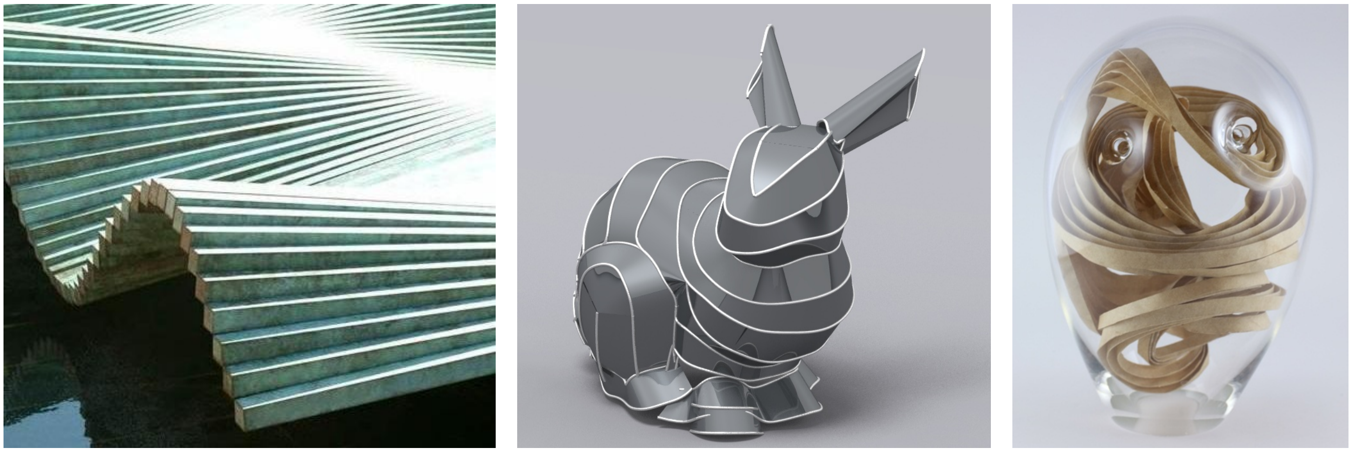
Ruled surfaces and developable surfaces

Johannes Wallner, Graz University of Technology



Ruled surfaces and developable surfaces: The "waves" sculpture by Santiago Calatrava — The bunny approximated by a piecewise-developable surface — a curved-folding design Erik and Martin Demaine.

Contents

1 Ruled surfaces and developable surfaces

1A Representations of ruled surfaces

Primal representation. Ruled surfaces are traced out by the movement of a straight line through space, and they are usually described by a correspondence between parametric curves $\mathbf{a}(u)$ and $\mathbf{b}(u)$: The parametric description of a ruled surface is

$$\begin{aligned}\mathbf{x}(u,v) &= (1-v)\mathbf{a}(u) + v\mathbf{b}(u)\\ &= \mathbf{a}(u) + v\mathbf{r}(u), \quad \text{where} \quad \mathbf{r}(u) = \mathbf{b}(u) - \mathbf{a}(u).\end{aligned}$$

For fixed u, the expression $\mathbf{x}(u,v)$ describes the straight line which connects points $\mathbf{a}(u)$ and $\mathbf{b}(u)$. It is called a *ruling*. This representation of ruled surfaces is referred to as the primal one, in order to distinguish it from the dual representation which is defined later.

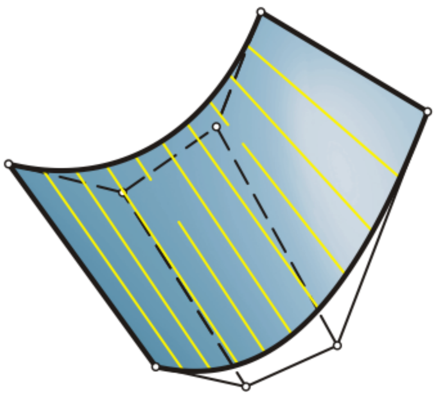


FIGURE 1.1: *Ruled surface defined by the correspondence between two curves* $\mathbf{a}(u)$, $\mathbf{b}(u)$.

Much of the behaviour of ruled surfaces is governed by the rotation of the tangent plane as one moves along a ruling. Either that motion is a complete rotation, or there is no rotation at all. This fact is responsible for the modeling capabilities, and the modeling restrictions a designer is faced with.

Lemma 1.1 *Consider the ruling $R(u) = \mathbf{a}(u) \vee \mathbf{b}(u)$ of the surface defined by the correspondence between curves $\mathbf{a}(u)$, $\mathbf{b}(u)$.*

1. *If the tangent plane of the surface is different in 2 different points of $R(u)$, it rotates about 180 degrees when $R(u)$ is traversed along its entire length (there are no singular points on $R(u)$).*
2. *If the tangent plane is the same in 2 different points of $R(u)$, it is the same for all points of $R(u)$ (there may be 1 singular "regression" point without tangent plane on $R(u)$).*
3. *If two points are without tangent plane, the entire ruling is.*

These three cases are characterized by the vectors

$$\dot{\mathbf{a}}, \dot{\mathbf{b}}, \mathbf{b}-\mathbf{a}, \quad \textit{or equivalently,} \quad \dot{\mathbf{a}}, \dot{\mathbf{r}}, \mathbf{r}$$

spanning a subspace of dimensions 3, 2, *and* 1 *respectively.*

Proof: A normal vector of the surface is computed by

$$\mathbf{x}_u \times \mathbf{x}_v = (\dot{\mathbf{a}} + v\dot{\mathbf{r}}) \times \mathbf{r} = \dot{\mathbf{a}} \times \mathbf{r} + v\dot{\mathbf{b}} \times \mathbf{r}.$$

Either the two vectors $\dot{\mathbf{a}} \times \mathbf{r}$, $\dot{\mathbf{b}} \times \mathbf{r}$ are both zero (case 3) or are parallel (case 2) or are not parallel (case 1). Correspondingly the normal vector is zero (case 3), or does not change its direction but may vanish for 1 value of v (case 2) or rotates by 180 degrees without ever vanishing (case 1). Q.E.D

FIGURE 1.2: *Sculpture by Santiago Calatrava. One can clearly see that when a point progresses along a ruling of a ruled surface, the tangent plane in that point rotates abou the ruling. The total rotation is 180 degrees.*

Definition 1.2 *A ruled surface where all rulings have only 1 single tangent plane is called a torse, or a developable ruled surface.*

Example 1.3 *General cylinders and cones are torses, and so are the surfaces traced out by the tangents of a space curve* $\mathbf{a}(u)$.

The easy proofs of these statements have been relegated to Exercises 1.1 and 1.5 (page 6).

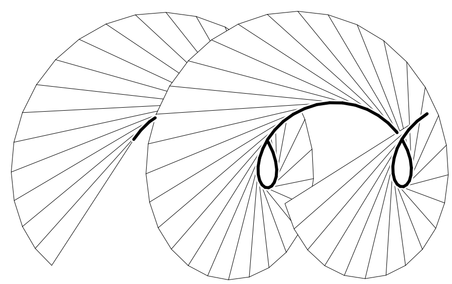


FIGURE 1.3: *Tangent surfaces: The surface traced out by the tangents of a curve $\mathbf{c}(u)$ is a developable ruled surface. The curve itself is a sharp edge on the surface. Here only one half of each tangent is shown.*

Dual representation. Another way of describing a time-dependent straight line $R(u)$ is via the envelope of a moving plane $T(u)$:

$$T(u) \quad \ldots \quad \mathbf{n}^\top \mathbf{x} + n_0 = 0 \quad \text{where} \quad \mathbf{n} = (n_1, n_2, n_3)$$

and each of $n_0, \ldots, n_3$ is a function of u. Since the intersection of two successive planes $T(u)$ and $T(u+h)$ is a straight line, this is also true for the limit $h \to 0$, and it is not difficult to compute the rulings of the surface enveloped by the family $T(u)$:

$$R(u) = \lim_{h\to 0} T(u) \cap T(u+h).$$

That ruling is not difficult to compute, since the condition that $\mathbf{x} \in T(u) \cap T(u+h)$ can be modified as follows:

$$\mathbf{n}(u)^\top \mathbf{x} + n_0(u),\ \mathbf{n}(u+h)^\top \mathbf{x} + n_0(u+h) \iff$$

$$\iff \begin{cases} \mathbf{n}(u)^\top \mathbf{x} + n_0(u), \\ \left(\dfrac{\mathbf{n}(u+h) - \mathbf{n}(u)}{h}\right)^\top \mathbf{x} + \dfrac{n_0(u+h) - n_0(u)}{h} = 0. \end{cases}$$

The limit $h \to 0$ now yields the conditions

$$R(u) \ \ldots \ \mathbf{n}^\top \mathbf{x} + n_0 = \dot{\mathbf{n}}^\top \mathbf{x} + \dot{n}_0 = 0.$$

The ruling $R(u)$ is the common solution of these two equations. The vector indicating the direction of the ruling is accordingly computed as

$$\mathbf{r} = \mathbf{n} \times \dot{\mathbf{n}}.$$

The surface traced out by the lines $R(u)$ is ruled, and obviously the tangent plane of the surface along the entire ruling is the plane $T(u)$. The representation of ruled surfaces as an evelope of planes works only for torses (developable ruled surfaces), and is called the dual represention.

Discrete ruled surfaces. Both for computations and for discrete theories (e.g. discrete differential geometry) it makes sense to study discrete representations of ruled surfaces. While a general ruled surface is simply a sequence of lines or a sequence of straight line segments, the condition of developability is best expressed by requiring that successive lines or successive line segments are co-planar (see Figure 1.4).

Figure 1.5 shows a discrete model of a developable ruled surface. It suggests properties which are known to be true for continuous developable surfaces, namely the existence of a curve $\mathbf{c}(u)$ of singular points, and the fact that the tangents of that curve are exactly the rulings of the developable surface.

1B Intrinsically flat surfaces

Developable surfaces constitute a class of surfaces whith many interesting properties relevant to different kinds of applications. Unfortunately the mathematical statements which express the relations between these defining properties are complicated. Developable surfaces are notorious for statements which are not true in the mathematical sense but are nevertheless true for all practical purposes.

We have already defined developability as a special property of ruled surfaces. This word comes from the fact that such developables can be unfolded into the plane without stretching or tearing, and in a manner of speaking, also the converse statement is true. This unfoldability is the more literal meaning of "developable". It is however convenient to require this property only in a weaker sense, because we want to be able to call cylinders developable, and a cylinder can only be unfolded if it is first cut open.

Definition 1.4 *A surface is intrinsically flat ("developable" in the literal sense), if every point has a neighbourhod which can be mapped to a planar domain in an isometric way, meaning that curves within the surface do no change their length.*

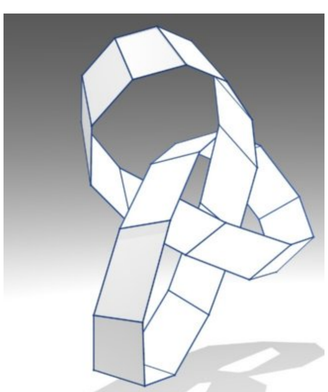


FIGURE 1.4: *A discrete torse is formed by a sequence of line segments such that each segment and its successor are coplanar.*

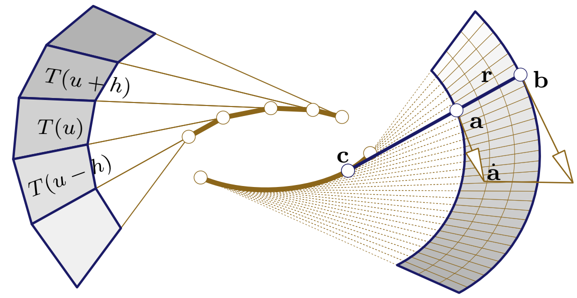


FIGURE 1.5: *Developable ruled surface defined as the envelope of a family $T(u)$ of planes. The rulings occur as limits of $T(u) \cap T(u+h)$ as $h \to 0$ (i.e., a ruling is the intersection of infinitesimally close planes). The points of regression $\mathbf{c}(u)$ occur as limits of $T(u-h) \cap T(u) \cap T(u+h)$ as $h \to 0$ (i.e, a regression point is the intersection of 3 infinitesimally close planes, or 2 infinitesimally close rulings).*

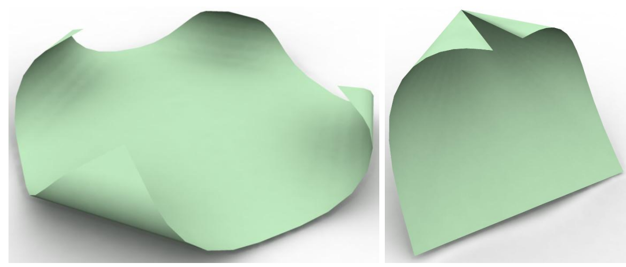


FIGURE 1.6: *Surfaces created by isometric bending of a rectangular sheet of paper (images by Solomon et al.* [2012]*). The left hand surface consists of a planar part and 4 individual ruled parts.*

By gluing 2 opposite edges of a rectangle together we obtain a metric space which is isometric to a right circular cylinder; by cutting a right circular cylinder along a ruling yields a surface which can be isometrically mapped to a rectangle. Therfore the right circular cylinder is an intrisically flat surface.

One can also glue together the remaining 2 opposite edges of a cylinder and ask the question if there exists a surface in 3-space which is isometric to this intrinsically flat Riemannian manifold. This question was answered affirmative by John Nash via his famous embedding theorem:

FIGURE 1.7: *The cylindrical part of a tin can is a right circular cylinder, and therefore intrinsically flat. This property is not lost upon isometric deformation.*

Theorem 1.5 (J. Nash 1954) *If M is an m-dimensional Rieman-*

nian manifold, then there is a C^1 surface in $\mathbb{R}^n$ isometric to M, provided $n > m$ and there is a surface in $\mathbb{R}^n$ diffeomorphic to M

One could attempt to create such a "flat torus" by bending a cylinder such that its two circular boundaries come together. In practice attempts to produce a smooth surface with this property do not succeed (Figure 1.7). Only recently an explicit smooth flat torus was given (Figure 1.8). Note that a polyhedral flat torus is easy to create (Figure 1.9).

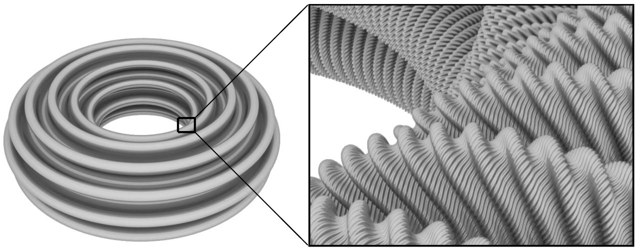


FIGURE 1.8: *A flat torus. From afar it looks like a torus with "waves" on it. A closer look reveals that the waves have waves which themselves have waves and so on, ad infinitum. Borrelli et al.* [2012] *constructed this surface recursively and showed C^1 smoothness of the limit.*

Theorem 1.6 (Hartman and Winter 1950, Theorem 4) *A C^2 surface is intrisically flat $\iff$ its Gauss curvature vanishes.*

If higher smoothness is assumed, this theorem is part of the usual differential and Riemannian geometry courses. By manipulating the known formulae regarding the first and second fundamental forms one finds out that the Gauss curvature

$$K = \frac{\det(\text{2nd fundamental form})}{\det(\text{1st fundamental form})}$$

can also be expressed in terms of the 1st fundamental form alone. It follows that isometric mappings do not change Gauss curvature. Consequently if an isometric mapping to the plane exists, the Gaussian curvature must equal the plane's Gaussian curvature, i.e., zero.

The reverse implication is usually proved by considering parallel transport along curves, which for general surfaces depends on the curve, but for flat surfaces does not. Relations between Gauss curvature, the Riemann curvature tensor, and parallel transport eventually yield the result.

Example 1.7 *General cylinders and cones are intrinsically flat, and so are surfaces traced out by the tangents of a space curve.*

Proof: We show only the 3rd statement. Consider a curve $\mathbf{a}(u)$ traversed with unit speed, i.e., $\|\dot{\mathbf{a}}\| = 1$. The curvature $\kappa(u)$ of $\mathbf{a}(u)$ obeys $\ddot{\mathbf{a}}(u) = \kappa(u)\mathbf{e}_2(u)$, where $\mathbf{e}_2(u)$ is the principal normal vector field. The tangents of the curve form the surface

$$\mathbf{x}(u, v) = \mathbf{a}(u) + v\dot{\mathbf{a}}(u).$$

For any curve $\mathbf{c}(t)$, its length is given by $\int \|\dot{\mathbf{c}}\|\, dt$. Assume that $\mathbf{x}(u, v)$ is a surface and $\mathbf{c}(t) = \mathbf{x}(u(t), v(t))$ is a curve in it. From

$$\dot{\mathbf{c}} = \mathbf{x}_u\dot{u} + \mathbf{x}_v\dot{v}, \quad \|\dot{\mathbf{c}}\|^2 = \dot{u}^2\mathbf{x}_u\mathbf{x}_u + 2\dot{u}\dot{v}\mathbf{x}_u\mathbf{x}_v + \dot{v}^2\mathbf{x}_v\mathbf{x}_v$$

we see that its length can be computed by knowing $u(t), v(t)$ and the scalar products $\mathbf{x}_u \cdot \mathbf{x}_u, \dots$ of partial derivatives of $\mathbf{x}(u, v)$:

$$\mathbf{x}_u = \dot{\mathbf{a}} + v\ddot{\mathbf{a}} = \dot{\mathbf{a}} + v\kappa\mathbf{e}_2, \quad \mathbf{x}_v = \dot{\mathbf{a}} \implies$$
$$\mathbf{x}_u \cdot \mathbf{x}_u = 1 + v^2\kappa^2, \quad \mathbf{x}_u \cdot \mathbf{x}_v = 1, \quad \mathbf{x}_v \cdot \mathbf{x}_v = 1$$

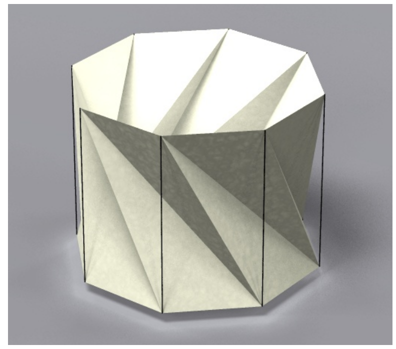


FIGURE 1.9: *A flat polyhedral torus. Developability around vertices follows from the polyhedral Gauss-Bonnet theorem which says that angle defects sum to 0. Since all vertices are equal, all angle sums in vertices equal 2π.*

We see that any curvature-preserving change in $\mathbf{a}$ causes the tangent surface $\mathbf{x}$ to evolve isometrically. Since there is a planar curve which has precisely curvature $\kappa(u)$, we can isometrically map the original tangent surface to a planar domain. Q.E.D

1C Relation between developability and ruledness

It seems to be well known that smooth and intrinsically flat surfaces are ruled, but appearances are deceptive: It is possible that the rulings are not smooth, even if the surface is C^∞ smooth. A precise statement is the following:

Theorem 1.8 (Pogorelov 1969, p. 694f) *If p is a point on a C^2 surface in $\mathbb{R}^3$ whose Gaussian curvature vanishes everywhere, then*

1. *either M contains a neighbourhood of p which lies in a plane;*
2. *or M contains a unique straight line passing through p which ends only at the boundary of M, and the tangent plane is the same in all points of that line.*

This statement implies in particular that the planar parts of the surface are bounded by straight lines which are the boundaries of ruled parts of the surface. E.g. the piece of paper illustrated by Figure 1.6, left, conists of a central planar part bordered by 4 ruled parts.

Theorem 1.8 was also proved by Hartman and Nirenberg [1959] who go on to make (p. 916f) a more precise statement which involves the notion of "planar point", meaning a point of the surface where both principal curvatures vanish.

Proposition 1.9 *Assume a C^2 surface is parametrized over the unit disk as parameter domain, and non-planar points lie still dense in this domain. Then the surface is equivalently described by a torsal ruled surface $\mathbf{x}(u, v) = \mathbf{a}(u) + v\mathbf{r}(u)$. Further,*

1. *If there are no planar points, $\mathbf{r}(u)$ enjoys C^1 smoothness.*
2. *For each planar point there is an entire ruling of planar points. $\mathbf{r}(u)$ is continuous but in general not smooth.*

The two theorems above are usually summarized as:

Folklore Statement 1.10 *Surfaces which are intrisically flat (and which have zero Gaussian curvature) are torsal ruled surfaces.*

This statement is true only for C^2 surfaces, and only if the surface has no planar parts (otherwise there may be several ruled parts). Furthermore, the ruled surface associated with an intrinsically flat surface might be non-smooth. Another common knowledge statement is the following, which is illustrated by Figure 1.10.

Folklore Statement 1.11 *Developable surfaces can be decomposed into planar parts, cylinders, cones, and tangent surfaces (which are swept by the tangents of a space curve).*

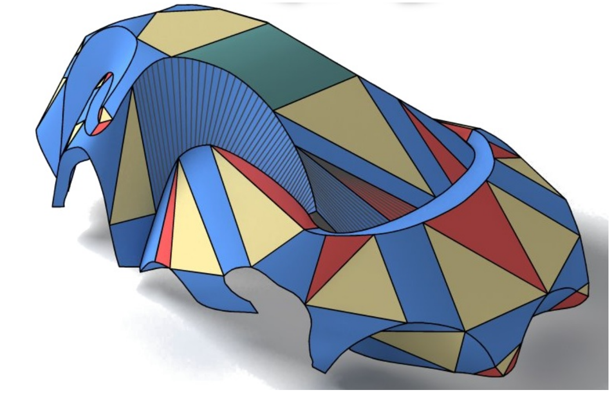


FIGURE 1.10: *A developable car designed by Gregory Epps. It is a piecewise-smooth surface and its decomposition into planar, cylindrical, conical and general tangent-surface-type developables is indicated by colors (image taken from* [Kilian et al. 2008]*).*

The wording “developable” already assumes the equivalence of “intrinsically flat surfaces” and “torsal ruled surfaces”. The precise statement is as follows:

Proposition 1.12 *Consider a C^2 ruled surface parametrization of a flat surface with rulings $R(u)$. Each open interval I on the u parameter line contains an open interval J such that for all $u \in J$ one of the following applies:*

1. *rulings $R(u)$ are pallel (cylinder case)*
2. *rulings $R(u)$ pass through a common point (cone case)*
3. *rulings $R(u)$ are the tangents of a curve $\mathbf{c}(u)$.*

Each of these surface may have the additional property that it is contained in a plane.

Proof: Consider a parametrization $\mathbf{x}(u, v) = \mathbf{a}(u) + v\mathbf{r}(u)$ where $\mathbf{r}$ is a unit vector field. If all rulings are parallel in I, then we have the cylinder case. Otherwise, since $\dot{\mathbf{r}}$ is continuous, there is an interval $J \subseteq I$ where $\dot{\mathbf{r}} \neq \mathbf{o}$. Comparing with the formulae in the proof of Lemma 1.1 we see that on each ruling there is a singular point $\mathbf{c}(u) = \mathbf{x}(u, v^*(u))$ where $\mathbf{x}_u, \mathbf{x}_v$ are parallel. Either $\mathbf{c}(u)$ is constant, then all rulings pass through that point, and we are in the cone case. Otherwise, since $\dot{\mathbf{c}}$ is continuous, there is some interval J where $\dot{\mathbf{c}}(u) \neq \mathbf{o}$. From $\dot{\mathbf{c}} = \mathbf{x}_u + \mathbf{x}_v \dot{v}^*$ we see that $\dot{\mathbf{c}}$ and $\mathbf{x}_v$ are parallel, so the ruling $R(u)$ is a tangent of the curve $\mathbf{c}(u)$. Q.E.D

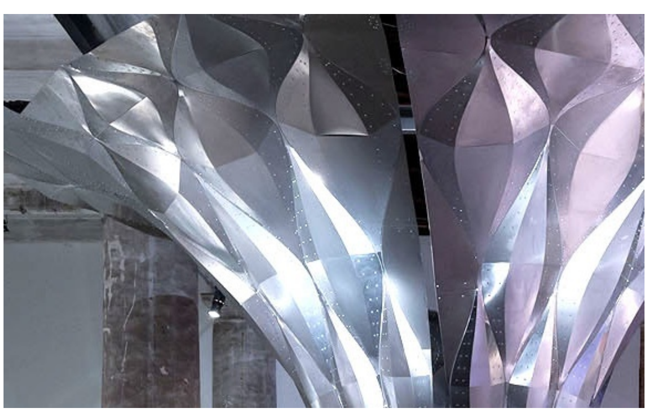


FIGURE 1.11: *The “Arum” surface was designed by Zaha Hadid Architects in cooperation with Robofold for the 2012 Venice Biennale. It consists of metal sheets folded along curved creases.*

1D Developables with creases

The folding of geometric objects from paper is an ancient subject of great interest and beauty, and even folding paper along curved creases goes back to the 1927 Bauhaus. The surfaces which occur in this way are intrinsically flat, but only piecewise-smooth. Origami is not the only “application” of folding along curved creases. Others are design and architecture (see Figure 1.11 for an installation by Zaha Hadid) and production processes (see Figure 1.12). We also point to curved-crease sculptures by G. Epps (Figure 1.10 and also 1.11) and M. and E. Demaine (Figure 1.13) and mention that the existence of such surfaces in the mathematical sense in some cases is an open problem [Demaine et al. 2011]. We return to the topic of modeling developables with creases in §4F.

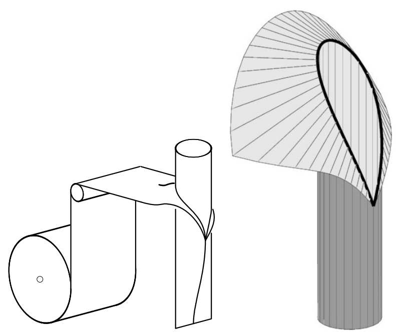


FIGURE 1.12: *Sketch of a packaging machine and closeup of an ideal shoulder surface, which is developable with a crease in it.*

FIGURE 1.13: *Sculpture by Erik and Martin Demaine created by folding an annulus along concentric circles.*

Exercises to §1

1.1. Show that cylinders and cones are developable surfaces. Give primal representations by curves $\mathbf{a}(u), \mathbf{b}(u)$, and dual representations by planes $T(u)$.

1.2. Give an explicit representation of a ruled Möbius strip.

1.3. Consider the hyperbola in the xy plane which is given by the equation $\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$. Rotating that hyperbola about the y axis yields a surface of revolution whose equation is given by

$$\frac{x^2+z^2}{a^2} - \frac{y^2}{b^2} = 1.$$

Show that it is a ruled surface. *Hint:* Intersect with the tangent plane "$x = a$" (see Figure 1.15).

1.4. Conversely, show that rotating a straight line about an axis creates a hyperboloid (see Fig. 1.15).

1.5. Show that the ruled surface traced by the tangents of a space curve $\mathbf{c}(u)$ is developable. Give a primal representation.

1.6.* Show that surfaces which have constant slope α w.r.t. a horizontal reference plane are actually developable ruled surfaces – see Figure 1.16. *Hint:* The surface is graph of the function $\phi(x, y)$. Consider the curves of steepest descent defined by $\binom{\dot{x}}{\dot{y}} = \nabla\phi(x, y)$. They are straight because $\|\nabla\phi\| = \alpha$ eventually implies $\binom{\ddot{x}}{\ddot{y}} = \binom{0}{0}$; the surface normal vector along these curves is constant because $\nabla\phi$ is.

1.7. Show: A surface is developable $\iff$ all apparent contours, for whatever camera/eye position, are straight lines.

1.8. Check if the outer surfaces of the Los Angeles "Disney concert hall" (Figure 1.17) are developable ruled surfaces, and answer the same question for the the ruled Möbius strip of Figure 1.14. *Hint:* The first question is more difficult to answer and requires studying several images.

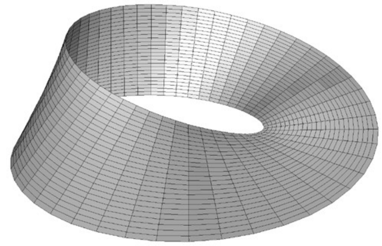


FIGURE 1.14: *Ruled Möbius strip.*

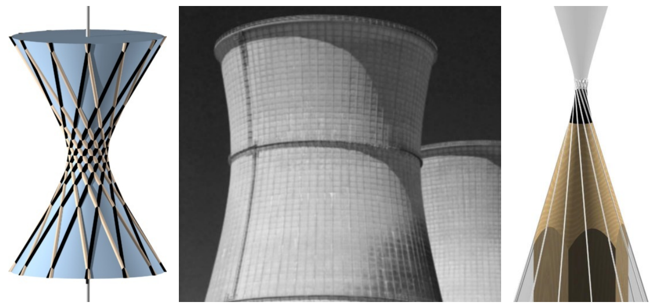


FIGURE 1.15: *Left: Hyperboloid with rulings. Center: Hyperboloids occur naturally in cooling towers, since the broader base, narrower waist, and widening on top is necessary for proper functioning of the tower, and the ruled property allows us to use straight elements for building. Right: Hyperboloids occur when sharpening a pencil with a misaligned pencil sharpener.*

FIGURE 1.16: *Spoil tip of the Heringen potash mine (Germany), and the chapel of S. Benedetg (Sumvigts, upper Rhine, Switzerland). These constant-slope surfaces, hence developable.*

FIGURE 1.17: *Disney Concert Hall, Los Angeles.*

* Starred exercises need more mathematics than others.

2 Modeling with ruled surfaces

It is easy to model ruled surfaces using the tools available in geometric design, since Bézier curves and also B-spline curves can easily be made straight. Even if it is not easy to interactively model developable surfaces, splines are still very useful for that purpose. Generally, splines are important tools in geometric modeling. Their main purpose is to approximate the potentially infinite-dimensional manifold of curves and surfaces by a finite-dimensional set of spline curves and spline surfaces – each of these special curves and surfaces is described by a finite number of control points. In this way the shape of curves and surfaces can be described by a finite number of unknowns and becomes computationally accessible.

2A B-spline curves and surfaces

Definition 2.1 *Assume that a finite sequence of real numbers, called "knots" $u_0 \leq u_1 \leq u_2 \leq \ldots$ and a polynomial degree d are given. We require that the multiplicity of knots is $d+1$ for the first and last knot, and that otherwise it does not exceed $d+1$.*

Then the spline space defined by these data consists of all functions which are polynomial of degree $\leq d$ within each subinterval $[u_i, u_{i+1})$, and which enjoy C^{d-m} continuity at every knot of multiplicity m.

A curve $\mathbf{c}(u)$ whose component functions are elements of the spline space is called a spline curve.

It is not difficult to draw the graphs of functions which belong to a certain spline space determined by knots u_i and smothnesses k_i, see Figure 2.1.

The theory of splines is very much developed, and one of the basic facts is how to compute the so-called B-spline basis functions

$$N_0(u),\ N_1(u),\ \ldots,$$

of a certain spline space. For degree 0 these functions are piecwise-constant; for degree 1 they are piecewise-linear, and so on. They have minimal possible support. Instead of printing a theorem here, we simply show some example, see Figure 2.1. A spline curve is a linear combination of spline basis functions:

$$\mathbf{c}(u) = \sum \mathbf{c}_i N_i(u),$$

where $\mathbf{c}_1, \ldots$ are called the control points. A *ruled surface* defined by two spline curves $\mathbf{a}(u)$, $\mathbf{b}(u)$, can be seen in Figure 2.2.

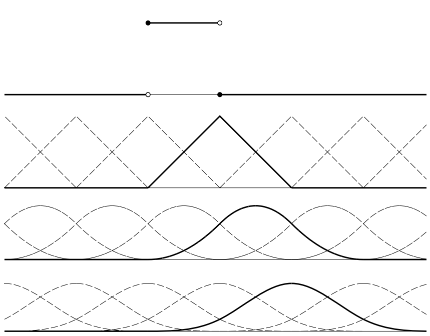


FIGURE 2.1: *Sample Basis functions of spline spaces. From top to bottom: degrees 0, 1, 2, 3, with respective requirements of no continuity, continuity, C^1 smoothness, and C^2 smoothness.*

We mention de Boor's recursive algorithm for evaluating spline curves, which can be seen as an alternative and constructive definition of a spline curve by specifying how it depends on the knots and control points:

Theorem 2.2 (de Boor's algorithm) *Assume that a degree d and an admissible knot sequence $\{u_j\}$ is given, and a spline curve $\mathbf{c}(u)$ is defined by these data and control points $\{\mathbf{c}_j\}$. To evaluate $\mathbf{c}(u)$, for $u \in [u_l, u_{l+1})$, we let $\mathbf{c}_i^0 := \mathbf{c}_i$ and recursively compute*

$$\mathbf{c}_i^r = (1-\alpha_i^r)\mathbf{c}_{i-1}^{r-1} + \alpha_i^r \mathbf{c}_i^{r-1}, \text{ where } \alpha_i^r = \frac{u - u_i}{u_{i+d+1-r} - u_i},$$

for $i = l-d, \ldots, l$. Then $\mathbf{c}(u) = \mathbf{c}_l^d$.

The curve's tangent (resp. osculating plane) curve is spanned by $\mathbf{c}_l^{d-1}, \mathbf{c}_{l-1}^{d-1}$ (resp., $\mathbf{c}_l^{d-2}$, $\mathbf{c}_{l-1}^{d-2}$, $\mathbf{c}_{l-2}^{d-2}$) which are computed during the recursion.

For a proof see e.g. [de Boor 1978]. The statement about the tangent and osculating plane is illustrated by Figure 2.2. An important case is the following:

Example 2.3 *The knot sequence $0, 0, 1, 1$ yields linear spline curves, with 2 control points $\mathbf{a}_0$ and $\mathbf{a}_1$, which are evaluated as*

$$\mathbf{a}(v) = (1-v)\mathbf{a}_0 + v\mathbf{a}_1.$$

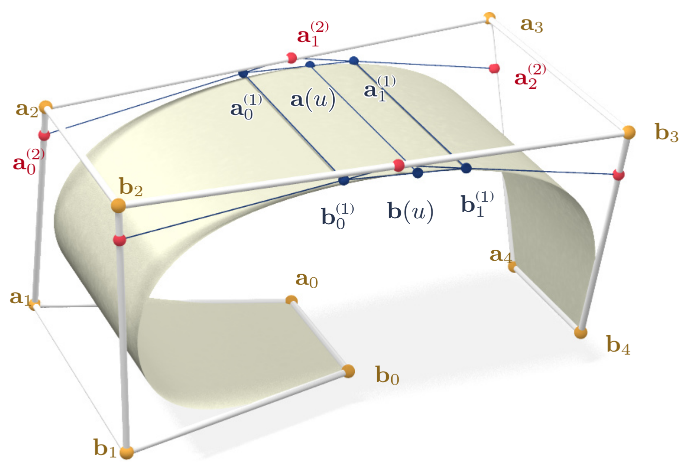


FIGURE 2.2: *A spline curve $\mathbf{a}(u)$ defined by control points $\mathbf{a}_0$, $\mathbf{a}_1, \ldots$ is equipped with auxiliary first derivative points $\mathbf{a}_0^{(1)}(u)$, $\mathbf{a}_1^{(1)}(u)$ which span the tangent, and second derivative points $\mathbf{a}_0^{(2)}(u)$, $\mathbf{a}_1^{(2)}(u)$, $\mathbf{a}_2^{(2)}(u)$ which span the osculating plane of the curve. These auxiliary points are computed with de Boor's algorithm. Developability of the ruled surface defined by curves $\mathbf{a}$, $\mathbf{b}$ is equivalent to coplanarity of $\mathbf{a}_0^{(1)}$, $\mathbf{a}_1^{(1)}$, $\mathbf{b}_0^{(1)}$, $\mathbf{b}_1^{(1)}$.*

Spline surfaces. The well known "tensor product" spline surfaces are well suited for modelling with ruled surfaces.

Definition 2.4 *Having chosen two knot sequences and having established two bases $N_1(u), N_2(u), \ldots$ and $\bar{N}_1(v), \bar{N}_2(v), \ldots$, any surface*

$$\mathbf{x}(u,v) = \sum_{i,j} \mathbf{x}_{ij} N_i(u) \bar{N}_j(v)$$

is a "tensor product" spline surface. It can be seen as a B-spline in the variable u with v-dependent control points:

$$\mathbf{x}(u,v) = \sum_i \underbrace{\Big(\sum_j \mathbf{x}_{ij} N_j(v)\Big)}_{\mathbf{x}_i^*(v)} N_i(u)$$

or as a B-spline in the variable v with u-dependent control points:

$$\mathbf{x}(u,v) = \sum_j \Big(\underbrace{\sum\nolimits_i \mathbf{x}_{ij} N_i(u)}_{\mathbf{x}_j^*(u)}\Big) N_j(v).$$

Its evluation therefore repeatedly calls de Boor's algorithm.

By choosing a knot sequence for the variable u and the special knot sequence $0, 0, 1, 1$ fore the variable v, we obtain ruled B-spline surfaces. Choose the spline control points $\{\mathbf{a}_i\}$ and $\{\mathbf{b}_i\}$ of of two curves $\mathbf{a}(u)$ and $\mathbf{b}(u)$. The rectangular arrangement

$$\begin{matrix} \mathbf{a}_0 & \mathbf{a}_1 & \cdots & \mathbf{a}_l \\ \mathbf{b}_0 & \mathbf{b}_1 & \cdots & \mathbf{b}_l \end{matrix}$$

defines the control points of a B-spline surface which evaluates to

$$\begin{aligned} \mathbf{x}(u,v) &= (1-v)\Big(\sum\nolimits_i \mathbf{a}_i N_i(u)\Big) + v\Big(\sum\nolimits_i \mathbf{b}_i N_i(u)\Big) \\ &= (1-v)\mathbf{a}(u) + v\mathbf{b}(u). \end{aligned}$$

This is the familiar representation of ruled surfaces. Figures 2.2 and 2.3 show examples.

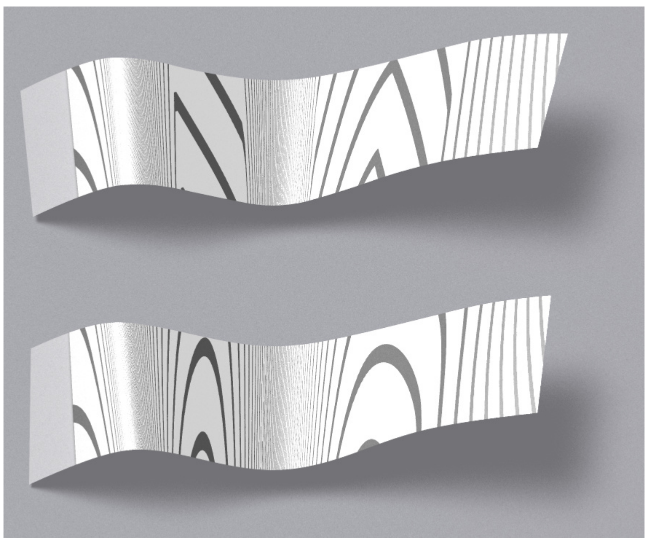


FIGURE 2.3: *The smoothness of a surface can be visualized using reflection lines. These two ruled B-spline surfaces are piecewise quadratic/cubic, they C^1/C^2 smoothness, and reflections in them are continuous/C^1.*

2B Modeling ruled surfaces using optimization

When building up larger systems of geometric primitives which fit together in certain ways it frequently happens that the shape and position of primitives are described by a large number of unknowns, while the constraints are expressed by an equally large number of equations. In many cases it is not possible to solve the system of constraints directly, and one has to resort to iterative and approximate methods. There are, however, certain cases where this is possible, and we are going to discuss one of them here.

Proposition 2.5 *A quadratic function $f : \mathbb{R}^n \to \mathbb{R}$ defined by*

$$f(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x} + 2\mathbf{b}^\top \mathbf{x} + \gamma$$

is stationary exactly for such $\mathbf{x}$ which obey

$$A\mathbf{x} + \mathbf{b} = 0.$$

In case f is bounded from below, this linear condition characterizes the minima of f.

Proof: It is elementary to compute the directional derivative $\frac{d}{dt}\big|_{t=0} f(\mathbf{x}+t\mathbf{v}) = 2(A\mathbf{x}+\mathbf{b})^\top\mathbf{v}$. That derivative is zero for all $\mathbf{v}$ precisely if the condition above is fulfilled. The statement about minima follows from the well known classification of quadratic functions. Q.E.D

Typically requirements imposed in real-world geometric modeling are incompatible with each other. In case they are expressed as incompatible linear constraints, the following is extremely useful:

Example 2.6 *In order to solve the over-constrained linear system*

$$A\mathbf{x} = \mathbf{b}$$

in the least squares sense, i.e.,

$$\|A\mathbf{x} - \mathbf{b}\| \to \min,$$

we have to solve

$$A^\top A\mathbf{x} = A^\top \mathbf{b}.$$

Proof: This is a corollary of Prop. 2.5, since $\|A\mathbf{x} - \mathbf{b}\|^2 = \mathbf{x}^\top A^\top A\mathbf{x} - 2\mathbf{b}^\top A\mathbf{x} + \mathbf{b}^\top\mathbf{b}$. Q.E.D

Proposition 2.7 *Consider the optimization problem*

$$f(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x} \to \min$$

under the constraint

$$g(\mathbf{x}) = \mathbf{x}^\top B\mathbf{x} = 1$$

(matrices A, B can be made symmetric without changing f, g by replacing them with $\frac{1}{2}(A + A^\top)$ resp. $\frac{1}{2}(B + B^\top)$). If a solution exists, it is found as follows:

1. *Determine the smallest λ with $\det(A - \lambda B) = 0$.*
2. *Solve the homogeneous linear system $(A - \lambda B)\mathbf{x} = 0$*
3. *Normalize the solutions $\mathbf{x}$ such that $g(\mathbf{x}) = 1$.*

Proof: (Sketch) Recall that a minimum must have the property that gradients ∇f, ∇g are proportional to each other, leading to $A\mathbf{x} - \lambda B\mathbf{x} = \mathbf{o}$. This linear system has a solution $\mathbf{x} \neq \mathbf{o}$ only if $A - \lambda B$ has not full rank, leading directly to the procedure above. As to which solution λ we must take in 1., observe that the conditions 1.–3. imply $f(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x} = \lambda\mathbf{x}^\top B\mathbf{x} = \lambda$. Q.E.D

Example: Choosing variables for composite surfaces. Suppose that we want to model a composite surface which consists of individual ruled pieces, like the ones shown by Figures 2.4 and 2.7. Let us discuss the degrees of freedom available for geometric design for the surface of Figure 2.4. We obviously have 6 ruled surfaces, each being described by the correspondence of two spline curves. We choose individual parameter domains, polynomial degrees (e.g. $d = 3$) and knot vectors, which in the present case leads to 10 control points per surface patch. Patch No. i has control points

$$\mathbf{a}_0^{(i)}, \ldots, \mathbf{a}_5^{(i)}, \qquad \mathbf{b}_0^{(i)}, \ldots, \mathbf{b}_5^{(i)}.$$

Since B-splines have the endpoint-interpolating property (provided boundary knots are chosen with multiplicity $d + 1$) we can make the composite surface continuous by simply identifying some of the control points, thereby reducing the number of variables. In the same way we can achieve other things like two boundary curves of the same ruled surface to meet in a common endpoint.

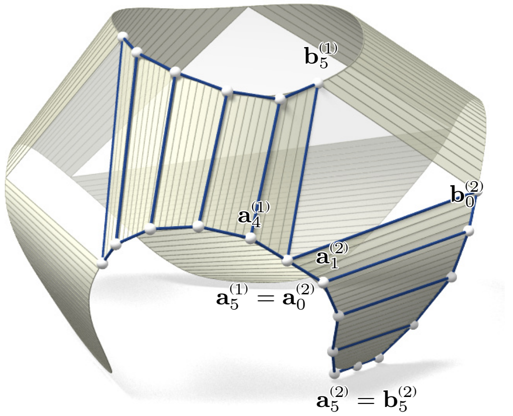


FIGURE 2.4: *Surface consisting of both planar and ruled pieces. Some of the spline control points coincide. Smooth transitions are guaranteed if the spline control points fulfill certain conditions.*

The condition of tangent-plane continuity in all 9 points is a bit more complicated to incorporate in its full generality. Geometrically the condition is simple: Since B-spline curves have the property that the boundary tangents are the first and last segment of the control polygon, all we have to do is to make sure that

$$\{\mathbf{a}_4^{(1)}, \quad \mathbf{a}_5^{(1)} = \mathbf{a}_0^{(2)}, \quad \mathbf{a}_1^{(2)}\} \quad \text{collinear}$$

and the same for 8 other tangent-continuous transitions. Further, we would like to have the property that the ruled surfaces join the planar parts in a smooth manner. Thus we have to require that boundary tangents lie in the plane, e.g. by requiring

$$\{\mathbf{a}_4^{(1)},\, \mathbf{a}_5^{(1)} = \mathbf{a}_0^{(2)},\, \mathbf{a}_1^{(2)},\, \mathbf{b}_0^{(2)},\, \mathbf{b}_1^{(2)},\, \mathbf{b}_5^{(1)},\, \mathbf{b}_4^{(1)}, \ldots\} \text{ co-planar}$$

and 2 more conditions of this kind (see Figure 2.4). Unfortunately these conditions are not linear, but quadratic or cubic depending on our choice of variables. This is problematic. There are two ways out of this dilemma: We can impose more strict conditions (paying the price of fewer degrees of freedom available for modeling) or we use more sophisticated and time-consuming modeling tools. Since we have only simple methods at our disposal at the moment, we opt for reducing the number of degrees of freedom and impose *linear* relations which effectively reduce the number of variables present. We could e.g. replace collinearity by

$$\mathbf{a}_5^{(1)} = \mathbf{a}_0^{(2)} = \frac{1}{2}(\mathbf{a}_4^{(1)} + \mathbf{a}_1^{(2)}),$$

and we can replace co-planarity by

$$\begin{aligned} \mathbf{a}_4^{(1)} &= \mathbf{a}_0^{(2)} - \alpha(\mathbf{b}_0^{(2)} - \mathbf{a}_5^{(1)}) \\ \mathbf{a}_1^{(2)} &= \mathbf{a}_0^{(2)} + \alpha(\mathbf{b}_0^{(2)} - \mathbf{a}_5^{(1)}) \text{ and so on,} \end{aligned}$$

which effectively leaves only the endpoins of control polygons as independent variables and eliminates their immediate neighbours.

Distance functions. In geometric modeling it frequently happens that a spline curve or spline surface has to be close to a reference shape. This constraint can be arbitrarily complex and there is actually much literature on this topic. Real-world applications, as a rule, always involve constraints of that sort. As a typical example we consider the problem of approximating the design shape of the Cagliari musem project (Figure 2.6) by a sequence of ruled surfaces which fit together in a smooth way.

In the following we briefly discuss some standard methods to deal with such proximity constraints. The first step is to understand the distance field of a surface, and to develop methods to evaluate distances and compute closest points. In the present lecture notes there is no room for a systematic discussion. We mention only "fast sweeping" methods which compute distances and closest point projection for all vertices of a voxel grid surrounding the reference shape, e.g. [Tsai 2002].

The next ingredient in our discussion is information on how the distance field of a curve or a surface can be replaced by a simpler function. Recall that locally we can represent a surface M in $\mathbb{R}^3$ as graph of a function. We choose an adapted coordinate system with origin on M, and the x and y axes aligned with the principal directions. Then the surface is the graph of a function $f(x, y)$ which has the Taylor expansion $z = f(x, y) = \frac{1}{2}(\kappa_1 x^2 + \kappa_2 y^2) + \ldots$ The following result gives information on the best local quadratic approximant to the function $\operatorname{dist}(\mathbf{x}, M)^2$.

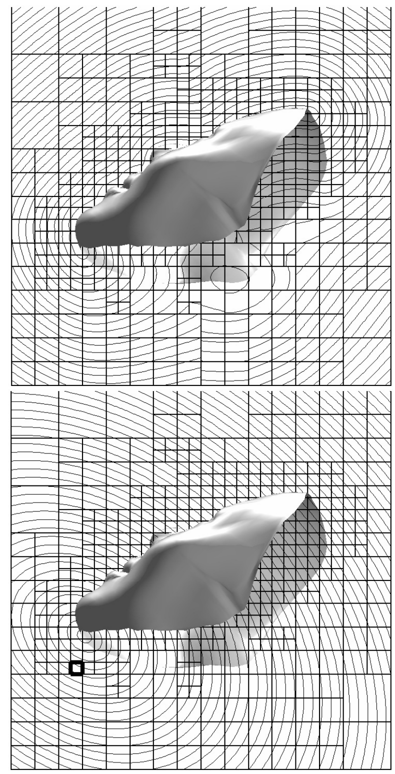


FIGURE 2.5: *A slice through the distance field of a surface M in space.* Bottom: *Level sets of the quadratic function which best approximates $\phi(\mathbf{x}) = \operatorname{dist}(\mathbf{x}, M)^2$ in the highlighted small cell.* Top: *Each cell contains the level sets of that quadratic function which best approximates $\operatorname{dist}(\mathbf{x}, M)^2$ in this cell.*

Lemma 2.8 (Ambrosio and Mantegazza 1998) *If a curve resp. surface M is given as graph of a function whose 2nd order Taylor polynomial reads*

$$y = \frac{\kappa}{2}x^2, \quad \textit{resp.} \quad \frac{1}{2}(\kappa_1 x^2 + \kappa_2 y^2),$$

then the function $\phi(\mathbf{x}) = \operatorname{dist}(\mathbf{x}, M)^2$ in the point $(0, d)$ resp. $(0, 0, d)$ on the surface normal has the Taylor expansion

$$\begin{aligned} \operatorname{dist}(\mathbf{x}, M)^2 &= \frac{d\kappa}{d\kappa - 1}x^2 + y^2 + \ldots, \quad \textit{resp.} \\ \operatorname{dist}(\mathbf{x}, M)^2 &= \frac{d\kappa_1}{d\kappa_1 - 1}x^2 + \frac{d\kappa_2}{d\kappa_2 - 1}y^2 + z^2 + \ldots \end{aligned}$$

Proof: Following [Pottmann and Hofer 2003], we approximate the reference shape by a simpler one where distances can be computed, and subsequently compute the Taylor polynomial of the square of

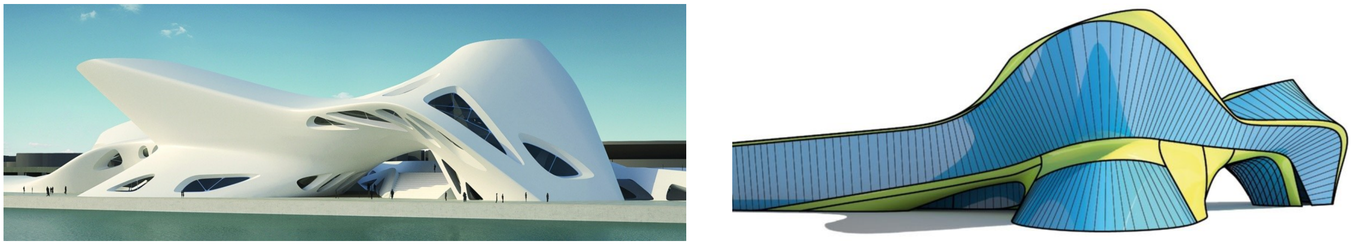


FIGURE 2.6: *Large parts of an architectural design may be representable as a ruled surface.* Left: *design by Zaha Hadid architects for a museum project in Cagliari, Sardinia, which eventually was not realized.* Right: *ruled surfaces approximating this design, holes omitted*

distance. Such a simpler shape is a circle of radius $1/\kappa$ in the curve case, and a torus whose principal radii are $1/\kappa_1, 1/\kappa_2$ in the surface case. The actual computations are a bit tedious. Q.E.D

The important consequence of this result is the following corollary which basically says that from afar, the reference shape looks like a point, but if we are close to it, one does not even see the curvature (we know that anyway, of course).

Corollary 2.9 *Consider a point $\mathbf{x}$ and the closest point $\mathbf{x}^*$ on a smooth reference shape M. Consider also the tangent (resp. tangent plane) T of M in $\mathbf{x}^*$.*

- *If $\mathbf{x}$ is far from M, then locally around $\mathbf{x}$, the distance from M is well aproximated by the following simpler distances:*
  - *the distance from x^* itself, if $\kappa_1\kappa_2 \neq 0$.*
  - *the distance from the ruling contained in the surface, if $\kappa_1\kappa_2 = 0$,*
  - *the distance from the tangent plane, if $\kappa_1 = \kappa_2 = 0$.*
- *If $\mathbf{x}$ is close to M, then locally around $\mathbf{x}$, the distance from M is well approximated by the distance from T.*

For the meaning of "well approximated" see Lemma 2.8.

Proof: If $d \to \infty$ in Lemma 2.8, then $\mathrm{dist}(\mathbf{x}, M)^2$ converges to $\mathrm{dist}(\mathbf{x}, \mathbf{o})^2$, or to the distance from the coorindate axis, or to the distance from the tangent plane – depending on how may coefficients κ_1, κ_2 vanish.

If $d \to 0$, then the limit is y^2 for curves resp. z^2 for surfaces. Q.E.D

Adding approximation constraints to modeling. We consider the following scenario: A surface is to approximate a reference shape M, and the user interactively changes the position of control points. We expect that in the background we perform computations which bring the modified surface back into proximity with M. This task is not specific to ruled surface, but applies to geometric modeling in general. In order to be useful for geometric modeling, it must be performed quickly. The reader might think of interactive modeling of the ruled surfaces which occur in Figure 2.6.

One way to algorithmically treat this interactive modeling situation is to express all desired properties as linear equations. We assume that we have surfaces $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots$ which are defined by control points $\mathbf{a}_j^{(i)}$, $\mathbf{b}_j^{(i)}$. In the following we list desired properties together with a *weight* which tells the algorithm how much we desire them.

- Relations between the variables having to do with the fitting together of invididual surfaces (see paragraph on that topic above) These relations should always be fulfilled. We give a high weight $\omega = 1$.
- Proximity to a reference surface. This is expressed by the requirement that a substantial number of samples $\mathbf{p}_j^i = \mathbf{x}^{(i)}(u_j, v_j)$ is close to M. These conditions should be fulfilled to the extent possible. We give a lower weight, e.g. $\omega = 0.01$. For our algorithmic treatment it is necessary to compute, for each sample $\mathbf{p}_j^i$, its closest point $\mathbf{q}_j^i \in M$ and the equation $\mathbf{n}_j^i \cdot \mathbf{x} = \nu_j^i$ of the tangent plane in that point. We linearize the proximity condition $\mathrm{dist}(\mathbf{p}_j^i, M) \to \min$ in two ways as
$$\mathbf{p}_j^i = \mathbf{x}_j^{*i}, \quad \text{or} \quad \mathbf{n}_j^i \cdot \mathbf{p}_j^i = \nu_j^i.$$
Since the dependence of $\mathbf{p}_j^i$ on the control points is linear, once u_j, v_j is fixed, these are linear equations. According to Cor. 2.9 we give weights $\omega\alpha$ and $\omega(1-\alpha)$ to these two equations, where $\alpha = 1$ if the sample is far away from the surface, and $\alpha = 0$ if it is close.
- The control points have intended locations $\bar{\mathbf{a}}_j^{(i)}$, $\bar{\mathbf{b}}_j^{(i)}$, which are either the previous position or the position given by a user interactively dragging a control point. These conditions should also be satisfied reasonably well. We state them simply as linear equations $\bar{\mathbf{a}}_j^{(i)} = \mathbf{a}_j^{(i)}$ and $\bar{\mathbf{b}}_j^{(i)} = \mathbf{b}_j^{(i)}$ and give them a low weight, e.g. $\omega = 0.1$.

The algorithm now works as follows:

1. Collect all variables in a vector $\mathbf{x} \in \mathbb{R}^N$, where N is 3 times the number of control points needed to describe all spline surfaces involved.
2. Add all linear equations collected above, and multiply each equation with its corresponding weight. This yields a linear system of M equations which we write as $A\mathbf{x} = \mathbf{x}$, where $A \in \mathbb{R}^{M\times N}$, $\mathbf{s} \in \mathbb{R}^M$. Typically $M > N$.
3. Solving $A^\top A\mathbf{x} = A^\top \mathbf{s}$ yields the minimizer $\mathbf{x}$ of $\|A\mathbf{x} - \mathbf{s}\|$.
4. We repeat this process several times, each time recomputing closest points $\mathbf{q}_j^i$ and tangent planes.

This procedure is conceptually simple even if its implementation takes some time (it involves computing closest points, for instance). Note how the nonlinear ingredient in the procedure, namely the distance field of the reference surface, has been dealt with: The variables which depend on the control points in a nonlinear way have simply been fixed during one pass of the algorithm.

It is worth nothing that the speed of convergence is much influenced by the manner of linearization of the distance constraint (using closest points alone yields slower convergence than employing tangent planes).

2C Modeling capabilities of ruled surfaces

Regading degrees of freedom when modeling interactively, ruled surfaces are a bit like curves, and in fact a ruled surface for the purposes of geometric modeling can be considered simply as two curves without any additional constraints. Interesting geometry comes into play when we ask for ruled surfaces wich approximate surfaces especialy well, or for ruled surfaces with special properties, or for composite surfaces which have higher smoothness. Postponing discussion of developables we refer e.g. to [Flöry and Pottmann 2010] and [Flöry et al. 2012].

Smooth composite surfaces. A composite surface formed of ruled strips can be smooth even if the rulings of neighbouring strips meet at an angle – see Figure 2.7. Setting aside for a moment the question how that condition is treated algorithmically, we are interested in the more basic question if it is possible to approximate a given freeform shape by such a sequence of ruled strips. That would be an instance of the *rationalization* problem which occurs in the context of freeform architecture: Can we replace a freeform architectural skin by a sequence of ruled surfaces?

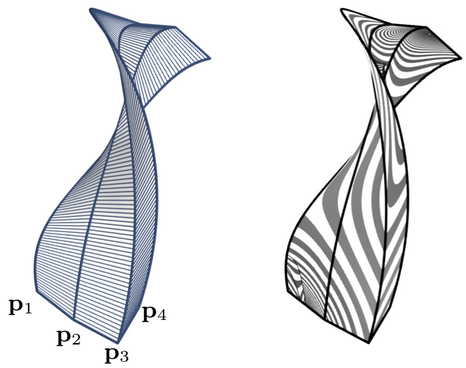


FIGURE 2.7: *A composite surface which enjoys C^1 smoothness as a subset of $\mathbb{R}^3$, without the invidual ruled pieces joining in a smooth manner. Smoothness of the surface is revealed by continuity of reflection lines.*

Asume that $\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3\ldots$ are the vertices of a polyline whose edges are ruling segments of the individual strips (see Figure 2.7). Clearly the two edges

$$\mathbf{p}_{i-1}\mathbf{p}_i \quad \mathbf{p}_i\mathbf{p}_{i+1}$$

must lie in the tangent plane of the composite surface in the point $\mathbf{p}_i$. If a polyline is considered a discrete curve, then its edges are the tangents of that discrete curve, and the plane spanned by 3 successive vertices are its osculating planes. If the polyline in question is to approximate a reference surface M, then it must follow a curve in M with the property that its osculating planes are tangent to M.

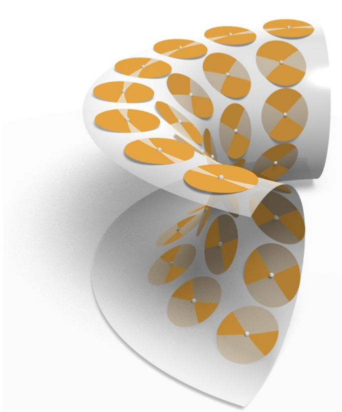


FIGURE 2.8: *Computing asymptotic directions by intersecting a surface with its own tangent planes.*

Differential geometry tells us that such curves exist if the surface is locally saddle-shaped (negatively curved) – they are precisely the asymptotic curves, and they can be computed by intersecting the surface with its own tangent planes (Figure 2.8) and integrating the resulting line field (see Fig. 2.9).

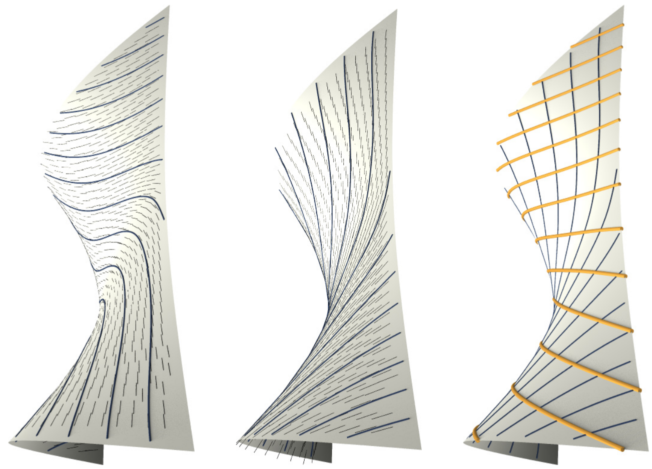


FIGURE 2.9: *Initializing a piecewise-ruled surface which is smooth and which approximates a given surface.* Left and Center: *Computing asymptotic curves by integrating the line field of asymptotic directions.* Right: *A family of curves transverse to the asymptotic curves yields strip boundaries. The asymptotic curves yield the rulings.*

In this way on any negatively curved surface we get information on rulings. The strip boundaries themselves may be chosen arbitrarily, but transverse to the rulings.

Applications in architecture. Figure 2.6 exhibits a smooth union of ruled strips obtained in this way, but it is not very well visible, being on the underside to the right. The reason why we want to approximate that surface by ruled surfaces, is manufacturing: The surface is to be made from concrete, which requires a formworks resp. underconstruction. This underconstruction is much easier to make if straight elements can be used. The actual optimization procedure is performed in a manner similar to the description above:

- We choose an initial collection of ruled surfaces, each of them defined by spline control points $\{\mathbf{a}_i^{(j)}\}$ and $\{\mathbf{b}_i^{(j)}\}$, where j indicates which surface we are in, and i is the running index of control points within each surface.
- The proximity to the reference shape is linearized by conditions which say that a sample $\mathbf{x}_k$ coincides with its closest point projection $\mathbf{x}_k^*$, or alternatively, that the sample is contained in the tangent plane T_k^* (depending on the distance form the reference shape).
- In contrast to our previous discussion we do not require a mathematically precise watertight surface. We rather want to exploit the degrees of freedom of splines and only require proximity of ruled surface boundaries. This done is the same way as proximity to the reference shape, by sampling one boundary curve and treating its partner as the reference shape, and vice versa.
- The smoothness of the composite surface is taken care of by the manner of initialization. In the examples described by [Flöry et al. 2012] this was sufficient.
- Some fairness term is also necessary. It is customary to require that 2nd forward differences or similar expressions are small. Recall that the entire modeling problem was formulated as a single system of linear equations which was solved in the least-squares sense. If we keep within this formalism,

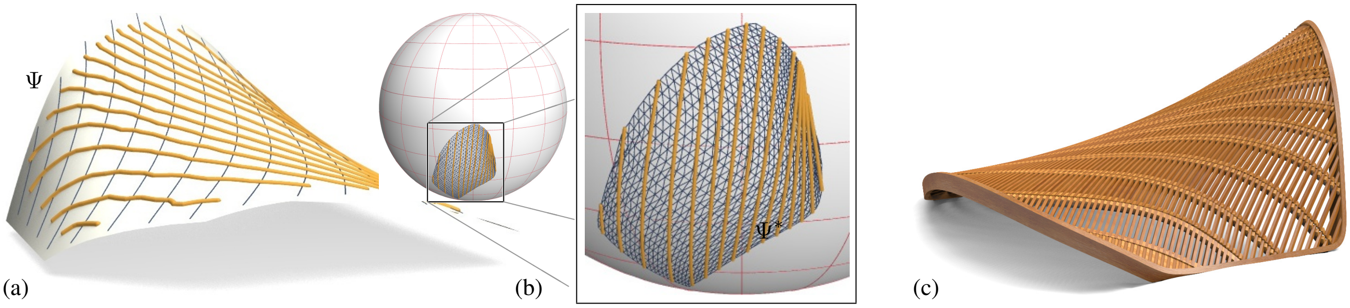


FIGURE 2.10: *Initializing conoidal strips. (a) Suggested strip boundaries in the mesh Ψ (yellow). (b) If the mesh Ψ is mapped to the unit sphere via the asymptotic directions, then strip boundaries are mapped to geodesics in the image mesh Ψ^* (i.e., to great circles). Note that Ψ^* is covered by a family of great circles which are in general position to each other, quite unlike the system of meridians of the sphere. (c) final result. Conoidal surfaces can be made from wooden elements of constant thickness.*

we do not set up a quadratic fairness energy, but simply postulate linear equations

$$\Delta_i^2 = \mathbf{a}_{i-1}^{(j)} - 2\mathbf{a}_i^{(j)} + \mathbf{a}_{i+1}^{(j)} = \mathbf{o}, \quad \text{for all } i, j,$$

and similar for $\mathbf{b}_i^{(j)}$. These equations, when squared and added up, amount to a quadratic fairness functional. They are given a low weight. Other possible fairness functionals involving third order differences or mixed differences:

$$\Delta_i^3 = \mathbf{a}_{i-1}^{(j)} - 3\mathbf{a}_i^{(j)} + 3\mathbf{a}_{i+1}^{(j)} - \mathbf{a}_{i+2}^{(j)} = \mathbf{o},$$
$$\Delta_i \Delta_j = \left(\mathbf{b}_{i+1}^{(j)} - \mathbf{a}_{i+1}^{(j)}\right) - \left(\mathbf{b}_i^{(j)} - \mathbf{a}_i^{(j)}\right) = \mathbf{o}.$$

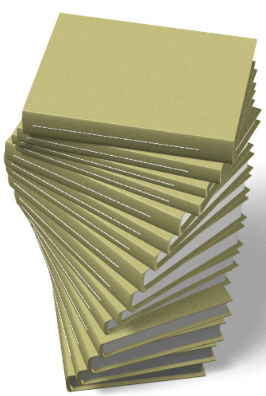


FIGURE 2.11: *A stack of books is bounded by (a discrete version of) four conoidal ruled surfaces: All rulings are parallel to a fixed plane.*

Conoid surfaces. An example of ruled surfaces with special geometry is provided by the conoidal surfaces. It is possible to approximate any saddle-shaped surface by conoidal ruled surfaces: We refer to [Flöry et al. 2012] and Figures 2.11, 2.10.

Exercises to §2

2.1. Compute the dimension of the spline space defined by an admissible knot sequence. That number equals the number of control points needed to describe a spline curve.

2.2. Draw the graphs of spline functions which belong to the knot sequences $(0,0,1,1)$ and $(0,0,0,1,2,3,3,3)$ and $(0,0,0,0,1,2,3,3,3,3)$. *Hint:* The polynomial degree by definition is the multiplicity of boundary knots minus 1.

2.3. Implement de Boor's algorithm to evaluate B-spline surfaces.

2.4. Study the tensor product B-spline surfaces with knots $0,0,1,1$ for the variable u and again $0,0,1,1$ for the variable v. Show that these surfaces are ruled twice, i.e., they carry two families of straight lines.

2.5. Implement the procedures of Prop. 2.5, Example 2.6, and Prop. 2.7.

2.6.† Take a simple surface Φ, e.g. the inner part of a torus, and find a sequence of ruled surfaces which is almost smooth and which approximates Φ.

† Exercises marked this way require more involved tools.

3 Developables in classical surface theory

We already discussed topics of differential geometry in sections 1B and 1C. We saw that the relations between intrinsic flatness (developability) and other surface properties (ruledness) have been cleared up only in the 1950s, i.e., rather late in the history of differential geometry.

The present section is of a different nature: The developable ruled surfaces associated with general smooth surfaces we discuss here have been understood quite well ever since the 18th century. Starting from the 1930's, *discrete* surfaces have emerged, and the relations of discrete developables with discrete surfaces have been studied (cf. the textbook [Sauer 1970] for a summary of this older work).

Recently discrete parametric surfaces have received much interest because of their fundamental relation to discrete integrable systems, and for the deeper understanding of classical surface theory (and transformation theory) which can be obtained by studying a discrete master theory. That aspect of discrete surfaces is the topic of [Bobenko and Suris 2009].

In these lecture notes we do not go beyond elementary properties. We first discuss meshes with planar faces, of which discrete developables are a special case.

3A Conjugate nets

Discrete conjugate nets. Recall that a sequence of planar quadrilaterals is a discrete developable surface, cf. Figure 1.4. It follows that a mesh with quadrilateral faces and regular grid combinatorics can be thought to be made up by a sequence of discrete developables. Such a discrete surface is called a discrete conjugate net.

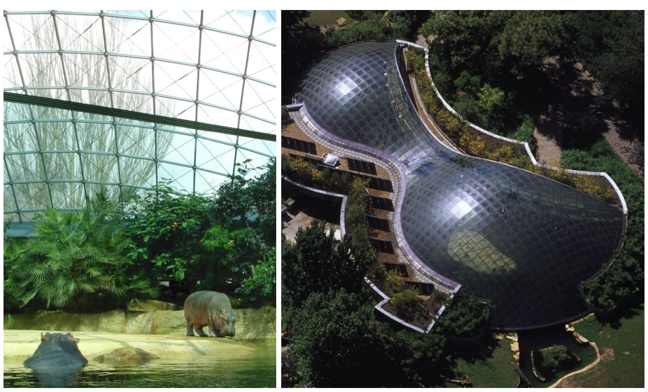


FIGURE 3.1: *Mesh with planar faces and regular grid combinatorics (Hippo house, Berlin Zoo: interior view and aerial view). Each strip of successive quads is a discrete developable. Since this is a surface of very simple geometry, generated by translating one polyline along another, all these developables are cylinders.*

In a limit process where the discrete surface converges to a smooth one via refinement, such that always quadrilateral faces remain planar, we visualize a developable strip converging to a curve, namely the line of contact of the smooth surface with a tangential developable. The lines which carry the edges between succesive quads become the rulings of those developables. Such a curve network is called a smooth conjugate net.

Semidiscrete conjugate nets. There are limit processes which refine only of of the two grid directions. In that case the discrete surface, consisting of finitely many discrete developables, converges to as many smooth developables, see Figure 3.2. We call that a semidiscrete conjugate net.

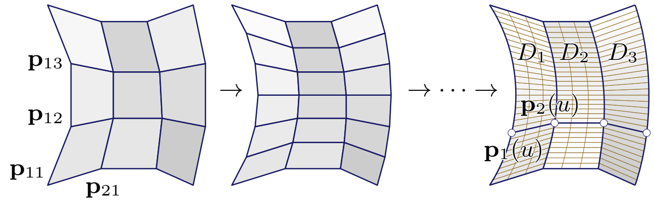


FIGURE 3.2: Semi-discrete surfaces as limits of discrete ones. *Partially subdividing quadrilateral meshes with vertices $\mathbf{p}_{i,j}$ and planar faces $\mathbf{p}_{i,j}\mathbf{p}_{i+1,j}\mathbf{p}_{i+1,j+1}\mathbf{p}_{i,j+1}$ yields, in the limit, a sequence of developables D_i.*

The discrete version, the semidiscrete version, and the smooth version of a conjugate net are only 3 incarnations of the fully discrete 'master' object. It is easy to believe that the discrete objects approximates the smooth one, and in fact a mathematical statement to that effect can be proved [Bobenko and Suris 2009].

Smooth conjugate nets. In differential geometry the smoth conjugate nets are well known. They are networks of curves on surfaces where in each point two curves intersect, and their tangents are conjugate. Conjugacy means that they are orthogonal w.r.t. the second fundamental form. A visual interpretation is given by Figure 3.3:

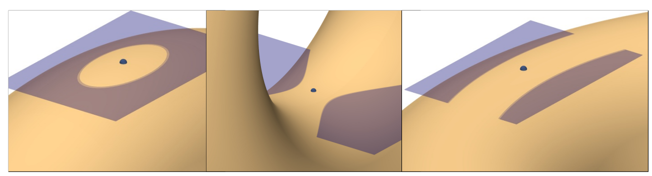


FIGURE 3.3: Conjugate directions. *The local behaviour of a surface Φ is seen from the indicatrix, which is the limit shape of intersection of Φ with a plane very close to the tangent plane. Any parallelogram whose sides are tangent to the indicatrix (in a certain sense of the word) indicate conjugate directions. The parallelograms shown here have this property.*

The *indicatrix* of a surface Φ is a conic which is the limit shape of an almost-tangential intersection. It is a conic whose axes are the principal directions of Φ. If we choose a coordinate system in the tangent plane, then the indicatrix has the form

$$\mathbf{t}^\top \mathbb{II}\, \mathbf{t} = 1, \quad \mathbf{t} \in \mathbb{R}^2$$

with a symmetric 2 by 2 matrix $\mathbb{II}$ (it is no coincidence that this matrix is called $\mathbb{II}$. It is the matrix of the second fundamental form of the surface [do Carmo 1976]). Vectors $\mathbf{t}_1$, $\mathbf{t}_2$ indicate conjugate directions if and only if

$$\mathbf{t}_1^\top \mathbb{II}\, \mathbf{t}_2 = 0$$

If in addition these directions are orthogonal, they are the principal directions. If $\mathbb{I}$ is the matrix of the usual Euclidean scalar product, then the principal directions are computable as eigenvectors as follows:

$$\left.\begin{array}{r} \mathbf{t}_1^\top \mathbb{II}\, \mathbf{t}_2 = 0 \\ \mathbf{t}_1^\top \mathbb{I}\, \mathbf{t}_2 = 0 \end{array}\right\} \iff \left\{\begin{array}{l} \mathbb{I}^{-1}\, \mathbb{II}\, \mathbf{t}_1 = \lambda \mathbf{t}_1 \\ \mathbb{I}^{-1}\, \mathbb{II}\, \mathbf{t}_2 = \mu \mathbf{t}_2 \end{array}\right.$$

We establish the connection to the usual terminology [do Carmo 1976]: In the differential geometry of surfaces, one usually uses

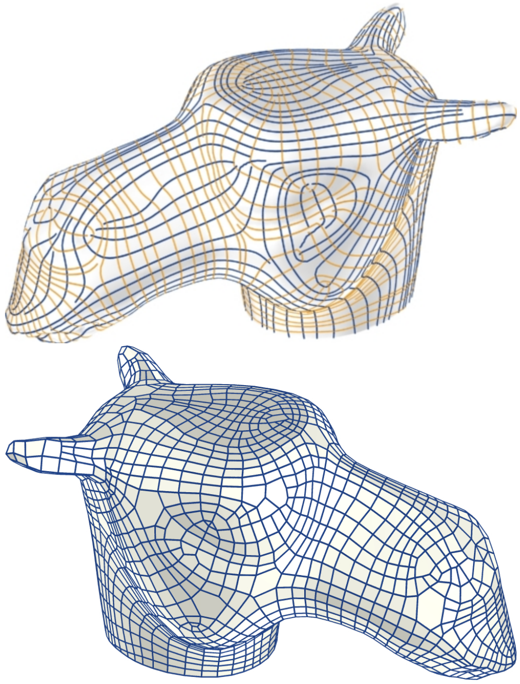


FIGURE 3.4: *The principal network of curves on a surface, and a discrete-conjugate net which is the result of nonlinear optimization initialized from the principal net.*

the partial derivatives $\mathbf{x}_u$, $\mathbf{x}_v$ of the surface $\mathbf{x}(u, v)$ as a basis. Then the first fundamental form is described by the matrix $\mathbb{I} = \left(\begin{smallmatrix} \mathbf{x}_u \cdot \mathbf{x}_u & \mathbf{x}_u \cdot \mathbf{x}_v \\ \mathbf{x}_u \cdot \mathbf{x}_v & \mathbf{x}_v \cdot \mathbf{x}_v \end{smallmatrix}\right)$ of scalar products of basis vectors, and the second fundamental for is described the matrix $\mathbb{III} = \left(\begin{smallmatrix} \mathbf{x}_{uu} \cdot \mathbf{n} & \mathbf{x}_{uv} \cdot \mathbf{n} \\ \mathbf{x}_{uv} \cdot \mathbf{n} & \mathbf{x}_{vv} \cdot \mathbf{n} \end{smallmatrix}\right)$.

In a practical situation it is not difficult to compute the principal directions and the indicatrices. One only has to locally approximate the surface by a quadratic surface, which meanwhile is a standard task (see e.g. [Cazals and Pouget 2003]). After the field of principal directions is computed, one can integrate it and obtain the network of principal curvature lines. Similary one can choose more general conjugate directions and integrate them to obtain a conjugate network (see Figures 3.4 and 3.5).

Solvability of modeling tasks: planar quad meshes. The probem of approximating a surface by a discrete-conjugate net or by a semidiscrete-conjugate net is not easy. It is extensively discussed by [Liu et al. 2006] and [Pottmann et al. 2008]. Basically one has to solve a highly nonlinear optimization problem which has no chance of success unless we start optimization from a point which is already close to the solution. The relation between discrete, semidsicrete, and continuous conjugate nets is extremely helpful in finding such a good starting point.

E.g. we can find a mesh with planar quadrilateral faces approximating the camel head of Figure 3.4 by finding a conjugate network, initializing the quad mesh from the curve network, and apply optimization. Already the initial mesh will have almost-planar faces, and optimization has not much to do. If we start optimization from a mesh which is not yet almost-conjugate, optimization will fail, because "not much" is exactly what optimization manages to do in this case.

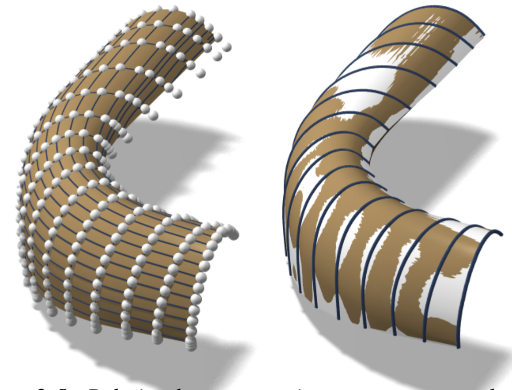


FIGURE 3.5: *Relation between conjugate curve networks and developables.* Left: *Surface Φ with a conjugate curve network and an initial choice of B-spline control points for the purpose of generating develpoable strips.* Right: *Superposition of Φ with the strips resulting from nonlinear optimization.*

FIGURE 3.6: *Discrete developables as a shading system. The roof of the Robert and Arlene Kogod Courtyard in the Smithsonian American Art Museum, by Foster and partners exhibits a mesh with quadrilateral faces and a support structure associated with it. The faces of the mesh are not planar – only the view from outside reveals that the planar glass panels which function as a roof do not fit together.*

Solvability of modeling tasks: develpoable strips. A similar task is to approximate a surface by a sequence of developables. Figure 3.5 gives an example. The developables in question are represented as ruled B-spline spline surfaces, and developability is imposed as a nonlinear constraint. Initialization is done from a conjugate network, so the ruled surfaces to be optimized are already almost-developable, and optimization has not much to do. If we had started form general ruled surfaces, optimization would have failed. For more details see [Pottmann et al. 2008].

3B Developables in support structures

The term 'support structure' in the context of discrete differential geometry has been coined by [Pottmann et al. 2007]. In other publications in the field they are called discrete line congruences [Bobenko and Suris 2009] or, using a more precise terminology, a *torsal* discrete line congruence [Wang et al. 2013]. The following paragraphs discuss a few applications without going into algorithmic details.

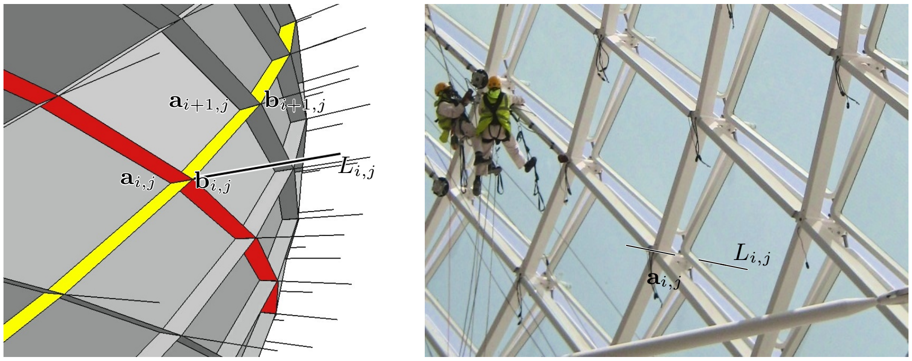


FIGURE 3.7: *A support structure (i.e., a discrete congruence) $\mathcal{L}$ is defined by connecting corresponding vertices of quad meshes A, B where corresponding edges are co-planar. In this way discrete developables (red and yellow) occur along mesh polylines. The support structure is used to align beams such that their intersection in nodes is nice (Yas Marina hotel, Abu Dhabi. Construction by Waagner Biro Stahlbau, Vienna. Mesh by Evolute).*

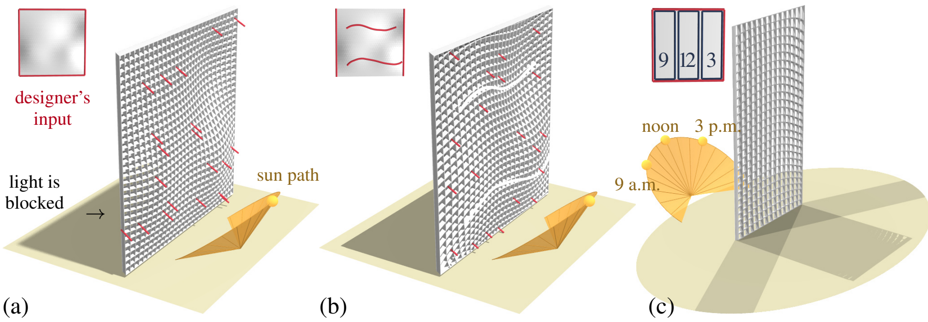


FIGURE 3.8: *Example of the use of discrete developables for shading, taken from* [Wang et al. 2013]. *(a) Light is to be blocked, and developables are aligned with the boundary. (b) Here developables are aligned with a user's design strokes. (c) Here a flat facade is equipped with a shading system whose different parts block light emitted from different sun positions.*

Discrete developables in support structures. If two meshes are combinatorially equivalent and corresponding edges are planar, then the lines which connect corresponding vertices, together with the planes which connect corresponding edges, consitute a support structure. Applications of this concept are e.g. in steel construction (Figure 3.7) or in shading systems (Figures 3.6 and 3.8).

The continuous differential-geometric equivalent of such discrete objects are 2-parameter families of lines with distinguished 1-parameter families of lines (ruled surfaces) in them. Especially we ask the question if the system contains developable surfaces. It turns out that this question is very similar to finding the principal curves in a surface (this branch of differential geometry has gone out of fashion, but see [Pottmann and Wallner 2001]).

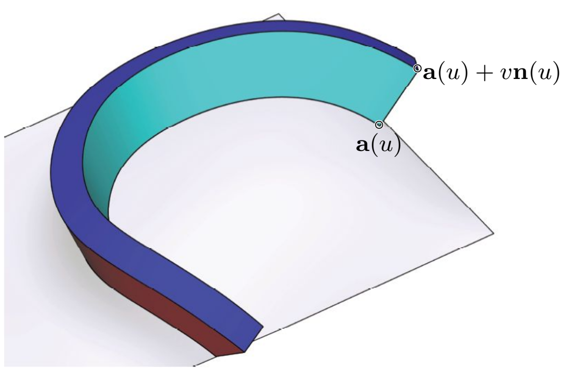


FIGURE 3.9: *Developables orthogonal to a surface mark principal curvature lines. If $\mathbf{a}(u) + v\mathbf{n}(u)$ is the ruled surface defined by a curve $\mathbf{a}(u)$ and the unit normal vectors along that curve, then developability means that $\dot{\mathbf{a}}$ and $\dot{\mathbf{n}}$ are parallel, which is precisely the definition of a principal direction as eigenvector of the shape operator (image:* [Schiftner et al. 2012]*).*

Developables and principal curves. For any surface, there is a default system of lines associated with it, namely the unit normals. We have the following result (cf. Figure 3.9):

Proposition 3.1 *If $\mathbf{a}(u)$ is a smooth curve contained in a smooth C^2 continuous surface, then the normals along that curve form a developable $\iff$ $\mathbf{a}(u)$ is a principal curvature line.*

Proof: The proof is very easy if one is familiar with the differential geometry of surfaces: The condition of principality is that the unit normal vector $\mathbf{n}(\mathbf{a}(u))$ along the curve has the property that $\dot{\mathbf{n}} = \kappa \cdot \dot{\mathbf{a}}$ (i.e., $\dot{\mathbf{a}}$ is an eigenvector of the shape operator with eigenvalue κ). This is exactly the condition of developability of the ruled surface $\mathbf{x}(u, v) = \mathbf{a}(u) + v\mathbf{n}(\mathbf{a}(u))$. Q.E.D

This property will be important in the next section, where we discuss an actual case of freeform architecture realized with the aid of mathematicians.

3C Design dilemmas

The geometric knowledge gathered in this section is sometimes important when one wants to find out if a certain design problem can be solved or not, and how many degrees of freedom are availabe. Take as an example the realization of a freeform shape as a quadrilateral mesh with planar faces — we know that edges must roughly follow the curves of a conjugate network, of which the principal curves are an example. Since orthogonality of edges (at least approximately) is a frequent design intention, we are left with the conclusion that quad meshes with planar faces have almost no degrees of freedom apart from the size of faces. A more thorough study of degrees of freedom by [Zadravec et al. 2010] confirms this impression.

Other design situations involve more degrees of freedom, e.g. assigning a support structure to a general quad mesh (Figure 3.7).

Situations with only few degrees of freedom cause difficulties in freeform architectural design:

FIGURE 3.10: *Doubtless these non-planar quads were subdivided after the design phase.*

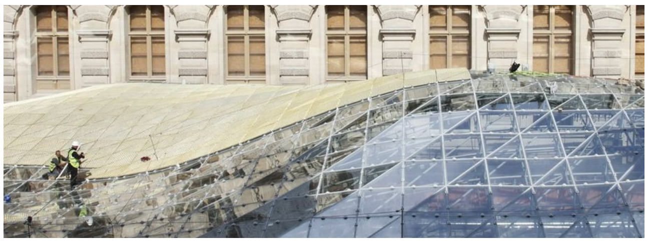


FIGURE 3.11: *Polyhedral surface in the Louvre, Paris, by Mario Bellini Architects and Rudy Ricciotti, during construction. It has only as many triangular faces as are necessary to realize the architect's intentions, and as many quadrilateral faces as possible in order to lighten the load and reduce the number of parts.*

- One may overlook certain geometric constraints in the design phase (see e.g. Figure 3.10).
- Engineering aspects are frequently worked out in detail only after the architectural design has been complete (see e.g. Figure 3.11).
- Any situation where no design freedom is left is unacceptable, since architects or designers are robbed of their function (see following paragraph).

The Eiffel Tower Pavilions. The newly opened pavilions in the first floor of the Eiffel tower are a very good example of developables which occur in a freeform architectural design, and also an example of how to avoid the design dilemmas mentioned above [Schiftner et al. 2012; Baldassini et al. 2013]. For us, the most important aspect of their design is a freeform glass facade which consists of curved sheets of glass, separated by curved beams (see Figures 3.12 and 3.13).

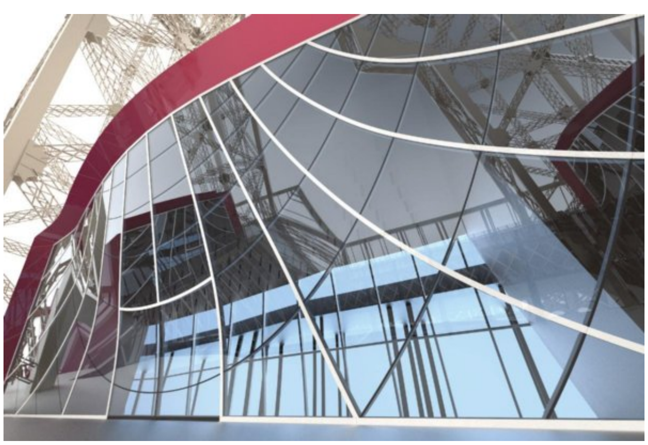


FIGURE 3.12: *Eiffel tower pavilions, rendering.*

.

FIGURE 3.13: *The Eiffel Tower Pavilions feature beams with rectangular cross-section whose side flanks are manufactured by bending. This makes them developable, and makes the entire beam arrangement a semidiscrete version of a principal support structure (image: Evolute).*

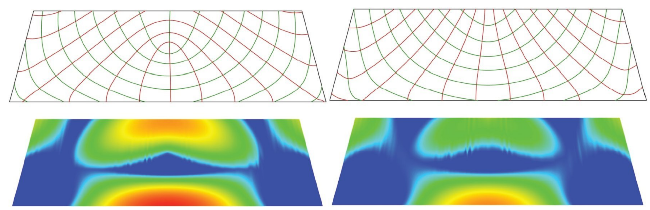


FIGURE 3.14: *A slight modification of a surface can have a great influence on the principal curvature lines (image taken from* [Schiftner et al. 2012]*).*

Since the curved beams have rectangular cross-section and are welded from pieces which are bent from originally flat pieces, the sides of each beam are developables situated orthgonal to the design surface. With Prop. 3.1 we conclude that the beams have to follow the principal curves of the design surface. Once the design surface is fixed, there is no freedom to design the beams except their spacing.

In this case there was early cooperation between architects, engineers (RFR, Paris) and mathematicians (i.e., Evolute, Vienna) and a way was found to restore design freedom: The principal curves change in an unstable manner when the surface is changed. It was possible to change the design surface imperceptibly so that the principal curves (i.e., beams) behave as the architect intended (Fig. 3.14).

We should also mention that the glass between the "vertical" beams has been realized as a union of cylindrical panels whose radii and orientation are been optimized such that the gaps and kink angles between vertically adjacent panels are as small as possible.

4 Modeling with developable surfaces

4A Convex developables

The convex hull $\mathrm{co}(M)$ of a set M is the intersection of all halfspaces which contain M. Conversely the complement of $\mathrm{co}(M)$ is the union of all halfspaces *not* containing M. Especially when determining the convex hull of a space curve, but also in other situations, it happens that the outer surface $\partial\,\mathrm{co}(M)$ of the convex hull is the envelope of a plane which moves while it touches M in two points. This movement envelopes developable surfaces, part of which then occur in $\partial\,\mathrm{co}(M)$. Figure 4.1 shows an example of this.

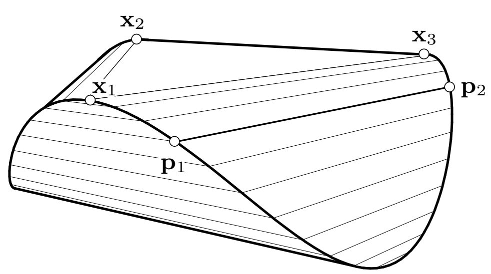


FIGURE 4.1: *The boundary of the convex hull of a space curve M is generated by rolling a plane such that it has 2 contact points $\mathbf{p}_1$, $\mathbf{p}_2$ with M. Planar parts of $\partial\,\mathrm{co}(M)$ occur whenever the plane has 3 or more simultaneous contact points, such as $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$.*

By discretizing the space curve and replacing it by a polygon, we can compute an approximation of the above-mentioned developables with any algorithm capable of computing convex hulls of 3D point clouds. Figure 4.2 shows an example of such a polytopal convex hull (it is computed in a different way, but the result is the same).

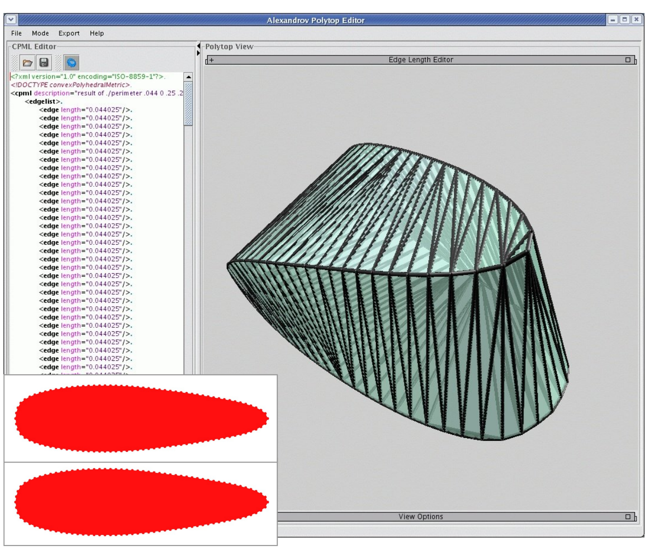


FIGURE 4.2: *Convex hull of a polygon which approximates a space curve. The boundary of the convex hull approximates a developable. This figure is a screenshot from Stefan Sechelmann's "Alexandrov Polyhedron Editor" which computes a convex polytope from its development. In this example, the development is two congruent faces with many edges of constant length which are glued together along their boundaries.*

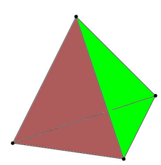
Computing surfaces from an unfolding. Since a piecwise-developable surface has an unfolding into the plane (at least after cutting), it is interesting to study the question under what circumstances the unfolding determines the developable. This is the case if the surface is convex. The corresponding mathematical statement is given twice; once for convex polytopes whose unfolding is given in the shape of poygons, and another time for convex surfaces whose unfolding is given as a collection of flat domains (faces with curved boundaries).

Definition 4.1 *Assume that planar polygonal faces f_i are to be glued together along their edges, so that each edge has a unique partner of the same length, and the result of gluing is a closed surface X. The gluing data are locally convex if for each point $x \in X$ which results from identifying points $x_j \in f_{i_j}$, we can position all these faces in the plane such that points x_j come together but their respective interiors not overlap.*

The non-overlapping condition is fulfilled if for each vertex the sum of angles of incident faces does not exceed 2π.

Theorem 4.2 (A. D. Alexandrov, 1942) *Any abstract surface generated by the locally convex gluing of polygons is isometric to a unique convex polytope.*

For a proof see e.g. [Alexandrov 2005]. An algorithmic and constructive proof was given by [Bobenko and Izmestiev 2008]. Theorem 4.2 does not say that there exists a convex polytope whose edges corresponds to the edges of the faces used for gluing. The polytope may use diagonals of original faces as edges, or original edges may disappear as adjacent faces lie in a common plane. See Figure 4.3 for an unfolding of a tetraehedron which yields to gluing data where face boundaries do *not* correspond to the original polytope's edges.



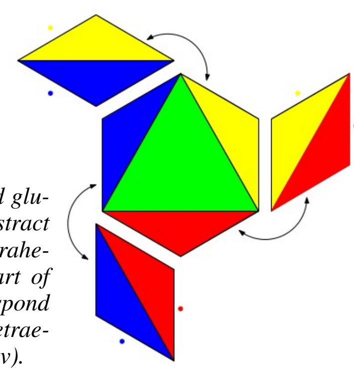


FIGURE 4.3: *Tetrahedron and gluing data which describe an abstract surface isometric to the tetrahedron. The colors are not part of the gluing data. They correspond to the faces of the original tetrahedron (images: Ivan Izmestiev).*

Figure 4.2 shows an example. Given is a convex N-gon with large N and all edges of the same length. A second copy of the same N-gon is glued to the first one such that the i-th edge of the first polygon is glued to the $(i+k)$-th edge of the second polygon, for all i (indices modulo N). The convex polytope which is isometric to this abstract surface generated by gluing is shown by Figure 4.2.

A similar statement holds for surfaces. We extend the definition of locally convex gluing to a collection of faces whose edges might be curves (and whose length is the usual arclength of curves). The non-overlapping condition is satisfied if for each vertex the sum of angles of incident faces does not exceed 2π, and if for each pair $\mathbf{c}_1(u)$, $\mathbf{c}_2(u)$ of edges which are identified, their respective curvatures, measured w.r.t. normal vectors pointing inwards, obey $\kappa_1(u)+\kappa_2(u) > 0$. This is fulfilled automatically for convex faces.

Theorem 4.3 *Theorem 4.2 is true also for a locally convex gluing of faces with curved boundaries.*

FIGURE 4.4: *Formworks for concrete. The wooden structure fills a quarter of a tunnel (the remaining three quarters are filled by similar constructions). It connects a circular opening with a rectangular one. The outer surface of the formworks is a developable defined by boundary curves C_1, C_2 where C_1 is a circle and C_2 is a rectangle. In this special case the developable is part of $\partial\,\mathrm{co}(C_1 \cup C_2)$ (Silvretta reservoir, Austria.* `www.spezialschalungen.com`*).*

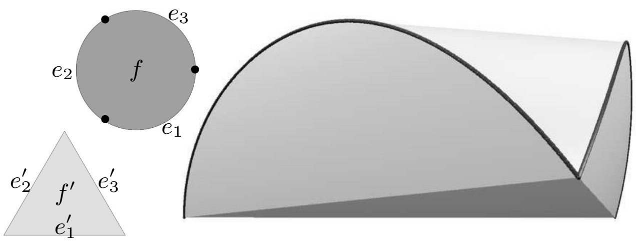


FIGURE 4.5: *Gluing curved faces f, f' together. The circle f must be artificially endowed with 3 vertices to create edges e_1, e_2, e_3 which can be paired with the triangle's edges e_1', e_2', e_3' (image:* [Bobenko and Izmestiev 2008]*)*

A proof can be found in [Pogorelov 1969]. Obviously Alexandrov's theorem 4.2 is a special case of Theorem 4.3. An instance of Theorem 4.3 can be approximated by an instance of Alexandrov's theorem, and can be algorithmically solved using the method of [Bobenko and Izmestiev 2008], but in many cases can also be solved manually, see Figures 4.5 and 4.6.

FIGURE 4.6: *"D-forms" are convex surfaces defined by their development which consists of two planar convex curves. Here they are used as poster walls (design by Tony Wills).*

4B Developables via their dual representation

We mentioned in §1A that a developable is defined by the family of its tangent planes. We used that approach in the previous section (§4A) where we discussed developables which occur on the boundary of convex hulls. We continue our discussion of modeling with developables via their tangent planes, but we drop the assumption of convexity.

Developables from boundaries. In order to find a developable through two curves in space, simply let a plane move such that it touches these two curves in 2 points. In general this yields a 1-parameter family of planes which envelopes a developable. Since the two original curves are tangent to all planes, they will be part of the envelope. This construction has already been seen in Figure 4.1. It has also been used to create the admittedly very simple developable of Figure 4.4 which is interesting because it shows an example of the use of a developable in building construction.

The developable defined by a boundary is not unique: A cylinder and a double cone are defined by the same 2 circles as boundaries. But even if one rules out self-intersecting developables, there might be ambiguity (see Figure 4.7).

Suppose C is the boundary of an as yet unknown developable, and we want to find out what rulings can possible pass through a point $\mathbf{p} \in C$, consider the pencil of planes whose axis is the tangent $T_\mathbf{p}$ of the curve in that point, and find the points $\mathbf{q}_1(\mathbf{p}), \mathbf{q}_2(\mathbf{p}), \ldots$ where such a plane touches the curve. As $\mathbf{p}$ traverses the entire boundary we thus assemble the complete set of correspondences between points of C. Rose et al. [2007] give an algorithmic and interative aproach to categorize these correspondences so that one ends up with a simple and nice developable which has the given curve as its boundary.

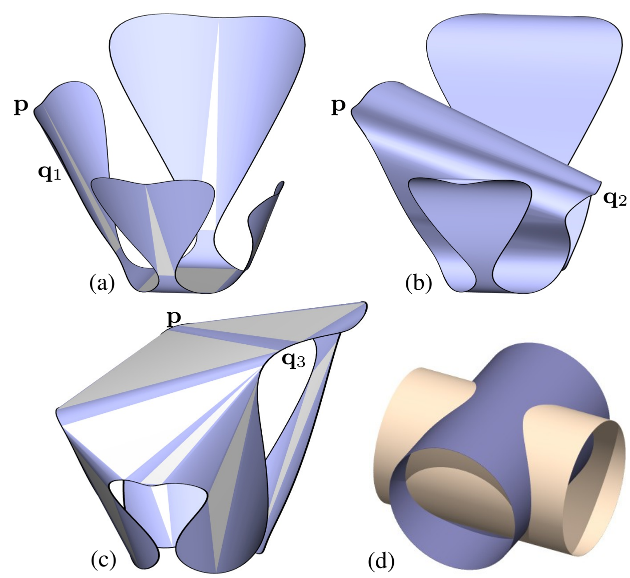


FIGURE 4.7: (a)–(c) *Three different developables with the same boundary (images taken from* [Rose et al. 2007]*).* (d) *the intersection curve C of two cylinders has two developables whose boundary is C, namely either cylinder.*

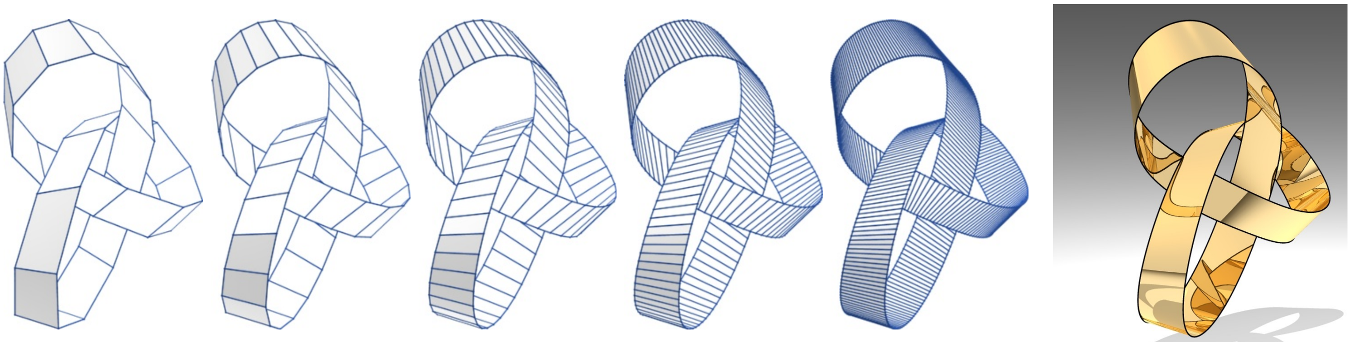


FIGURE 4.8: *A discrete developable undergoes subdivision and optimization (for planarity of quads) in an alternating way. The resulting surface is curvature-continuous, which is proven not mathematically but visually by smoothness of reflection lines.* [Liu et al. 2006].

Dual splines. We consider again the dual representation of a developable surface which was introduced in §1A. A plane of $\mathbb{R}^3$ is described by its equation:

$$n_0 + n_1 x + n_2 y + n_3 z = 0.$$

The coefficients (n_0, n_1, n_2, n_3) are homogeneous and not unique. Planes which are not vertical (i.e., not parallel to the z axis) can also be described as graphs of linear functions

$$z = n_0 + n_1 x + n_2 y.$$

This corresponds to the previous equation if we let $n_3 = -1$. Here the coefficients (u_0, u_1, u_2) are unique.

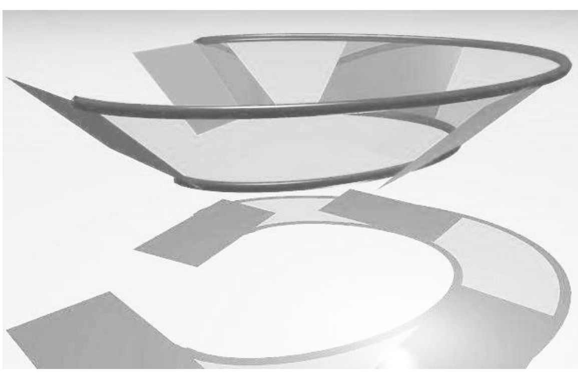


FIGURE 4.9: *Four planes with coordinates $(n_{0,i}, n_{1,i}, n_{2,i})$, $i = 1, \ldots, 4$, serve as control elements of a "dual" Bézier 1-parameter family in plane space, which in turn defines a developable ruled surface.* [Pottmann and Wallner 1999].

A 1-parameter family of planes is described by functions $u_0(u)$, $u_1(u)$, $u_2(u)$ and thus corresponds to a curve in $\mathbb{R}^3$. It may be approximated by a spline curve, where all vectors, including the control poins of the spline, have an interpretation as planes, see Figure 4.9. [Pottmann and Wallner 1999] studied the properties of such splines. It is not so easy to perform geometric modeling if one wants to avoid singularities (see Figure 4.10).

4C Developables as quadrilateral meshes

We have already seen in previous sections that a developable surface can be seen as the limit of a sequence of planar quadrilaterals, see Figures 3.2, 4.8, and 4.12. It therefore makes sense to approach the modeling of developables via the modeling of strips of planar

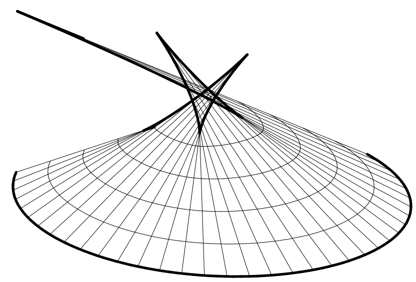


FIGURE 4.10: *Singularities (curve of regression) of a developable surface defined by its tangent planes. The locus of singular points behaves in an unpredictable way.*

quadrilaterals. One could imagine that similar to subdivision algorithms for curves, a coarse strip serves as control elements, and a refinement procedure yields, in the limit, a smooth developable. Assume that two polylines

$$\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N, \quad \mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N$$

are given such that $\mathbf{p}_i, \mathbf{p}_{i+1}, \mathbf{q}_i, \mathbf{q}_{i+1}$ are coplanar for all i. We can apply one of the well-known subdivision rules to the sequences $\{\mathbf{p}_i\}$ and $\{\mathbf{q}_i\}$, e.g. the one named after Lane and Riesenfeld which is known to produce cubic B-spline curves. We let

$$\mathbf{p}_{2j}^{(1)} = \frac{1}{8}\mathbf{p}_{j-1} + \frac{3}{4}\mathbf{p}_j + \frac{1}{8}\mathbf{p}_{j+1}$$
$$\mathbf{p}_{2j+1}^{(1)} = \frac{1}{2}\mathbf{p}_j + \frac{1}{2}\mathbf{p}_{j+1},$$

and similar for $\mathbf{q}_j^{(1)}$.

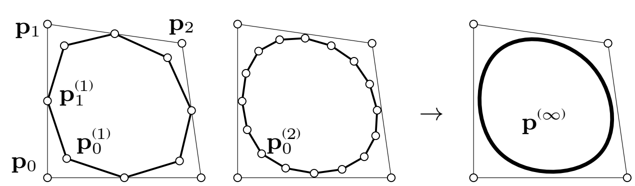


FIGURE 4.11: *Applying the linear Lane-Riesenfeld subdivision algorithm iteratively to a sequence of control points $\mathbf{p}_i$ yields finer and finer polygons $\{\mathbf{p}_0^{(1)}\}$, $\{\mathbf{p}_0^{(2)}\}, \ldots$ which converge to the B-spline curve $\mathbf{p}^{(\infty)}$ defined by those same control points and uniform knots.*

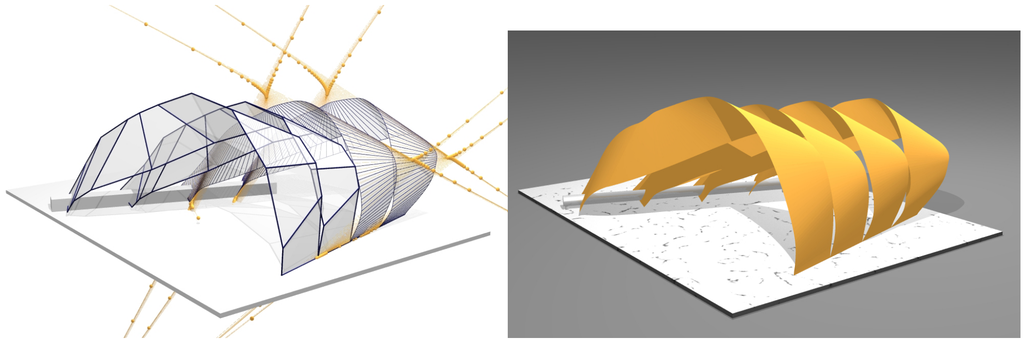


FIGURE 4.12: *A combination of linear subdivision and nonlinear optimization provides a refinement procedure for discrete developables. We show the initial coarse discretization and the result of subdivision. Information on the locus of singularities is also provided* [Liu et al. 2006].

Unfortunately planarity of quadrilaterals derived from the sequences $\mathbf{p}_i$, $\mathbf{q}_i$ does not imply planarity of quadrilaterals derived from the refined sequences $\mathbf{p}_i^{(1)}$, $\mathbf{q}_i^{(1)}$. It is not very difficult, by using black box algorithms for nonlinear optimization, to modify the newly acquired points as little as possible to obtain sequences which again have the coplanarity property, see Figures 4.8 and 4.12. By iterating the procedure we obtain a refinement process for discrete developables. Experimental evidence confirms convergence to smooth developables.

4D Spline techniques

The problem of representing developable surfaces by B-splines (polynomial or rational) has produced many individual contributions on aspects of this approach, in particular on counting the degrees of freedom which are available when developability is imposed as a constraint on a degree $n \times 1$ spline surface (we refer to the respective introductions of the paper [Solomon et al. 2012] and [Tang et al. 2015] for more detailed references).

The combined subdivision + optimization method described in the previous paragraph has aspects of a spline representation as well, since the developable surface which is generated by that method is determined by an initial 'control' shape (even if the dependence on that control shape is nonlinear).

Pottmann et al. [2008] used splines for approximating reference shapes by piecewise-developables. They imposed developability as a nonlinear constraint within their optimization procedures and therefore had to initialize optimization close to the solution.

It is only recently that truly *interactive* methods for modeling with constrained meshes and also developables have been proposed [Tang et al. 2014; Tang et al. 2015]. The principle is the following: One represents the objects at hand by a collection $\mathbf{x}$ of variables, while the contraints are given by some equations $F(\mathbf{x}) = \mathbf{o}$. Some constraints are linear — see our previous discussion in §2B — some are not.

Constraint equations relevant to meshes. If a quadrilateral is defined by the coordinates of its vertices via $\mathbf{x} = (\mathbf{v}_1, \dots, \mathbf{v}_4)$, the planarity is expressed by the single equation

$$F(\mathbf{x}) = \det(\mathbf{v}_2 - \mathbf{v}_1, \mathbf{v}_3 - \mathbf{v}_1, \mathbf{v}_4 - \mathbf{v}_1) = 0.$$

which is cubic. If we insist on quadratic equations we may employ an old trick and introduce additional variables. In the case of planarity of quadrilaterals it is obvious to think of a normal vector. This leads to data $\mathbf{x} = (\mathbf{v}_1, \dots, \mathbf{v}_4, \mathbf{n})$ and the equation

$$F(\mathbf{x}) = \begin{pmatrix} \mathbf{n}^\top(\mathbf{v}_1 - \mathbf{v}_2) \\ \mathbf{n}^\top(\mathbf{v}_2 - \mathbf{v}_3) \\ \mathbf{n}^\top(\mathbf{v}_3 - \mathbf{v}_4) \\ \mathbf{n}^\top(\mathbf{v}_4 - \mathbf{v}_1) \end{pmatrix} = \mathbf{o},$$

which is quadratic. Only 3 of these equations are necessary since they imply the fourth, but for reasons of symmetry it is sometimes best to keep also redundant equations.

Other constraints, like proximity to a reference surface have a more profound nonlinear nature and cannot easily be simplified in an exact manner. They can, however, be linearized in ways we have already discussed (see §2B). Assuming the geometric object of interest is a mesh with M vertices and N faces, this approach leads to $3M + 3N$ variables with $3N$ (or even $4N$) quadratic equations which express planarity. The constraint solver of [Tang et al. 2014] works well if equations are quadratic.

Constraint equations relevant to developables. We return to the spline representation of ruled surfaces presented in §2A (see Figure 2.2). We have a ruled surface expressed in spline form

$$\mathbf{x}(u, v) = (1 - v)\mathbf{a}(u) + v\mathbf{b}(u), \quad \text{where}$$
$$\mathbf{a}(u) = \sum \mathbf{a}_i N_i(u),\ \mathbf{b}(u) = \sum \mathbf{b}_i N_i(u),$$

i.e., the ruled surface is determined by control points $\mathbf{a}_1, \dots, \mathbf{a}_r$ and $\mathbf{b}_1, \dots, \mathbf{b}_r$. Developability is expressed by the coplanarity of the tangents in the two endpoins of the segment $\mathbf{a}(u)\mathbf{b}(u)$:

$$\det(\dot{\mathbf{a}}(u),\ \dot{\mathbf{b}}(u),\ \mathbf{b}(u) - \mathbf{a}(u)) = 0, \quad \text{for all } u.$$

This is an infinite number of cubic conditions. We perform two simplifications: Firstly we make them quadratic by introducing a normal vector field $\mathbf{n}(u)$ and requiring

$$\mathbf{n}(u)^\top \dot{\mathbf{a}}(u) = 0$$
$$\mathbf{n}(u)^\top \dot{\mathbf{b}}(u) = 0$$
$$\mathbf{n}(u)^\top (\mathbf{b}(u) - \mathbf{a}(u)) = 0, \quad \text{for all } u.$$

Another simplification comes from the insight that this equation does not have to hold for all u, but only for a certain number of values u. Since both $\mathbf{a}, \mathbf{b}$ are splines of some polynomial degree d, the derivatives have degree $\le d - 1$, and the determinant condition is a spline of degree not exceeding $3d - 2$ (actually, $3d - 3$). We therefore have to require the determinant condition (or the equivalent condition involving normal vectors) only in $3d - 2$ places of each parameter interval between spline knots. We have thus described a developable spline surface by a finite number of equations. If we choose the spline control points as variables $\mathbf{x}$, then evaluating $\mathbf{a}(u), \dots$ for certain specified parameter values $u = u_j$ makes $\mathbf{a}(u), \dots$ linear functions of x, so the equations expressing developability are quadratic.

Solving constraing equations. [Tang et al. 2014] proposed a method to solve constraint equations which is basically a Newton

FIGURE 4.13: *Approximating the Stanford bunny by composite developable surfaces, using the method of* [Tang et al. 2015] *not for interactive design, but for fast solution of proximity + developability constraints.*

method. The variables are stored in a vector $\mathbf{x} \in \mathbb{R}^N$, the equations are represented as a function $F : \mathbb{R}^n \to \mathbb{R}^M$. Typically redundant equations are present, while on the other hand the equations do not fully determine the solution. Those equations are solved using a Newton method: An initial value $\mathbf{x}_0$ yields the linearization

$$F(\mathbf{x}) \approx F(\mathbf{x} - \mathbf{x}_0) + J \cdot (\mathbf{x} - \mathbf{x}_0) = J\mathbf{x} - \mathbf{p} = \mathbf{o},$$

where J is the matrix of partial derivatives of F and $\mathbf{p}$ is some vector. This linear system is not uniquely solvable because it does not contain the as many linearely independent equations as variables. Because typically the system contains redundant equations it cannot be solved directly anyway because of unavoidable numerical inaccuracies.

Desired properties which are not constraints (like fairness) are express in form of quadratic energies. We have already discussed fairness in §2C: A typical fairness energy involves the squares of forward differences, or other quadratic expressions. One can generally assume that the fairness energy has the form $\|K\mathbf{x} - \mathbf{q}\|^2$ where K is some matrix and $\mathbf{b}$ is some vector. We use this fairness energy to guide the solution of the linear system above, by solving

$$\|J\mathbf{x} - \mathbf{p}\|^2 + \epsilon^2 \|K\mathbf{x} - \mathbf{q}\|^2 \to \min,$$

which is a standard quadratic minimization problem (ϵ is a parameter which determines the influence of the regularizer). Its solution is the solution of the linear system $(J^\top J + \epsilon^2 K^\top K)\mathbf{x} = J^\top \mathbf{p} + \epsilon K^\top \mathbf{q}$. Thus one round of Newton iteration has been successfully performed, and we iterate. [Tang et al. 2014] and [Tang et al. 2015] demonstrate that the procedure is fast enough to allow interactive design of developables and composite surfaces consisting of developables.

4E Developables as triangle meshes

As an example of yet another discretization of developable surfaces we discuss *Lobel-type meshes*, i.e., meshes which consist of equilateral triangles. Since the angle sum around each vertex is 360 degrees in its regular parts, any such discrete surface is locally isometric to a planar domain [Jiang et al. 2015].

The geometry of such meshes is easily encoded in a system of quadratic constraints involving edgelengths, and their modeling can thus be performed using the method of [Tang et al. 2014] which has been discussed in §4D.

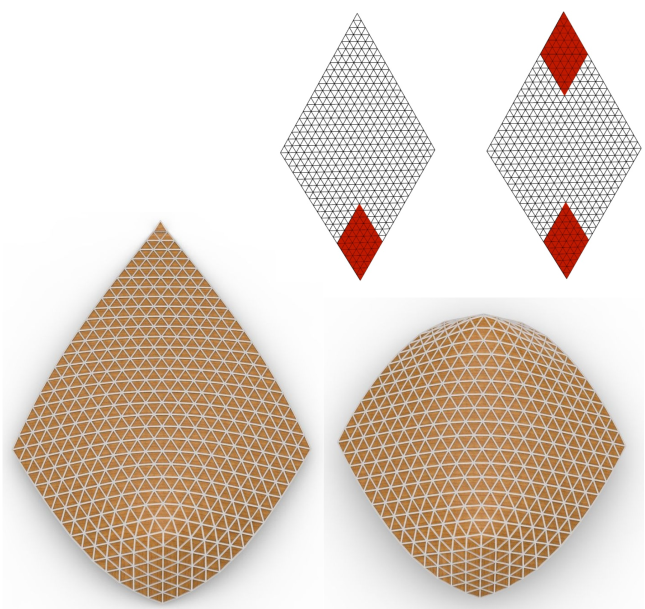


FIGURE 4.14: *Meshes of almost regular combinatorics whose faces are equilateral triangles. Each mesh is isometric to a diamond where one or two smaller diamond-shaped regions have been removed, and edges have been glued together to generate one or two valence 5 vertices (cone point singularities).*

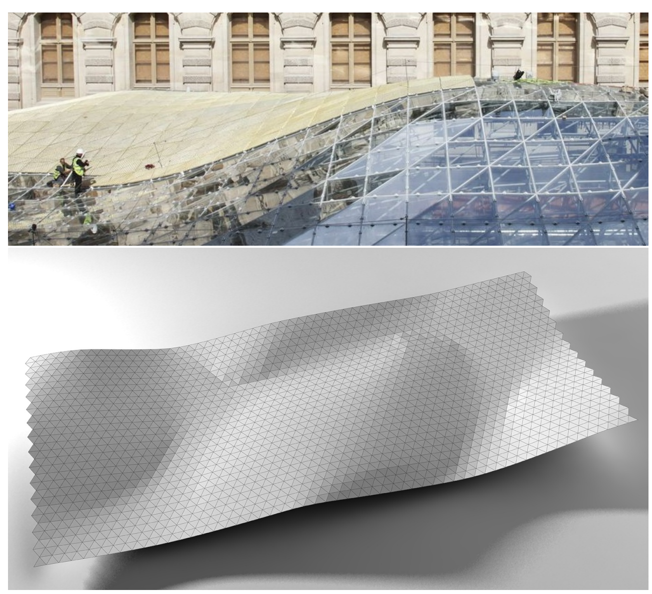


FIGURE 4.15: *The foof of the exhibition of Islamic Arts in the Louvre, Paris, is approximated by a mesh of regular combinatoris whose faces are equilateral triangles. This rendering evokes associations to crumpled paper.*

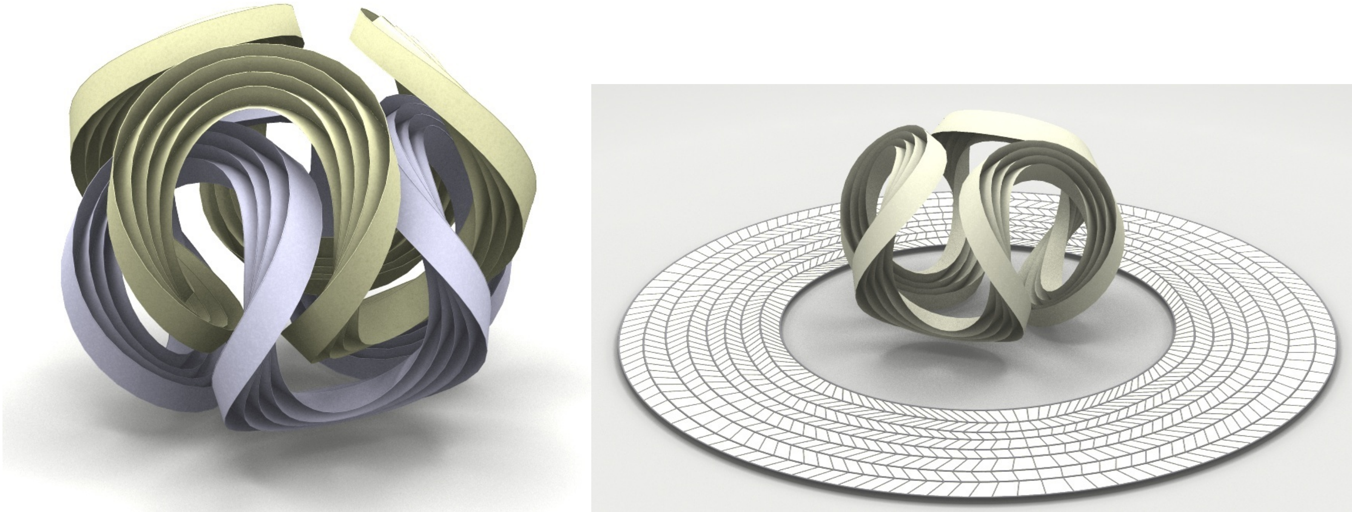


FIGURE 4.16: *Reconstructing the rulings which occur when an annulus is folded along concentric circles in the manner of the sculpture shown by Figure 1.13. This image was obtained by* [Tang et al. 2015] *and constitutes a numerical approximation. The existence of such surfaces is still an open problem.*

4F Modeling curved folds

The methods of §4D have been extended to treat surfaces which are not only piecewise-developable, but piecewise-smooth and entirely developable, meaning that the surface can be flattened into the plane without cutting it open along creases. We have already seen such objects in §1D (Figures 1.10, 1.11, 1.13). The most important geometric property of such surfaces is the following:

Proposition 4.4 *Suppose a piecewise-smooth surface exhibits smooth faces, smooth creases, and vertices where creases meet. Then the surface is everywhere locally isometric to a planar domain, if the following three conditions are met:*

1. *The smooth parts are torsal ruled.*
2. *All creases have the same geodesic curvature w.r.t. the smooth surface on either side. Equivalently, the principal normal of the crease $\mathbf{e}_2$, and the surface normals $\mathbf{n}_1, \mathbf{n}_2$ to either side are related via*
$$\mathbf{e}_2 = \frac{\mathbf{n}_1 - \mathbf{n}_2}{\|\mathbf{n}_1 - \mathbf{n}_2\|}.$$
3. *For each vertex the sum of angles in each face equals 2π.*

Proof: Property 1 implies developability in the interior of ruled parts, cf. §1B. Property 2 implies developability of creases, because geodesic curvature is invariant when unfolding a developable surface onto the plane: We have to make sure that the unfoldings of the crease (w.r.t. the surfaces to either side) coincide, which happens if and only if the curvatures do that. Finally, property 3 implies developability in the vertices. Q.E.D

The properties mentioned by Proposition 4.4 are geometric constraints which can be incorporated as further equations into the methods described in the previous section. We do not go into details but refer to [Tang et al. 2015].

Local and global developability. Proposition 4.4 establishes sufficient conditions for developability, which is a local property. Global developability follows from this local property if the surface is simply connected. The annululs of Figure 4.16 is not. If we want to ensure, during modelling, that a certain curved-crease sculpture keeps being isometric to a planar domain, that development has to be maintained during modeling. We do not go into details but refer to [Tang et al. 2015].

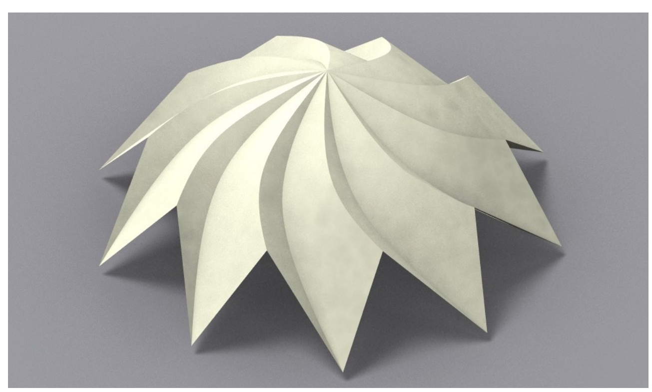


FIGURE 4.17: *We consider shapes foldable from a single sheet of paper which consist of ruled developables which join at smooth creases; the creases meet in vertices. Proposition 4.4 lists three conditions sufficient for developability, in particular an angle sum of 2π for all vertices.*

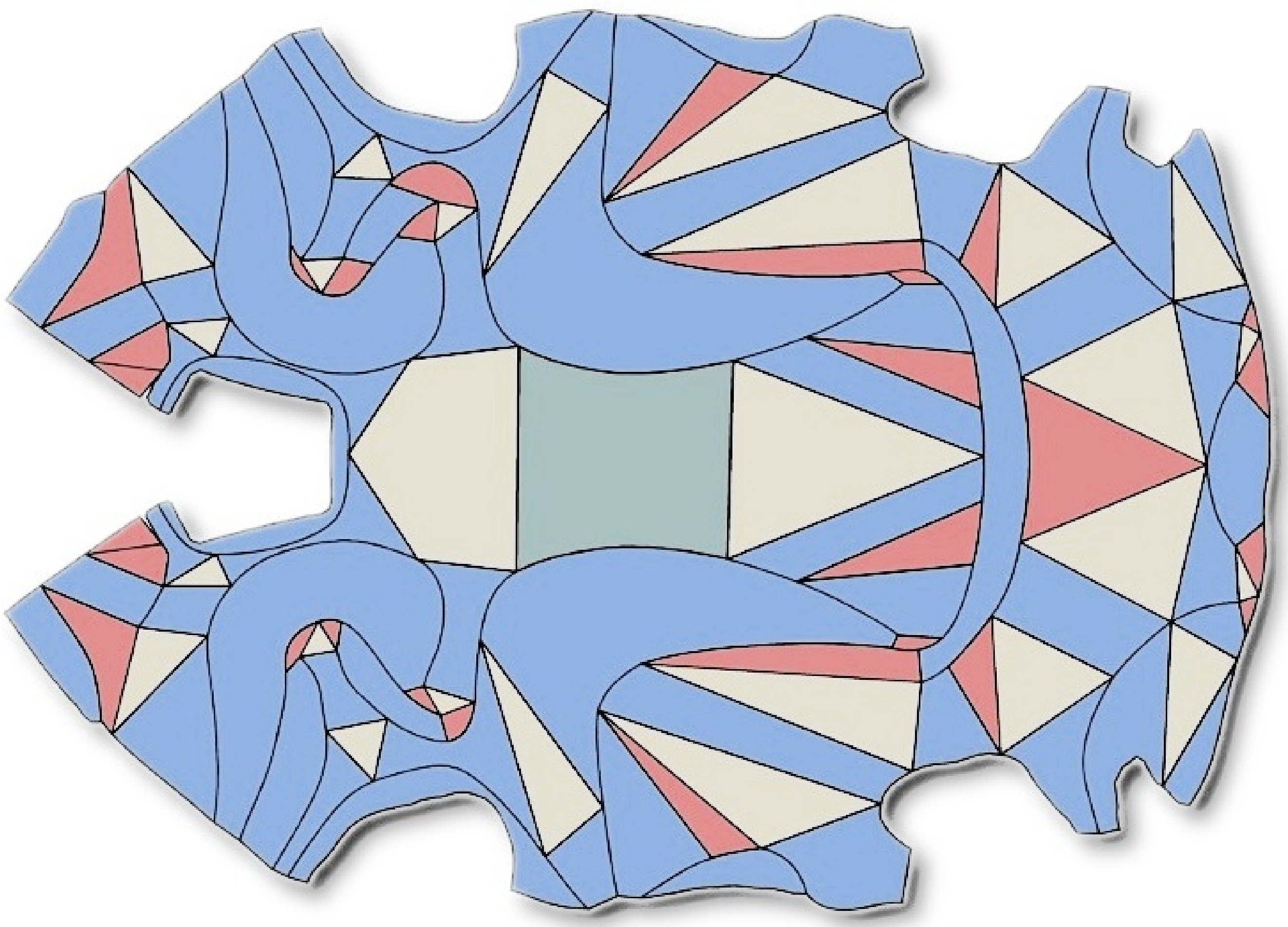


FIGURE 4.18: *The developable car designed by Gregory Epps shown by Figure 1.10 is here unfolded into the plane. The edges shown in this development are not all creases, but in many places only rulings where the surface type changes. The surface types are indicated by colors: red for cones, green for cylinder, light coloring for planes, and blue for general developables (image taken from* [Kilian et al. 2008]*).*

Exercises to §4

4.1. Use paper, scissors and glue to find a convex surface consisting of two developables, each of which develops unto an ellipse with principal axes 5cm and 2cm.

4.2.† Install Stefan Sechelmann's Alexdrov polyhedron editor [Sechelmann 2006] and study its functionality to upload gluing data. Use the software to recreate the previous example. *Hint*: Some predefined gluing data are directly available.

4.3. Given two convex domains D_1, D_2 in perpendicular planes, e.g. in the region $y > 0$ of the xy plane, and in the region $z > 0$ of the xz plane. Discuss how to find the rulings of the "convex" developable S which connects the curves $C_1 = \partial D_1$, $C_2 = \partial D_2$ and which has the property that $\mathrm{co}(C_1 \cup C_2) = D_1 \cup D_2 \cup S$.

4.4. Print Figure 4.18 on a piece of paper and try to fold the car of Figure 1.10.

4.5. Try to fold a sculpture like the one depicted by Figure 1.13 or 4.16 by folding an annulus along concentric circles (*Hint:* This is not easy).

References

ALEXANDROV, A. D. 2005. *Convex polyhedra*. Springer.

AMBROSIO, L., AND MANTEGAZZA, C. 1998. Curvature and distance function from a manifold. *J. Geom. Anal. 8*, 723–748.

BALDASSINI, N., LEDUC, N., AND SCHIFTNER, A. 2013. Construction aware design of curved glass facades: The Eiffel Tower Pavilions. In *Glass Performance Days Finland (Conference Proceedings)*. 406–410.

BOBENKO, A., AND IZMESTIEV, I. 2008. Alexandrov's theorem, weighted Delaunay triangulations, and mixed volumes. *Annales de l'Institut Fourier 58*, 2, 447–505.

BOBENKO, A., AND SURIS, YU. 2009. *Discrete differential geometry: Integrable Structure*. American Math. Soc.

BORRELLI, V., JABRANE, S., LAZARUS, F., AND THIBERT, B. 2012. Flat tori in three-dimensional space and convex integration. *Proc. Nat. Ac. Sc. 109*, 7219–7223.

CAZALS, F., AND POUGET, M. 2003. Estimating differential quantities using polynomial fitting of osculating jets. In *Symposium on Geometry Processing*. 177–178.

DE BOOR, C. 1978. *A Practical Guide to Splines*. Springer.

DEMAINE, E., DEMAINE, M., HART, V., PRICE, G., AND TACHI, T. 2011. (Non)existence of pleated folds: how paper folds between creases. *Graphs and Combinatorics 27*, 377–397.

DO CARMO, M. 1976. *Differential Geometry of Curves and Surfaces*. Prentice-Hall.

FLÖRY, S., AND POTTMANN, H. 2010. Ruled surfaces for rationalization and design in architecture. In *LIFE in:formation. On Responsive Information and Variations in Architecture*, A. Sprecher et al., Eds. 103–109. Proc. ACADIA.

FLÖRY, S., NAGAI, Y., ISVORANU, F., POTTMANN, H., AND WALLNER, J. 2012. Ruled free forms. In [?]. 57–66.

HARTMAN, P., AND NIRENBERG, L. 1959. On spherical image maps whose Jacobians do not change signs. *Amer. J. Math 81*, 901–920.

HARTMAN, P., AND WINTER, A. 1950. On the fundamental equations of differential geometry. *Amer. J. Math. 72*, 757–775.

JIANG, C., TANG, C., TOMIČIĆ, M., WALLNER, J., AND POTTMANN, H. 2015. Interactive modeling of architectural freeform structures – combining geometry with fabrication and statics. In *Advances in Architectural Geometry 2014*, P. Block, N. Mitra, J. Knippers, and W. Wang, Eds. Springer, 95–108.

KILIAN, M., FLÖRY, S., CHEN, Z., MITRA, N., SHEFFER, A., AND POTTMANN, H. 2008. Curved folding. *ACM Trans. Graph. 27*, 3, # 75, 1–9. Proc. SIGGRAPH.

LIU, Y., POTTMANN, H., WALLNER, J., YANG, Y.-L., AND WANG, W. 2006. Geometric modeling with conical meshes and developable surfaces. *ACM Trans. Graph. 25*, 3, 681–689. Proc. SIGGRAPH.

NASH, J. 1954. C^1 isometric imbeddings. *Annals of Math. 60*, 383–396.

POGORELOV, A. V. 1969. *Exterior geometry of convex surfaces*. Nauka, Moscow. [Russian].

POTTMANN, H., AND HOFER, M. 2003. Geometry of the squared distance function to curves and surfaces. In *Visualization and Mathematics III*, Springer, H.-C. Hege and K. Polthier, Eds., 223–244.

POTTMANN, H., AND WALLNER, J. 1999. Approximation algorithms for developable surfaces. *Comput. Aided Geom. Des. 16*, 539–556.

POTTMANN, H., AND WALLNER, J. 2001. *Computational Line Geometry*. Springer.

POTTMANN, H., LIU, Y., WALLNER, J., BOBENKO, A., AND WANG, W. 2007. Geometry of multi-layer freeform structures for architecture. *ACM Trans. Graph. 26*, #65,1–11. Proc. SIGGRAPH.

POTTMANN, H., SCHIFTNER, A., BO, P., SCHMIEDHOFER, H., WANG, W., BALDASSINI, N., AND WALLNER, J. 2008. Freeform surfaces from single curved panels. *ACM Trans. Graph. 27*, 3, #76,1–10. Proc. SIGGRAPH.

ROSE, K., SHEFFER, A., WITHER, J., CANI, M.-P., AND THIBERT, B. 2007. Developable surfaces from arbitrary sketched boundaries. In *Symp. Geom. Processing*. 163–172.

SAUER, R. 1970. *Differenzengeometrie*. Springer.

SCHIFTNER, A., LEDUC, N., BOMPAS, P., BALDASSINI, N., AND EIGENSATZ, M. 2012. Architectural geometry from research to practice — the Eiffel Tower Pavilions. In [?]. 213–228.

SECHELMANN, S., 2006. Alexandrov polyhedron editor.

SOLOMON, J., VOUGA, E., WARDETZKY, M., AND GRINSPUN, E. 2012. Flexible developable surfaces. *Comp. Graph. Forum 31*, 5, 1567–1576. Proc. Symposium Geometry Processing.

TANG, C., SUN, X., GOMES, A., WALLNER, J., AND POTTMANN, H. 2014. Form-finding with polyhedral meshes made simple. *ACM Trans. Graphics 33*, 4, #70,1–9. Proc. SIGGRAPH.

TANG, C., BO, P., WALLNER, J., AND POTTMANN, H., 2015. Interactive design of developable surfaces. submitted.

TSAI, R. 2002. Rapid and accurate computation of the distance function using grids. *J. Comput. Phys. 178*, 175–195.

WANG, J., JIANG, C., BOMPAS, P., WALLNER, J., AND POTTMANN, H. 2013. Discrete line congruences for shading and lighting. *Comput. Graph. Forum 32*, 5, 53–62. Proc. Symposium Geometry Processing.

ZADRAVEC, M., SCHIFTNER, A., AND WALLNER, J. 2010. Designing quad-dominant meshes with planar faces. *Comput. Graph. Forum 29*, 5, 1671–1679. Proc. Symposium Geometry Processing.