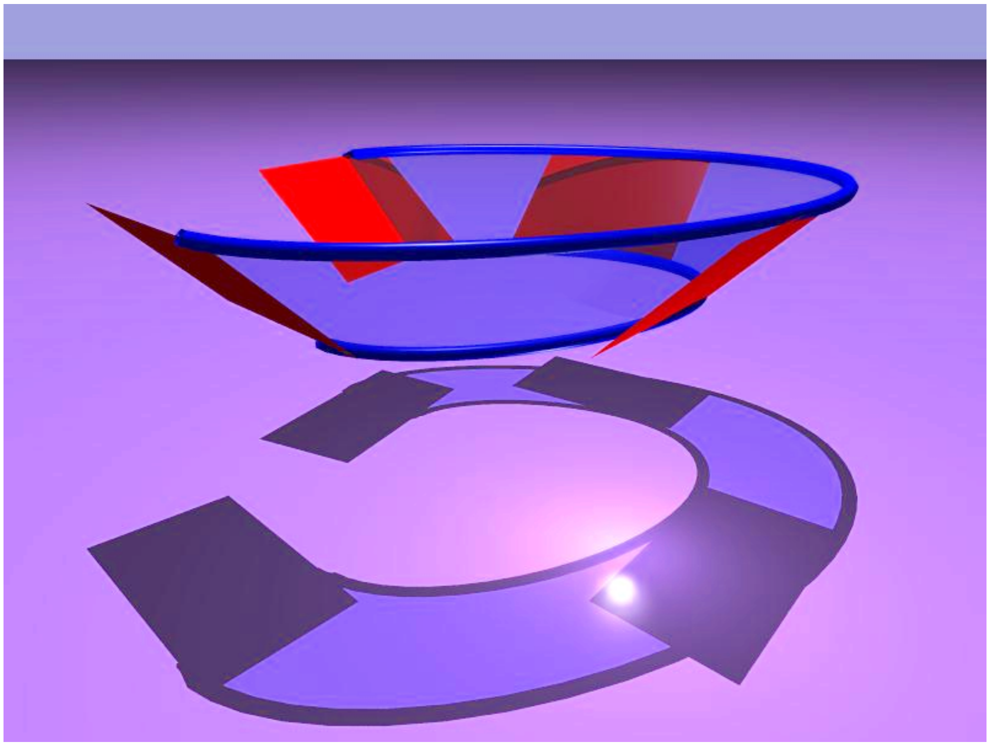

Figure 1: Approximation of a set of planes by a developable surface

A good choice for μ_i would be $w_i \chi_{D_i} dxdy$. An example of this can be seen in Fig. 1. The positive weights w_i can be used to assign more or less importance to the single parameter values v_i. It would also be possible to choose different

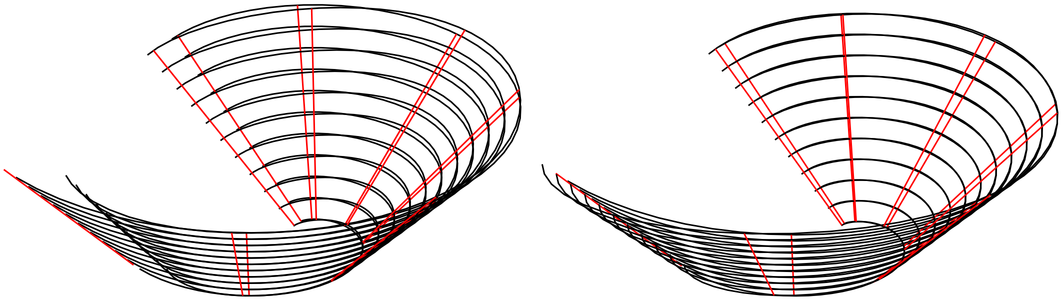

Figure 2: Approximation of a developable surface by a spline torse. Left: original parameter values. Right: Result after a parameter correction. Number of basis functions: 5

coordinate systems for different planes V_i, but this is not necessary, because it is equivalent to multiplying the weights w_i with appropriate factors. With $w_i = \sin^2 \gamma_i$, where γ_i is the Euclidean angle, which is enclosed between V_i and the z-axis, we can correct the influence of measuring distances in the z-direction of a fixed coordinate system for all i.

One may fix some boundary control planes in order to ensure a smooth join of subsequent surface segments. Note that the computation of the surface $U(t)$ is equivalent to a polynomial B-spline curve approximation problem using different Euclidean metrics at different points to be approximated. Working with the same μ or D for all planes, we get an ordinary curve approximation problem in Euclidean 3-space [7, 10, 18].

Since the parameters v_i have to be fixed in advance and another choice could have given better results, one will start with an initial guess and then improve it by *parameter correction*. With the Euclidean norms defined above, we can directly apply the known computational schemes [10]. An example is shown in Fig. 2.

If a given developable surface $V(t)$ has to be approximated, we may either work with discrete tangent planes as above, or approximate the parameterized surface $V(t),\ t \in [v_0, v_1]$, by minimizing the quadratic function

$$F_2 := \int_{v_0}^{v_1} d_{\mu(t)}(V(t), U(t))^2 dt. \tag{10}$$

An application of this is *approximate degree reduction* of developable NURBS surfaces.

4.3 Including data points and generators

Let us now discuss the introduction of generators and surface points into the approximation. We assume that a coordinate system has been defined and a

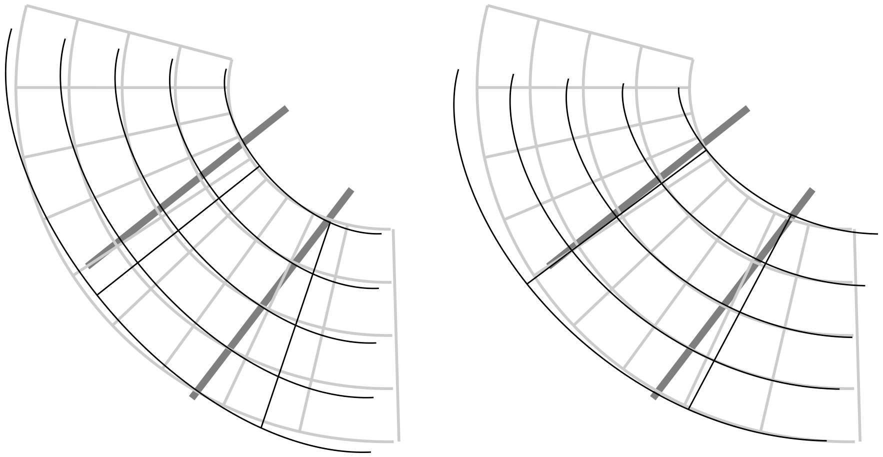

Figure 3: Approximation of a developable surface (light grey) by a developable spline surface. Left: Approximation (result is shown black) using only tangent planes (not shown, cf. Fig. 1). Right: Approximation of planes plus two generators (grey).

segmentation has been performed, such that the surface segment we are dealing with does not possess a generator parallel to the yz-plane or an inflection generator. Then, the deviation $\delta(g(t), h)$ of a given line h and the generator $g(t)$ of the NURBS surface $U(t)$ is measured, in analogy to subsection 4.1, with the help of a positive measure on the real line $\mathbb{R}$. For simplicity, we will formulate the definition only for the case that this measure is Lebesgue measure in an interval I.

Because the generator $g(t)$ is contained in the plane $\dot{U}(t) = (\dot{u}_0, \dot{u}_1, 1, 0)$, the projection of $g(t)$ into the plane $z = 0$ is the line $\dot{u}_0 + \dot{u}_1 x + y = 0$. Thus, we define

$$\delta_I(h, g(t))^2 := \int_I (h_0 - \dot{u}_0(t) + (h_1 - \dot{u}_1(t))\, x)^2 \, dx. \tag{11}$$

Here, $0 = h_0 + h_1 x + y$ is the projection of h into the plane $z = 0$. As remarked above, we could use here also discrete unit masses or other measures μ such that the linear and quadratic functions are in $L^2(\mu)$. An Example of approximation including generators is shown in Fig. 3

The function δ alone does not lead a positive definite metric, but when added to the distance measured between planes, it will serve as a correction term which accounts for the generator lines. In a similar way the distance of a point $P = (p_1, p_2, p_3)$ to a generator $g(t)$ is given by

$$\Delta(P, g(t)) := |\dot{u}_0(t) + \dot{u}_1(t) p_1 + p_2| \tag{12}$$

together with the corresponding tangent plane deviation.

Given m tangent planes V_i plus generators g_i, we can approximate these data by a NURBS surface of the form (3) as follows. After an appropriate segmentation (see the discussion above) and the choice of local coordinate systems, the plane coordinates $V_i = (\ldots, v_i, -1)$ with $v_i \neq v_j$ if $i \neq j$, already determine the parameters v_i which have to be used in formulas like (9). Then, with the measures μ_i from (9) and intervals I_i from (11), we define the quadratic function

$$F_3 := \sum_{i=1}^{m} \left(d_{\mu_i}(V_i, U(v_i))^2 + \alpha_i \delta_{I_i}(g_i, g(v_i))^2 \right). \tag{13}$$

Again, weights α_i can be used to correct error measurement directions or give more importance to certain indices i. The surface $U(t)$ then is found as the linear combination of the basis functions which minimizes F_3.

Analogously, we can incorporate data points into the approximation. If data points or generators are given without tangent planes, the latter must be estimated before this method can be applied. There is no problem in setting up the counterpart to (10) for the approximation of a given developable surface.

4.4 Controlling the curve of regression

Since its line of regression is a singularity of a developable surface, it is desirable that it should lie outside some pre-defined area of interest. To achieve this when approximating tangent planes and generators like in the previous subsection, we can do the following:

For a surface of type (3), the point of regression at the parameter t lies in the plane $\ddot{U}(t)$, which is given by $(\ddot{u}_0, \ddot{u}_1, 0, 0)$. Its x-coordinate is easily found to be $-\ddot{u}_0/\ddot{u}_1$. Thus, we have the following

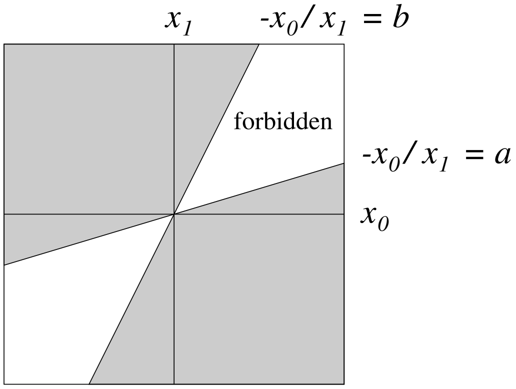

Figure 4: Forbidden region (white) for $\ddot{u}_0 = x_0$, $\ddot{u}_1 = x_1$

Lemma 5 *In order to keep the point of regression outside the area $a \leq x \leq b$, the surface $U(t)$ of type (3) must satisfy $-\ddot{u}_0/\ddot{u}_1 \notin [a,b]$, or, equivalently,*

$$\left| \frac{\ddot{u}_0(t)}{\ddot{u}_1(t)} + c \right| > r, \tag{14}$$

with $c = (b+a)/2$ and $r = (b-a)/2$.

The forbidden area can be seen in Fig. 4.

If the spline space which contains the functions $u_0(t)$ and $u_1(t)$ has the property that the second derivatives of its members are contained in the spline space which enjoys the two-dimensional convex hull property, it is easy to formulate a condition on the control points of the curve $(\ddot{u}_0, \ddot{u}_1)$:

Lemma 6 *In the situation mentioned above, if all control points of the curve $(\ddot{u}_0(t), \ddot{u}_1(t))$, are situated in one of the two connected components of the grey area indicated in Fig. 4, then the line of regression is outside the area $a \leq x \leq b$.*

Proof: This follows immediately from the convex hull property of the spline space, because then the curve $(\ddot{u}_0, \ddot{u}_1)$ is entirely contained in one of the two grey regions above. □

Note that 'outside the area $a \leq x \leq b$' does not mean 'on one of the two sides of the area $a \leq x \leq b$'. The curve of regression can have points at infinity and change from one side to the other. The two regions in Fig. 4 do *not* correspond to the two sides of the region $a \leq x \leq b$.

We should also remark that we tacitly also excluded the case $(\ddot{u}_0, \ddot{u}_1) = (0,0)$ as 'forbidden', because the corresponding point of regression is a point at infinity which is contained in the projective extension of *all* regions of the form $a \leq x \leq b$. This however does not matter very much because it does not occur in the generic case anyway, and if it does, we could apply a coordinate transformation.

It is well known that the second derivatives of B-spline functions are B-spline functions of a lower degree, so the lemma is applicable in this case.

Example: If $u_i(t)$ are B-spline functions of order two, their second derivatives are piecewise constant and it is very easy to test whether or not the line of regression meets the region $a \leq x \leq b$.

Example: If $u_i(t)$ are B-spline functions of order three, their second derivatives are piecewise linear and lemma 6 is sharp, which means, that the line of regression avoids the region $a \leq x \leq b$ if and only if the control points of the curve $(\ddot{u}_0, \ddot{u}_1)$ are contained in one of the two convex regions of Fig. 4.

We are going to describe an algorithm how to find the developable surface contained in a given spline space which is closest to a given developable surface in some sense which was previously defined.

Choose one of the two grey convex unbounded polytopes of Fig. 4 and call it K. The spline space which contains u_0 and u_1 shall have basis functions $f_1, \ldots, f_n$. The second derivatives $\ddot{f}_0, \ldots, \ddot{f}_n$ are contained in a spline space with basis functions $g_1, \ldots, g_m$ and can therefore be written as

$$\ddot{f}_i(t) = \sum_j r_{ij} g_j(t).$$

The r_{ij} are either well known or can be found numerically by differentiating the f_i twice and then approximating this function by a linear combination of the g_j.

Thus there is a linear mapping L which maps the sequence of control points of the curve $(u_0(t), u_1(t))$ to the sequence of control points of $(\ddot{u}_0(t), \ddot{u}_1(t))$. Now the sequence of control points of the second derivative curve is contained in $K \times \ldots \times K$ if and only if the sequence of control points of the curve (u_0, u_1) is contained in the convex polytope

$$\widetilde{K} = L^{-1}(K \times \ldots \times K).$$

The equations of K are very simple, therefore so are the equations of the m-fold product of K with itself. The equations of its L-preimage are easily found by solving a linear system.

Now we are able to reformulate the problem as follows: Given a convex polyhedron $\widetilde{K}$ in $\mathbb{R}^r$ together with a scalar product and a point o. Find the point $p \in \widetilde{K}$ which is closest to o in the sense of the metric which is defined by the scalar product. This problem is well known and there is an extensive literature about it.

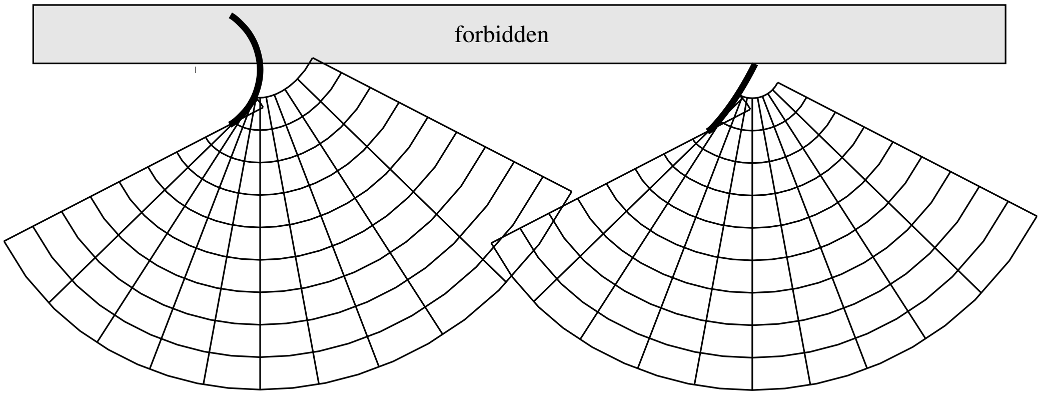

Figure 5: Left: Top view of developable surface with line of regression. Right: Approximation such that line of regression avoids forbidden area

In the case of B-spline functions the linear mapping L is onto. Therefore the structure of $\widetilde{K}$ is that of a product of $K \times \ldots \times K$ with an $\mathbb{R}^s$, s being the difference in dimensions of the spline space which contains the u_i and the spline space which contains their second derivatives. So it is easy to find linear

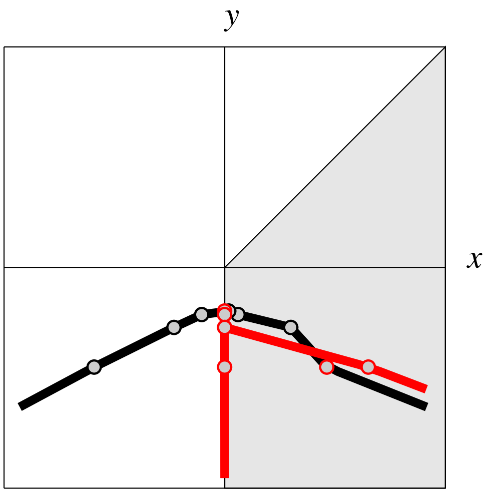

Figure 6: The (piecewise linear) curves $(\ddot{u}_0, \ddot{u}_1)$ before (black) and after (grey) pushing out the line of regression from the forbidden area.

equalities for $\widetilde{K}$ none of which is redundant, and such that exactly d of them define a d-codimensional face of $\widetilde{K}$. Thus it is not very difficult to see that to determine the point of $\widetilde{K}$ which is closest to o we have to follow the following **Algorithm:**

1. Choose an interior point p_0 of $\widetilde{K}$. An obvious choice is a point of $L^{-1}((\pm 1, 0) \times \ldots \times (\pm 1, 0))$. The sign in the definition of p_0 depends on which one of the two possible K's we have chosen.

2. If $o \in \widetilde{K}$, let $s = o$ and go to 6. If not, intersect the line segment $[p_0, o]$ with $\partial \widetilde{K}$. This gives the point p_1.

3. Initialize the value of the *current face* F with the perhaps not uniquely defined 1-codimensional boundary face of $\widetilde{K}$ which contains p_1, and let the *current codimension* $d = 1$. Let $p = p_1$. In the following, the symbol $[F]$ denotes the affine hull of F, and $n(o, F)$ denotes the point of $[F]$ which is closest to o.

4. Let $q = n(o, F)$. If $q \in F$, let $s = q$ and go to 6. If not, follow the line segment $[p, q]$ until it leaves F at the point r.

5. For all d-codimensional faces G of $\widetilde{K}$ which contain the $(d+1)$-codimensional face H of ∂F defined by r follow the oriented line segment $[r, n(o, G)]$. If there is a G such that this line segment points from r towards the interior of G, let $F = G$ and $p = r$, and go to 4. If there is none, let $F = H$, $p = r$, increase d by one, and go to 4.

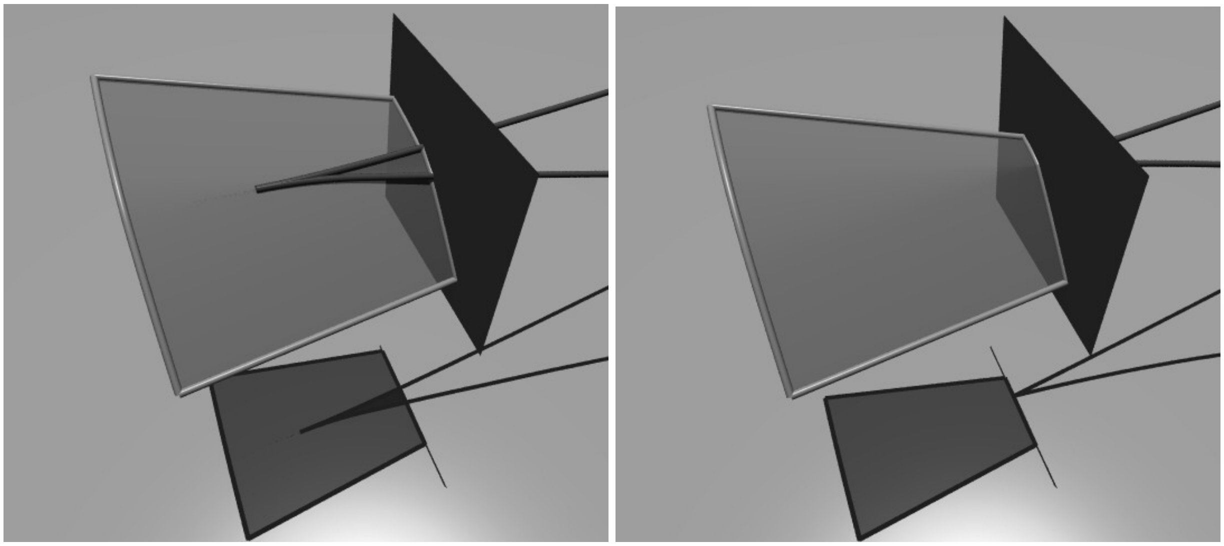

Figure 7: Perspective View of an approximation of a developable surface by another devolopable surface such that the line of regression is contained in the half-space right of the vertical plane.

6. Repeat the whole process for the other choice of K also. Among the two values of s choose the one with smaller distance to o.

Proof: (Sketch) If K is a smooth convex surface in $\mathbb{R}^n$, we consider the region K' of K which is illuminated if we think of o as of a light source. The distance to the point o is a smooth function defined in K with a nowhere vanishing gradient field. In K' the flow lines of this field never increase their distance. When following the gradient flow we arrive at the solution in finite time.

If K is a polyhedron, the flow lines are straight lines. We consider the smooth surface $K + \epsilon D$, D being the unit ball of $\mathbb{R}^n$. It contains 1-codimensional planar parts F_ϵ which are translates of the 1-codimensional faces F of K. Obviously, for $\epsilon \to 0$ the flow lines of the gradient field in F_ϵ converge to the lines in F which pass through $n(o, F)$.

Let G be a d-dimensional face of K, and consider the $(d+1)$-dimensional faces $G_1, \ldots, G_k$ which contain G. Because the distance of flow lines cannot increase, it is not possible that flow lines are emanating from G into more than one of the G_i. If no flow line leaves G, we say that the flow lines of G are *trapped* in dimension d. The same argument about the distance of flow lines shows that once flow lines are trapped in dimension d, they stay in faces of dimension $\leq d$.

To find the solution, we have therefore to do the following: Check if $o \in K$. If not, go to a point of K'. and follow the flow lines. The algorithm above does just this. Because there are a finite number of faces and each face occurs at most once in the algorithm (because following the flow lines decreases the distance to o), it converges to the solution in a finite time. □

Figures 5, 6 and 7 show examples. Of course other standard methods of convex programming can be employed also, for instance a barrier-generated path-following method. Such a method is more recommended when the structure of the polyhedron is not completely known [17]. We chose this algorithm, and we described it at length despited the fact that many algorithms for quadratic/convex optimatization problems can be found in the literature, because our polyhedron has such a simple structure that the most simple and obvious geometric algorithm, i.e., following the gradient lines on the polyhedron, does not lead to numerical difficulties.

4.5 Approximation via multiresolution analysis

There are several ways to apply the concept of multiresolution analysis to developable surfaces. One is, of course, to treat the surface of type (2) or type (3) as a curve in affine $\mathbb{R}^3$ or $\mathbb{R}^2$, respectively. This is nothing but multiresolution analysis for curves, and leads likewise to efficient filter bank decomposition and storage of developable surfaces.

A different problem is the following. Just like a planar curve can be approximated by an arc spline, i.e., a curve which consists of circular arcs, we may ask for an approximation of a developable surface by a surface consisting of certain quadratic cones. We could use cones of revolution, or we could use cones all of whose intersection curves with horizontal planes are circles. The latter will turn out to lead to the planar arc spline approximation problem.

In [25] the support functions of (locally) convex curves have been approximated by *trigonometric spline functions* and analyzed by a generalized multiresolution analysis which was introduced in [15]. Here we are interested only in the part of the developable surface which lies between two parallel planes. Without loss of generality we assume that these planes are $z = 0$ and $z = 1$. Then the surface has intersection curves c_0 and c_1 with these two planes. When we approximate both c_0 and c_1 by arc splines, the unique developable surface determined by the arc splines will be an approximant to the original surface. In order to diminish the number of lines of curvature disconinuity in the approximant even further, we choose the same spline space for approximation of c_0 and c_1. An example can be seen in Fig. 9.

To accomplish this, we have to define a distance between two such surfaces. First we define a distance between planes whose contour lines are parallel. It will be a positive definite quadratic form of the *oriented* distances h_0 and h_1 of the contour lines in $z = 0$ and $z = 1$, respectively. One obvious possibility is

$$q(h_0, h_1) = h_0^2 + h_1^2. \tag{15}$$

Another one is

$$q(h_0, h_1) = \int_0^1 h(z)^2 dz = \frac{1}{3}(h_0^2 + h_0 h_1 + h_1^2), \tag{16}$$

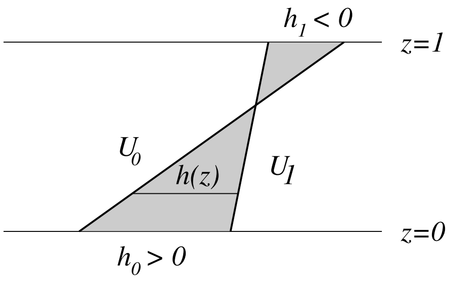

Figure 8: Deviation between planes U_1 and U_2

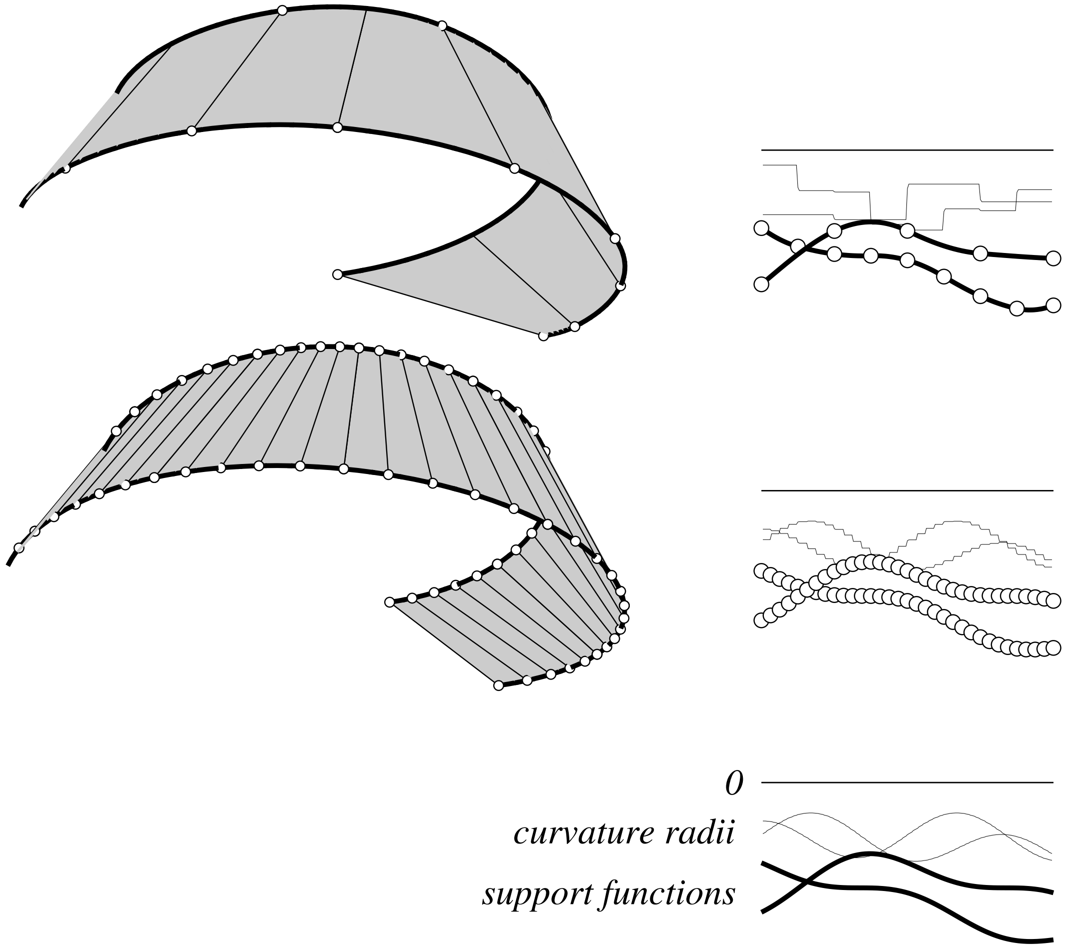

Figure 9: Multiresolution analysis of developable surfaces. The figure shows a fine and a coarse approximation as well as the support functions and the curvature radii of the contour lines in the planes $z = 0$ and $z = 1$. At the bottom right the support functions and curvature radii of the original surfare are shown.

where $h(z) = zh_1 + (1-z)h_0$.

Now suppose that $f_i(\varphi)$ and $g_i(\varphi)$ are the support functions of the intersection curves of two developable surfaces Φ and Ψ with planes $z = i$, $i = 0, 1$. For given φ, the distance between the tangent planes to Φ and Ψ which belong to the angle φ, can be expressed in terms of $h_0(\varphi) = f_0(\varphi) - g_0(\varphi)$ and $h_1(\varphi) = f_1(\varphi) - g_1(\varphi)$. Thus we define

$$d_\mu(\Phi, \Psi) = \int q(f_0(\varphi) - g_0(\varphi), f_1(\varphi) - g_1(\varphi)) d\mu(\varphi), \tag{17}$$

with an appropriate positive measure μ. Typically this will be Lebesgue measure in an interval. We further let

$$\beta((u_0, u_1), (v_0, v_1)) = \frac{1}{2}(q(u_0 + v_0, u_1 + v_1) - q(u_0, u_1) - q(v_0, v_1)), \tag{18}$$

which is the unique symmetric bilinear form whose restriction to the diagonal gives q. Then the distance (17) is induced by the following scalar product on $L^2(\mu) \oplus L^2(\mu)$:

$$\iota_\mu((f_0, f_1), (g_0, g_1)) = \int \beta((f_0(\varphi), f_1(\varphi)), (g_0(\varphi), g_1(\varphi))) d\mu(\varphi) \tag{19}$$

Lemma 7 *The scalar product (19) is positive definite in $L^2(\mu) \oplus L^2(\mu)$.*

Proof: Clearly $\int q(f_0, f_1) d\mu(\varphi) = 0$ implies $q(f_0, f_1) = 0$ almost everywhere (a.e.), so $f_0 = 0$ a.e. and $f_1 = 0$ a.e., by positive definiteness of β. □

Suppose we already have a generalized multiresolution analysis in $L^2(\mu)$, given by the sequence

$$\begin{aligned} &V_0 \subseteq V_1 \subseteq \dots \\ &V_{i+1} = V_i \oplus W_i \text{ and } W_i \perp V_i \\ &L^2(\mu) = \overline{\bigcup V_i} = V_0 \oplus W_0 \oplus W_1 \oplus \dots \end{aligned} \tag{20}$$

Then we have the following

Theorem 8 *The scalar product (19) is compatible with the direct sum topology of $L^2(\mu) \oplus L^2(\mu)$. If we let*

$$\widetilde{V_i} = V_i \oplus V_i \subset L^2(\mu) \oplus L^2(\mu) \textit{ and } \widetilde{W_i} = W_i \oplus W_i, \tag{21}$$

then the following is a generalized multiresolution analysis:

$$\begin{aligned} &\widetilde{V_0} \subseteq \widetilde{V_1} \subseteq \dots \\ &\widetilde{V_{i+1}} = \widetilde{V_i} \oplus \widetilde{W_i} \textit{ and } \widetilde{W_i} \perp_\iota \widetilde{V_i} \\ &L^2(\mu) \oplus L^2(\mu) = \overline{\bigcup \widetilde{V_i}} = \widetilde{V_0} \oplus \widetilde{W_0} \oplus \widetilde{W_1} \oplus \dots \end{aligned} \tag{22}$$

Proof: Clearly $\bigcup \widetilde{V}_i$ is dense in $L^2(\mu) \oplus L^2(\mu)$ because $\bigcup V_i$ is dense in $L^2(\mu)$. Let (a_{ij}) be the coordinate (2×2)-matrix of β. If $f_0, f_1 \in W_i$ and $g_0, g_1 \in V_i$, then $\iota((f_0, f_1), (g_0, g_1)) = \int \sum a_{ij} f_i(\varphi) g_j(\varphi) d\mu(\varphi) = 0$ because $V_i \perp W_i$. This implies $\widetilde{V}_i \perp_\iota \widetilde{W}_i$ and the orthogonal direct sum decomposition $L^2(\mu) \oplus L^2(\mu) = \widetilde{V}_0 \oplus \widetilde{W}_0 \oplus \widetilde{W}_i \ldots$

Let π_i denote the orthogonal projection $L^2(\mu) \to V_i$ and $\widetilde{\pi}_i$ denote the orthogonal projection onto $\widetilde{V}_i$. It is now clear that $\widetilde{\pi}_i(f_0, f_1) = (\pi_i(f_0), \pi_i(f_1))$. We have $(f_{0n}, f_{1n}) \to (f_0, f_1) \iff f_{0n} \to f_0$ and $f_{1n} \to f_1$, so the topology defined by ι coincides with the topology of the direct sum. □

Corollary 9 *Approximation of a developable surface in the sense that the support functions of both contour lines are chosen from the spline space V_i of trigonometric spline functions such that (17) is minimal, is done by approximating each of the support functions of the contour lines separately in the sense of $L^2(\mu)$.*

Proof: The proof of the previous theorem shows that the ι-orthogonal projection $\widetilde{\pi}_i$ onto $\widetilde{V}_i$ coincides with (π_i, π_i), where π_i is the orthogonal projection onto V_i. □

Acknowledgements

This work has been supported in part by grant No. P12252-MAT of the Austrian Science Foundation. The approximation and multiresolution algorithms have been implemented by M. Pfeifer and N. Pomaroli.

References

[1] G. Aumann, G., Interpolation with developable Bézier patches, Comput. Aided Geom. Design **8** (1991) , 409–420.

[2] G. Aumann, A closer look at developable Bézier surfaces, J. of Theoretical Graphics and Computing **7** (1994) , 12–26.

[3] R.M.C. Bodduluri, B. Ravani, Geometric design and fabrication of developable surfaces, ASME Adv. Design Autom. **2** (1992) , 243–250.

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

[5] J. Chalfant, T. Maekawa, Design for manufacturing using B-spline developable surfaces, preprint, MIT 1997.

[6] J. Chalfant, T. Maekawa, Computation of inflection lines and geodesics on developable surfaces, preprint, MIT 1997.

[7] G. Farin, *Curves and Surfaces for Computer Aided Geometric design*, Academic Press, Boston 1992.

[8] W.H. Frey, D. Bindschadler, Computer aided design of a class of developable Bézier surfaces, General Motors R&D Publication 8057 (1993) .

[9] J. Hoschek, Dual Bézier curves and surfaces, in: *Surfaces in Computer Aided Geometric Design*, R.E. Barnhill and W. Boehm, eds., North Holland, 1983, pp. 147–156.

[10] J. Hoschek, D. Lasser, *Fundamentals of Computer Aided Geometric Design*, A. K. Peters, Wellesley, Mass. 1993.

[11] J. Hoschek, H. Pottmann, Interpolation and approximation with developable B–spline surfaces, in *Mathematical Methods for Curves and Surfaces*, M. Dæhlen, T. Lyche and L.L. Schumaker, eds., Vanderbilt University Press, Nashville 1995, pp. 255–264.

[12] J. Hoschek, M. Schneider, Interpolation and approximation with developable surfaces, in: *Curves and Surfaces with Applications in CAGD*, A. Le Méhauté, C. Rabut and L.L. Schumaker, eds., Vanderbilt University Press, Nashville 1997, pp. 185–202.

[13] J. Hoschek, U. Schwanecke, Interpolationand approximation with ruled surfaces, in: *The Mathematics of Surfaces VIII*, R. Cripps Ed., Information Geometers, 1998, pp. 213–231.

[14] J. Lang, O. Röschel, Developable $(1, n)$-Bézier surfaces, Comput. Aided Geom. Design **9** (1992) , 291–298.

[15] T. Lyche, L.L. Schumaker, L-Spline Wavelets, in: *Wavelets: Theory, Algorithms, and Applications* C. Chui, L. Montefusco and L. Puccio Eds., Academic Press, 1994, pp. 197–212.

[16] M. Neamtu, H. Pottmann, L.L. Schumaker, Dual focal splines and rational curves with rational offsets, *Mathematical Engineering in Industry*, to appear.

[17] Yu. Nesterov, A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM, 1994.

[18] L. Piegl, W. Tiller, *The NURBS book*, Springer, 1995.

[19] H. Pottmann, Studying NURBS curves and surfaces with classical geometry, in *Mathematical Methods for Curves and Surfaces*, M. Dæhlen, T. Lyche and L. L. Schumaker, eds., Vanderbilt University Press, Nashville 1995, pp. 413–438.

[20] H. Pottmann, G. Farin, Developable rational Bezier and B–spline surfaces, Comput. Aided Geom. Design **12** (1995) , 513–531.

[21] T. Randrup, Approximation by cylinder surfaces. Technical Report No. 41, Institut für Geometrie, Vienna University of Technology 1997.

[22] M. Schneider, Approximation von Blechhalterflächen mit Torsen in B–Spline Darstellung, Diplomarbeit, TH Darmstadt 1996.

[23] E. Stollnitz, T. DeRose, H. Salesin, *Wavelets for Computer Graphics*, Morgan Kaufmann Publishers, San Francisco 1996

[24] R. Vatter, Approximation von Datenpunkten mit Torsen in B–Spline Darstellung, Diplomarbeit, TH Darmstadt 1996.

[25] J. Wallner, Generalized multiresolution analysis for arc splines, in: *Mathematical Methods for Curves and Surfaces II*, M. Dæhlen, T. Lyche, L. L. Schumaker (eds.), Nashville University Press, 1998. pp. 537–544