This paper has been published in : *Proc. SIGGRAPH'95, Computer Graphics, p377-386*

X-Splines : A Spline Model Designed for the End-User

Carole Blanc Christophe Schlick

LaBRI,[1] *351 cours de la libération, 33405 Talence (France)*
`[blanc|schlick]@labri.u-bordeaux.fr`

Abstract *This paper presents a new model of spline curves and surfaces. The main characteristic of this model is that it has been created from scratch by using a kind of mathematical engineering process. In a first step, a list of specifications was established. This list groups all the properties that a spline model should contain in order to appear intuitive to a non-mathematician end-user. In a second step, a new family of blending functions was derived, trying to fulfill as many items as possible of the previous list. Finally, the degrees of freedom offered by the model have been reduced to provide only shape parameters that have a visual interpretation on the screen. The resulting model includes many classical properties such as affine and perspective invariance, convex hull, variation diminution, local control and* C^2/G^2 *or* C^2/G^0 *continuity. But it also includes original features such as a continuum between B-splines and Catmull-Rom splines, or the ability to define approximation zones and interpolation zones in the same curve or surface.*

1 Introduction

Since the ground work in CAD during the late sixties, many different models of splines have been introduced. One specific characteristic of CAD is that the mathematical models developped by researchers are later manipulated by non-mathematician end users (designers, architects, animators). Therefore, rather than its complete mathematical properties, a major criterion for the evaluation of a spline model may be the ability to understand intuitively the degrees of freedom that it provides. A full study of existing spline models on that particular point lies not within the scope of this short introduction, but let us just take one or two examples.

The popular NURBS model is a good example in which the user has to be familiar with the mathematical structure to obtain best results. For instance, the manipulation of the knot vector is really complex: first the geometrical effects generated by these manipulations can hardly be predicted, second these effects are not robust because further knot manipulations may move them along the curve, and third the effects are propagated along the whole isoparametric curves in the case of surfaces. Even the manipulation of the weights may sometimes be confusing: for instance, the modifications of two adjacent weights are mutually cancelled [11].

The model that accounts the most for the ergonomics of the manipulation is undoubtedly the β-spline model [1] which includes intuitive shape parameters (tension and bias). Yet, if the behaviour of the model is really natural when using global tension and bias, the extended model [2] with local parameters is less convincing, mainly because these parameters are not directly related to the control points. Moreover, the C^0/G^2 continuity of the β-splines is lost by interpolation, this makes them inadequate for many applications [9].

This paper proposes a new spline model that has been designed to make user manipulations as intuitive as possible. Its formulation is presented in four steps: Section 2 presents the list of specifications for the new model, Section 3 explains the principle and the basic formulation, Section 4 derives a more complete expression including an original shape parameter, finally, Section 5 details the general formulation.

[1] Laboratoire Bordelais de Recherche en Informatique (*Université Bordeaux I* and *Centre National de la Recherche Scientifique*). The present work is also granted by the *Conseil Régional d'Aquitaine*.

2 Background

2.1 Definition

In their most general definition, splines can be considered as a mathematical model that associates a continuous representation (curve or surface) with a discrete set of points of an affine space (usually $\mathbb{R}^2$ or $\mathbb{R}^3$). In the case of curves, this definition can be expressed as follows: let $P_k \in \mathbb{R}^3$ with $(k = 0..n)$ be a set of points called *control points*, and let $F_k : [0,1] \rightarrow \mathbb{R}$ (with $k = 0..n$) be a set of functions called *blending functions*, the spline curve generated by the couples (P_k, F_k) is the curve C defined by the parametric equation:

$$\forall t \in [0,1] \qquad C(t) = \sum_{k=0}^{n} F_k(t)\ P_k \tag{1}$$

According to the shape of the blending functions, the resulting curve may either *approximate* the control points or *interpolate* them. Figure 1 and Figure 2 illustrate this distinction by showing two classical examples of spline curves (cubic uniform B-splines [12] in Figure 1, cubic Catmull-Rom [6] in Figure 2). Each figure is divided in two parts, the top shows the control lattice and the curve, the bottom shows the plots of the blending functions. The same graphical framework will be used throughout the paper.

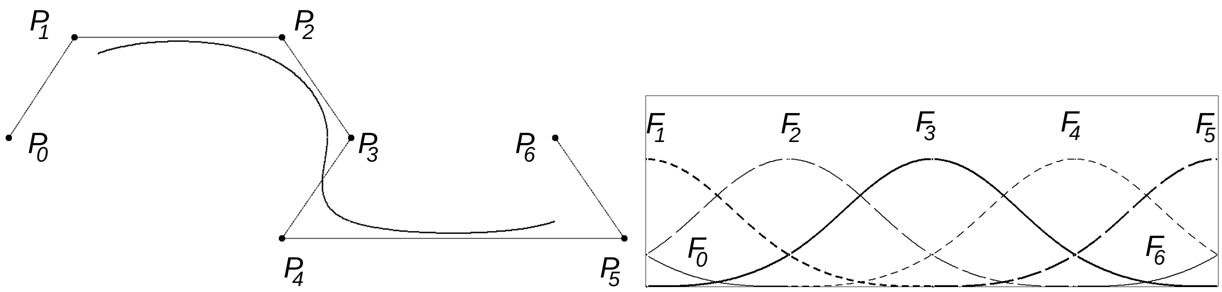

Figure 1: *Uniform B-spline curve*

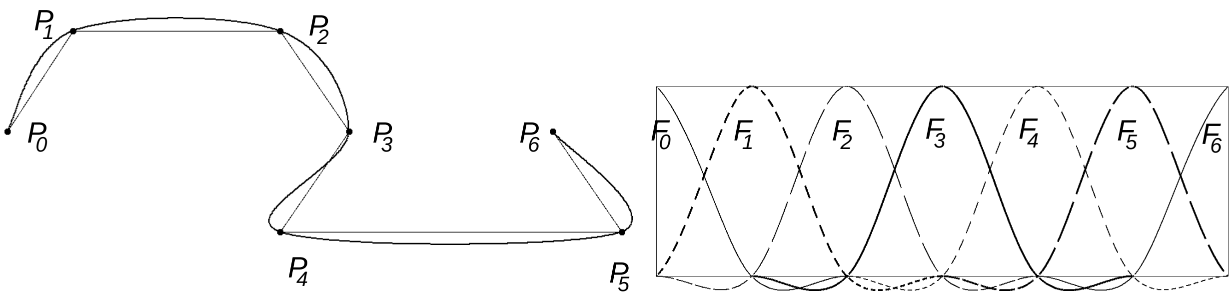

Figure 2: *Catmull-Rom spline curve*

2.2 Properties

The family of curves that obeys Equation 1 is extremely vast and thus many of its members are likely to be of little interest. In fact, the work done over the years in the literature has exhibited many properties that a spline model should include to become useful for geometric modeling. In a recent survey, we have shown that all these properties can be obtained by specific constraints on the blending functions [3].

Using that result, we are going to list now all the properties (as well as the corresponding constraints on the blending functions) that we have found vital or simply desirable to include in our user-oriented spline model:

• **Affine invariance**: The affine transformation of a spline should be obtained by applying the transformation to its control points. This is provided by the *normality constraint*:

$$\forall t \in [0,1] \qquad \sum_{k=0}^{n} F_k(t) = 1 \tag{2}$$

• **Convex hull**: The spline should be entirely contained in the convex hull of its control lattice. This is provided by the normality constraint combined with the *positivity constraint*:

$$\forall k = 0..n \qquad \forall t \in [0,1] \quad F_k(t) \geq 0 \tag{3}$$

• **Variation diminution**: The number of intersections between the spline and a plane (or a line, for 2D splines) should be at most equal to the number of intersections between the plane and the control lattice, which means that the spline should have less oscillations than its control lattice. This property is provided by combining the normality, the positivity with the *regularity constraint*:

$$\begin{array}{ccccc} & \forall k = 0..n & \exists T_k \in [0,1] \ / & & \\ \forall t < T_k & F'_k(t) \geq 0 & \text{and} & \forall k' = k{+}1..n & F_{k'}(t) \leq F_k(t) \\ \forall t > T_k & F'_k(t) \leq 0 & \text{and} & \forall k' = 0..k{-}1 & F_{k'}(t) \leq F_k(t) \end{array} \tag{4}$$

This constraint may appear complex at a first glance, but it simply says that the blending functions are bell-shaped and that two functions cannot cross each other in the zone where they are simultaneously increasing or decreasing.

• **Local control**: Each control point should only influence the shape of the spline in a restricted zone. This property is provided by the *locality constraint*:

$$\begin{array}{c} \forall k = 0..n \quad \exists (T_k^-, T_k^+) \in [0,1]^2 \ / \\ \forall t < T_k^- \ \ F_k(t) = 0 \quad \text{and} \quad \forall t > T_k^+ \ \ F_k(t) = 0 \end{array} \tag{5}$$

A spline may offer more or less local control according to the extent of the influence of a given control point. To quantify this aspect, the notion of L^p *locality* [3] can be used: a spline curve (resp. surface) has got L^p locality when each control point influences p segments (resp. patches) at most.

• **Smooth shapes/Sharp shapes**: The spline model should allow both smooth shapes and sharp shapes and more precisely mixing smooth zones and sharp ones in the same curve. It is well known that parametric continuity does not provided any information on the shape of the curve; therefore one has to use geometric continuity: smooth shapes are G^2 at least, sharp shapes are G^0 at most. On the other hand, parametric continuity is needed to provide smooth motion in animation; therefore the model should also provide C^2 continuity.

• **Intuitive shape parameters**: In addition to the control points, the spline should also provide other degrees of freedom, usually called *shape parameters*. But to be usable by a non-mathematician end user, the role of these parameters should be as intuitive as possible. Among all the shape parameters (knots, weights, tension, bias, curvature) that we can find in existing spline models, only the *local tension effect* (which allows the user to pull the curve locally toward one or several control points) appears totally intuitive.

• **Existence of refinement algorithms**: The spline model should allow the use of refinement or subdivision techniques which are powerful tools that increase the number of degrees of freedom for a spline (control points or shape parameters) without modifying its shape.

• **Representation of conics**: The spline model should be able to represent conic sections, and consequently a large set of curves and surfaces (circles, ellipses, spheres, cylinders, surfaces of revolution, etc) that are intensively used in CAD. The exact representation of conics is one reason for the popularity of the NURBS model [3]. Nevertheless, having only a close approximation (up to the resolution of the display, for instance) is sufficient for most applications.

• **Approximation/Interpolation**: For some applications or some users, approximation splines are preferable, whereas for others, interpolation splines are imperative. For that reason, the model should provide approximation splines and interpolation splines in a unified formulation. Among the existing models, only the general Catmull-Rom model [6] includes such a feature; but we would like to get a step further by allowing the creation of approximation zones and interpolation zones in the same curve.

In the following sections, we describe a new spline model which was designed to fulfill as many items as possible of the previous list. At the current stage in this development, all items but one (the existence of refinement techniques) are fulfilled by the model. The possibility of including the last item will be discussed in the conclusion.

3 Basic X-Splines

3.1 Principle

Building a new spline model from scratch implies defining a new family of blending functions. Among the constraints that have been listed in Section 2, the most difficult to fulfill is the normality constraint. Indeed, finding a family of functions $F_k(t)$ that sum to one whatever the value of t is a tricky task. For that reason, we have chosen to build our blending functions independently of the normality constraint, and then to apply in a final step, a normalization process which replaces $F_k(t)$ by $\overline{F}_k(t)$:

$$\forall t \in [0,1] \qquad \overline{F}_k(t) = \frac{F_k(t)}{\sum_{k=0}^{n} F_k(t)} \tag{6}$$

Thus, the actual blending functions $\overline{F}_k(t)$ will be normalized rational polynomials which, as a side-effect, adds the projective invariance property to the resulting curves.

By combining the different properties recalled in Section 2, we can establish that for a normal, positive, regular and local spline, each blending function $F_k(t)$ is bell-shaped, starts to grow at a given value T_k^-, reaches its unique maximum at a second value T_k and drops to zero at a third value T_k^+ (see Figure 3). In classical spline models, $F_k(t)$ is defined by a piecewise polynomial or a rational piecewise polynomial, composed of as many segments as consecutive intervals between T_k^- and T_k^+ (e.g. four segments with cubic B-splines).

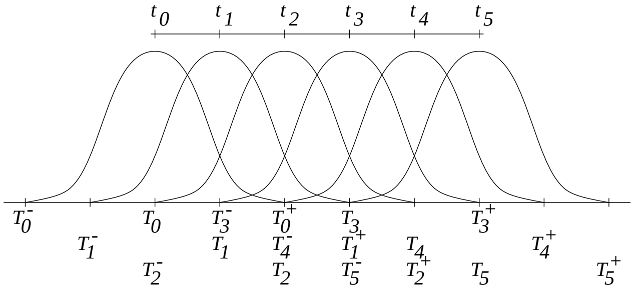

Figure 3: *Configuration of the blending functions*

The driving idea of the new model that we propose here is the following: the non-null part of the blending function should be composed of only two segments[2]. The first, called $F_k^-(t)$, is defined between T_k^- and T_k; the second, called $F_k^+(t)$, is defined between T_k and T_k^+. In order to make this idea clearer, let us take the case of a spline in which each control point P_k influences four segments of the curve (i.e. L^4 locality). This is a usual case (shared by every classical model of cubic splines, for instance) and is often considered [2] as the best trade-off between low degree splines on one hand (which are closely related to the control lattice and thus can hardly provide very smooth shapes) and high degree splines on the other hand (which can hardly provide very sharp shapes).

By definition, for an L^4 spline, each blending function is non-null over four consecutive intervals of the knot vector: $F_k(t)$ becomes non-null at knot t_{k-2}, is maximal at knot t_k and becomes null again at knot t_{k+2} (the knots are shown on the top of Figure 3). As $F_k(t)$ is composed only of two segments, it depends only on t_{k-2}, t_k and t_{k+2}. Thus there is a kind of alternation in the way the knots are taken into account (even points use even knots and odd points use odd knots). Moreover, as we will see, the blending functions F_{k-2} and F_{k+2} cross each other at knot t_k and all the derivation of the model is based on this crossing. For that reason, we have called this new model, **cross-splines** or **X-splines**, for short.

[2] In fact, we have also tried the case where the non-null part is composed of only one segment. But this makes the model much more expensive (degree 8 rational polynomials) with no additional features.

3.2 Formulation

In fact, once the general principle has been established, the basic formulation of the new model can be derived quite naturally. Let us first take the case of a uniform knot vector:

$$\forall k = 1..n \qquad t_k - t_{k-1} = \Delta$$

If we apply the following reparametrization to the curve,

$$u(t) = \frac{t - t_{k-2}}{t_k - t_{k-2}} = \frac{t - t_{k-2}}{2\Delta} \tag{7}$$

we are assured that $u = 0$ at knot t_{k-2} where $F_k(t)$ starts to grow and $u = 1$ at knot t_k where $F_k(t)$ reaches its maximum. Therefore, we have to find a polynomial $f(u)$ defined on the range $[0, 1]$ which can be linked to the left part of $F_k(t)$ by:

$$F_k^-(t) = f\left(\frac{t - t_{k-2}}{2\Delta}\right) \tag{8}$$

Because we want a C^2 continuous curve, the following constraints for the function $f(u)$ can be immediately derived:

$$f(0) = 0 \qquad f'(0) = 0 \qquad f''(0) = 0 \tag{9}$$

As the maximum of the blending function is reached at $u = 1$, its first derivative is necessarily null. Moreover, we can set $f(1) = 1$ because the normalization step will reduce the maximum to its exact value anyway. Finally, the second derivative at $u = 1$ can be set to a given constant (we call this constant $-2p$ to simplify the formulation):

$$f(1) = 1 \qquad f'(1) = 0 \qquad f''(1) = -2p \tag{10}$$

Thus we have derived a system of six constraints. As we search for a polynomial solution, it will necessarily be quintic, in order to get six degrees of freedom. By matching the constraints and the coefficients of the polynomial, we obtain:

$$f_p(u) = u^3 \left(10 - p + (2p - 15)\ u + (6 - p)\ u^2\right) \tag{11}$$

Moreover, the property of regularity requires an increasing function on the range $[0, 1]$ and thus a positive derivative. Therefore there is an additional condition on p:

$$0 \leq p \leq 10$$

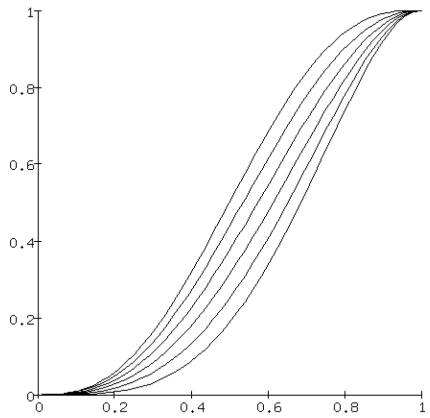

Figure 4: *Function $f_p(u)$ for $p = 0, 2, 4, 6, 8, 10$*

The function $f_p(u)$ (see Figure 4) provides the left part of $F_k(t)$ according to Equation 8. By reversing the direction and the origin of the reparametrization, the right part of $F_k(t)$ is obtained similarly:

$$F_k^+(t) = f_p\left(\frac{t_{k+2} - t}{2\Delta}\right) \tag{12}$$

The two functions F_k^- and F_k^+ join at knot t_k with C^2 continuity ($F_k'(t_k) = 0$ and $F_k''(t_k) = -p/2\Delta^2$) which means that the global blending function $F_k(t)$, and therefore the whole curve $C(t)$, are C^2 continuous.

Finally, we get the formulation for a segment of the curve $C(t)$ on the parameter range $[t_{k+1}, t_{k+2}]$, defined by the four control points $P_k, P_{k+1}, P_{k+2}, P_{k+3}$:

$$C(t) = \frac{A_0(t)\ P_k + A_1(t)\ P_{k+1} + A_2(t)\ P_{k+2} + A_3(t)\ P_{k+3}}{A_0(t) + A_1(t) + A_2(t) + A_3(t)} \tag{13}$$

$$A_0(t) = f_p\left(\frac{t_{k+2}-t}{2\Delta}\right) \qquad A_1(t) = f_p\left(\frac{t_{k+3}-t}{2\Delta}\right)$$

$$A_2(t) = f_p\left(\frac{t-t_k}{2\Delta}\right) \qquad A_3(t) = f_p\left(\frac{t-t_{k+1}}{2\Delta}\right)$$

The process defined above has provided a *quintic rational approximation spline model* that includes the properties of normality, positivity, regularity, locality and C^2 continuity. Moreover, the curves contain a degree of freedom $p \in [0, 10]$ which allows a (slight) modification of their shapes. It should be noticed that a very interesting case is obtained for $p = 8$. Indeed, after the normalization step, the blending functions are very close to the cubic uniform B-splines basis functions (see Figure 6). It means that the resulting curves — call them *basic X-splines* — are almost identical to the uniform cubic B-splines (compare Figure 5 and Figure 1).

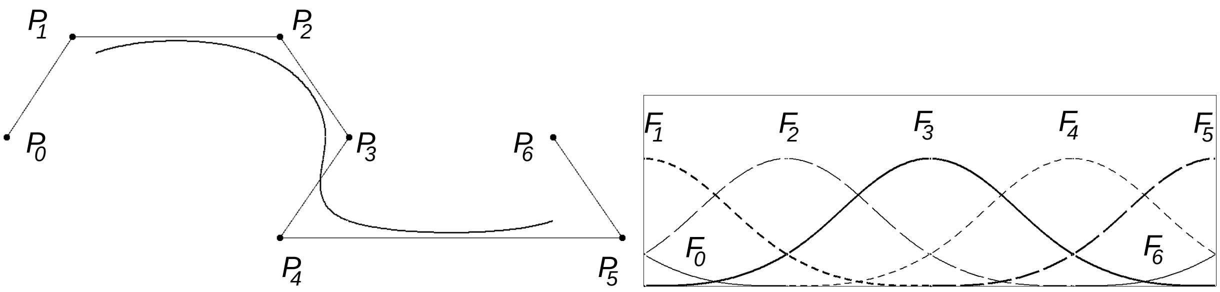

Figure 5: *Basic X-spline curve*

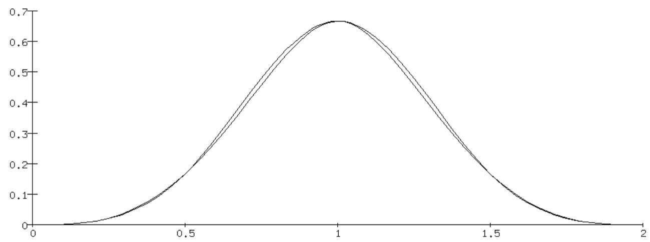

Figure 6: *Similarity of the cubic B-splines and the basic X-splines blending functions (for $p = 8$)*

4 Extended X-Splines

4.1 Formulation

The degree of freedom p in Equation 11 does not offer enough variety in the shapes of the blending functions (see Figure 4) to provide interesting effects on the resulting curves. Therefore, it appears somewhat useless in the formulation of the new model. In fact, the existence of this degree of freedom will be hidden to the end user. As we will see below, this parameter p is needed to manage another parameter s, that we are going to introduce now and which is the actual degree of freedom accessible by the end user.

Among the items of our list of specifications, *tension* and *angular shapes* (G^0 continuity) can be included in our model by the same derivation. Indeed, the basic idea which has led to the concept of tension in the spline literature is to be able to strain the curve (or the surface) in order to pull it toward the control lattice. At its limit, this process forces the curve to interpolate one or several control points, and due to the convex hull property, this interpolation will create sharp edges.

To bring the curve closer to a given part of the control lattice, one has to increase the influence of the corresponding control points. A straightforward idea to realize this process is to add a specific weighting coefficient to each control points. But, as we have recalled in Section 1, this solution (which is used in every classical rational spline) does not work in a satisfying way, because the influences of neighbouring weights are mutually cancelled. Therefore, we propose here an original solution to include the concept of tension, which does not contain the drawback of the existing models.

To illustrate this new solution, let us take the blending functions F_2, F_3 and F_4 in Figure 3. We know that F_3 reaches its maximum at t_3. But, as F_2 and F_4 are not null at t_3, the normalization process has set the actual maximum to $F_3/(F_2+F_3+F_4)$. Therefore, a way to increase this maximum, in order to bring the curve closer to the control point P_3, is to decrease $F_2(t_3)$ and $F_4(t_3)$.

We know that in the area of interest, F_2 (respectively F_4) decreases (respectively increases) monotonically in the range $[t_2, t_4]$. Thus, to obtain smaller values for these functions at t_3, one has to speed up the decrease of the former and to slow down the increase of the latter. To realize these two operations symmetrically, we actually push the *crossing point of* F_2 *and* F_4 down toward the horizontal axis. For that, we introduce a new degree of freedom $s_3 \in [0,1]$ at point P_3. This parameter will be used, first to compute the value T_2^+ (where F_2 becomes null) by interpolation between t_4 and t_3:

$$T_2^+ = t_3 + s_3\ (t_4 - t_3) = t_3 + s_3\ \Delta$$

and second, to compute the value T_4^- (where F_4 becomes non null) by interpolation between t_3 and t_2:

$$T_4^- = t_3 + s_3\ (t_2 - t_3) = t_3 - s_3\ \Delta$$

In other words, it means that F_2 (respectively F_4) is null all over the range $[T_2^+, t_4]$ (respectively $[t_2, T_4^-]$). The same operation can be done for each k. The resulting values (T_k^-, T_k^+) have to be replaced in the reparametrization equations (Equation 8 and Equation 12) as follows:

$$F_k^-(t) = f_p\left(\frac{t - T_k^-}{t_k - T_k^-}\right) \qquad F_k^+(t) = f_p\left(\frac{t - T_k^+}{t_k - T_k^+}\right) \tag{14}$$

The two parts of $F_k(t)$ still join at t_k, their first derivatives are still null but their second derivatives are different:

$$F_k''(t_k^-) = \frac{-2p}{(t_k - T_k^-)^2} \qquad F_k''(t_k^+) = \frac{-2p}{(t_k - T_k^+)^2} \tag{15}$$

Here is the point where our parameter p will finally be used. Indeed, in order to equal the left and right expressions, the only thing to do is to use a specific value for p (noted p_{k-1}) in F_k^- and another one (noted p_{k+1}) in F_k^+. Taking

$$p_{k-1} = \frac{2\ (t_k - T_k^-)^2}{\Delta^2} \quad \text{and} \quad p_{k+1} = \frac{2\ (t_k - T_k^+)^2}{\Delta^2} \tag{16}$$

gives

$$F_k''(t_k^-) = F_k''(t_k^+) = -\frac{4}{\Delta^2}$$

which provides C^2 continuity but assures also that the parameters p_k are in the range $[0, 8]$ as needed to get the property of regularity and to obtain the cubic B-splines as a limit case.

Therefore we can derive a new formulation[3] for the segment of the curve $C(t)$ on the range $[t_{k+1}, t_{k+2}]$ defined by the four control points $P_k, P_{k+1}, P_{k+2}, P_{k+3}$:

$$C(t) = \frac{A_0(t)\ P_k + A_1(t)\ P_{k+1} + A_2(t)\ P_{k+2} + A_3(t)\ P_{k+3}}{A_0(t) + A_1(t) + A_2(t) + A_3(t)} \tag{17}$$

$$A_0(t) = t > T_k^+ \quad ? \quad 0 \quad : \quad f_{p_{k-1}}\left(\frac{t - T_k^+}{t_k - T_k^+}\right)$$

$$A_1(t) = t > T_{k+1}^+ \quad ? \quad 0 \quad : \quad f_{p_k}\left(\frac{t - T_{k+1}^+}{t_{k+1} - T_{k+1}^+}\right)$$

$$A_2(t) = t < T_{k+2}^- \quad ? \quad 0 \quad : \quad f_{p_{k+1}}\left(\frac{t - T_{k+2}^-}{t_{k+2} - T_{k+2}^-}\right)$$

$$A_3(t) = t < T_{k+3}^- \quad ? \quad 0 \quad : \quad f_{p_{k+2}}\left(\frac{t - T_{k+3}^-}{t_{k+3} - T_{k+3}^-}\right)$$

$$p_{k-1} = \frac{2}{\Delta^2}\ (t_k - T_k^+)^2 \qquad p_k = \frac{2}{\Delta^2}\ (t_{k+1} - T_{k+1}^+)^2$$

$$p_{k+1} = \frac{2}{\Delta^2}\ (t_{k+2} - T_{k+2}^-)^2 \quad p_{k+2} = \frac{2}{\Delta^2}\ (t_{k+3} - T_{k+3}^-)^2$$

The expression of $C(t)$ seems complex but in fact it can be implemented very compactly and efficiently (12 lines of source code in C language).

So for the end user, an *extended X-spline* is totally defined by a set of quadruples (x_k, y_k, z_k, s_k) with $k = 0...n$. All these degrees of freedom have a very simple interpretation. The parameters $(x_k, y_k, z_k) \in \mathbb{R}^3$ are the coordinates of the control points P_k. The parameter $s_k \in [0, 1]$ symbolizes the *distance between the curve and the control lattice*: when $s_k = 1$, the curve passes relatively far away from point P_k; when s_k decreases, the curve comes closer and closer to P_k; finally when $s_k = 0$, the curve passes through P_k.

It should be noticed that the curve is always C^2 (due to the construction process that has been used), even when it interpolates a control point P_k. But in that case, the first and second derivatives drop to zero at t_k and therefore the curve is usually (when P_{k-1}, P_k and P_{k+1} are not aligned) only G^0 at P_k. In other words, it means that, even if it is always C^2, the model enables the creation of angular points or sharp edges.

4.2 Examples

This section demonstrates the role of the parameter s_k by showing its influence on the resulting shapes. The basic formulation defined in Section 3 is a particular case of the extended one, where all parameters s_k are set to one. As we have seen, basic X-splines are almost identical to uniform cubic B-splines.

A first variant consists in setting s_0 and s_n to zero in order to interpolate the end points of the control lattice and thus to enable better control of the curve boundaries. The resulting curves — call them extremal X-splines (see Figure 7)— are very close to the classical extremal cubic B-splines (also called non-periodic cubic B-splines).

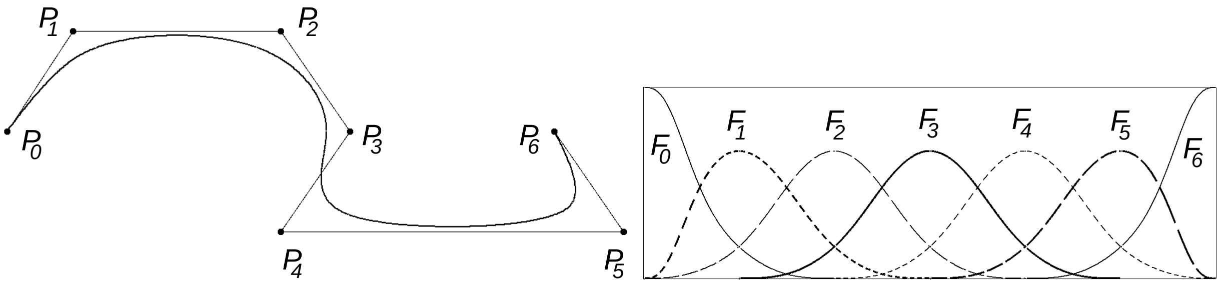

Figure 7: *Extremal X-spline curve*
$s_0 = 0, s_1 = 1, s_2 = 1, s_3 = 1, s_4 = 1, s_5 = 1, s_6 = 0$

Let us now decrease the value of one parameter s_k (say s_3). By comparing Figure 7 and Figure 8, one can see that the crossing point of F_2 and F_4 at knot t_3 has been pushed down.

[3] We use here the (*test ? a : b*) operator borrowed from the *C* programming language which allows one to write multiple expressions in a compact way.

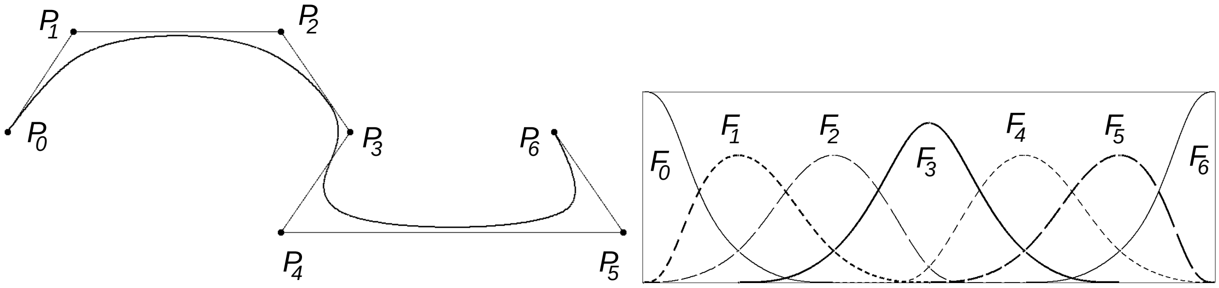

Figure 8: *Augmentation of the influence of point P_3*
$s_0 = 0, s_1 = 1, s_2 = 1, s_3 = 0.5, s_4 = 1, s_5 = 1, s_6 = 0$

Therefore, after the normalization step, the maximum of F_3 has been increased and the curve has been pulled toward P_3. Moreover, neither the maximum of F_2 nor the maximum of F_4 has been modified. This means that the curve has not changed near P_2 or P_4: all the modifications are localized in the neighbourhood of point P_3. More precisely, one can show that a shape parameter s_k influences only two segments of the curves which is half the extent of the other three coordinates (x_k, y_k, z_k) of point P_k (i.e. L^2 locality rather than L^4).

While s_3 decreases, the maximum of F_3 increases. Finally, for $s_3 = 0$, this maximum is set to one, which provides a "sharp" (G^0 continuous) interpolation of point P_3.

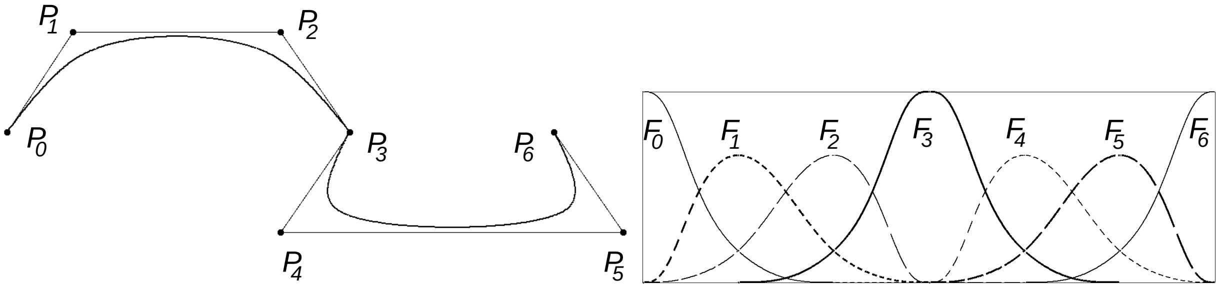

Figure 9: *Sharp interpolation of point P_3*
$s_0 = 0, s_1 = 1, s_2 = 1, s_3 = 0, s_4 = 1, s_5 = 1, s_6 = 0$

The L^2 locality of the influence of the parameters s_k allows the same kind of action on several adjacent control points. For instance, if we decrease s_2, s_3 and s_4, the curve is pulled simultaneously toward P_2, P_3 and P_4,

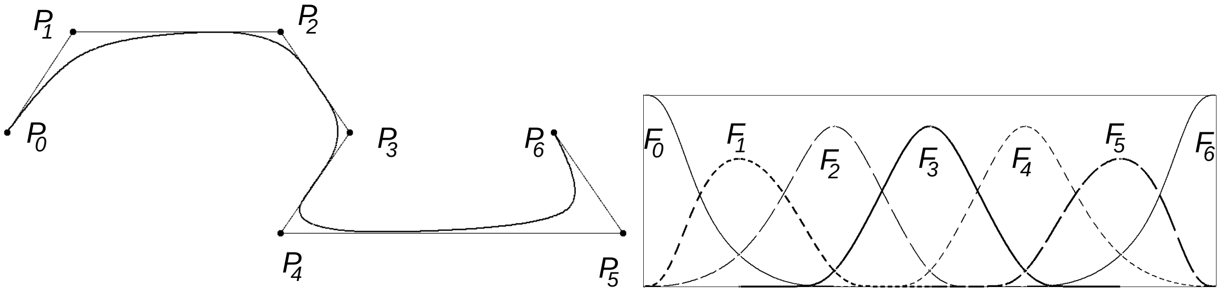

Figure 10: *Augmentation of the influence of P_2, P_3 et P_4*
$s_0 = 0, s_1 = 1, s_2 = 0.5, s_3 = 0.5, s_4 = 0.5, s_5 = 1, s_6 = 0$

and if we set the three parameters to zero, we obtain a sharp interpolation of P_2, P_3 and P_4 (see Figure 11). Finally, for the limit case where all the parameters s_k are set to zero, the curve merges with the control lattice (see Figure 12). But notice that the curve is *not* a linear spline because the parametrization is C^2 here, whereas it is only C^0 for linear splines.

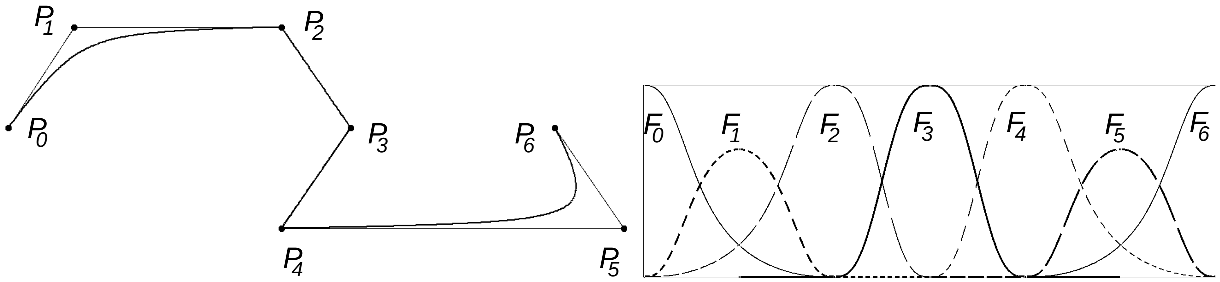

Figure 11: *Sharp interpolation of P_2, P_3 et P_4*
$s_0 = 0, s_1 = 1, s_2 = 0, s_3 = 0, s_4 = 0, s_5 = 1, s_6 = 0$

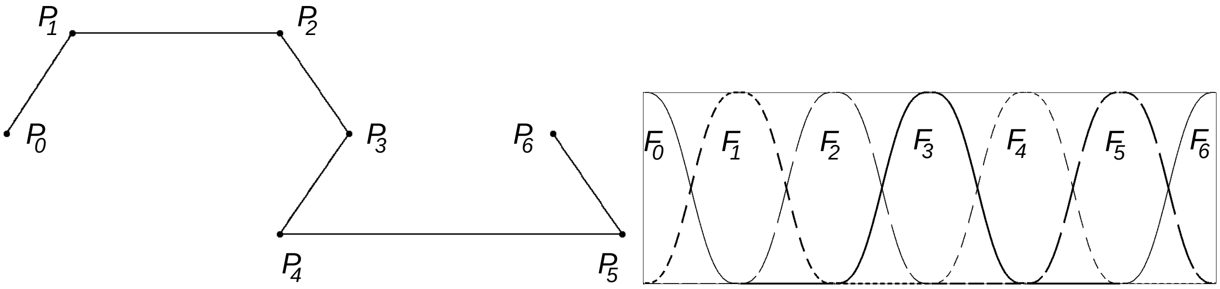

Figure 12: *Sharp interpolation of every control point*
$s_0 = 0, s_1 = 0, s_2 = 0, s_3 = 0, s_4 = 0, s_5 = 0, s_6 = 0$

This ability to mix smooth curves and sharp edges in an unrestricted way makes the extended X-spline model a candidate of choice for many applications. In vectorial font design, for instance, one switches frequenty between smoothness and sharpness. Therefore, the use of X-splines enables the design of characters with one single spline for the outline (plus eventually one spline for each hole) defined by a small number of control points (see left part of Figure 13)

To conclude this section, note that a very useful case is obtained when the control lattice forms a regular polygon and all the s_k are set to one: the resulting curve is a circle (see right part of Figure 13. In fact, this circle is only an approximated one but this approximation is so close (for 8 control points, the amplitude of the oscillations of the curve around the true circle represents less than a factor 10^{-3} of the radius, and for 12 control points, this variation is less than 10^{-6}) that it is sufficient for most of the applications[4]. Starting from that kernel case, other conics can be approximated as well with a similar accuracy [5].

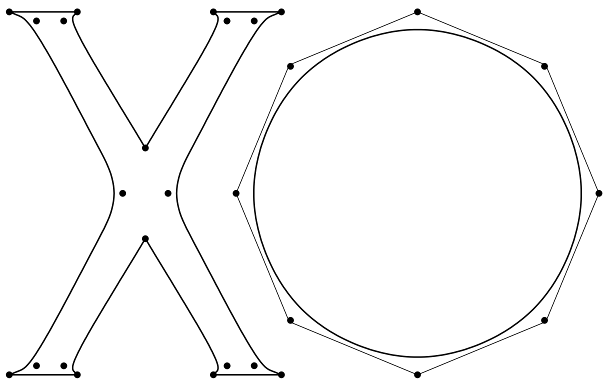

Figure 13: *Font design and representation of the circle*

[4] A similar result is obtained with B-splines [4], therefore it is not surprizing that it holds also for X-splines which approximate B-splines in that particular configuration.

5 General X-Splines

5.1 Formulation

As they have been formulated above, extended X-splines fulfill many of the properties listed in Section 2. Nevertheless, even if they allow interpolating one or several control points, extended X-splines are still approximation splines, because only sharp interpolations are provided. The last feature of our list was the ability to manipulate the same model either as an approximation spline or as an interpolation spline. The goal of this section is to show how this characteristic can be included in the X-spline model.

But, as recalled in Section 2, using interpolation splines implies forsaking the positivity of the blending functions and therefore the convex hull property. For some applications (and for some users), this is inconceivable. For that reason, we have purposely separated this extension from the previous section. So, the reader may choose between the formulation that fulfills the convex hull property and the formulation that provides the approximation/interpolation duality.

In Section 4, we saw that when the value of the parameter s_k is decreased, the blending function F^-_{k+1} (respectively F^+_{k-1}) becomes null between t_{k-1} and T^-_{k+1} (respectively T^+_{k-1} and t_{k+1}). At the limit case, when $s_k = 0$, F^-_{k+1} (respectively F^+_{k-1}) is null over the whole range $[t_{k-1}, t_k]$ (respectively $[t_k, t_{k+1}]$). Starting from that configuration of sharp interpolation, to get a "smooth" (G^2 continuity) interpolation of point P_k, we must allow F^-_{k+1} and F^+_{k-1} to become negative over these ranges. Moreover, in the same manner as we have sought to approximate cubic B-splines with the basic formulation, we will try to approximate cubic Catmull-Rom splines with this general formulation.

If we apply the following reparametrization to the curve,

$$u(t) = \frac{t - t_k}{t_{k+1} - t_k} = \frac{t - t_k}{\Delta} \tag{18}$$

we are assured that $u = -1$ at knot t_{k-1} where F^-_{k+1} gets negative, $u = 0$ at knot t_k where F^-_{k+1} gets positive, and $u = 1$ at knot t_{k+1} where F^-_{k+1} reaches its maximum. Therefore, we have to find two polynomials: $g(u)$ defined on $[0, 1]$ which represents the positive part of F^-_{k+1} and $h(u)$ defined on $[-1, 0]$ which represents its negative part. These two functions must join up at $u = 0$ with C^2 continuity. As in Section 3, we can derive a system of constraints but this time there are two functions, which means 12 constraints:

$$\begin{array}{lll} g(0) = 0 & g'(0) = q & g''(0) = 4q \\ g(1) = 1 & g'(1) = 0 & g''(1) = -2p \\ h(0) = 0 & h'(0) = q & h''(0) = 4q \\ h(-1) = 0 & h'(-1) = 0 & h''(-1) = 0 \end{array} \tag{19}$$

where q is a degree of freedom that controls the value of the first derivative at $u = 0$ (the same degree of freedom has been used by Duff in his *tensed interpolation spline* model [8]. All these constraints can be fulfilled by two quintic polynomials:

$$\begin{aligned} g(u) &= q\ u + 2q\ u^2 + (10-12q-p)\ u^3 \\ &\quad + (2p+14q-15)\ u^4 + (6-5q-p)\ u^5 \\ h(u) &= q\ u + 2q\ u^2 - 2q\ u^4 + q\ u^5 \end{aligned} \tag{20}$$

Starting from these equations, the same construction process detailed in Section 3 provides a rational quintic interpolation spline model that includes the properties of normality, locality and C^2 continuity. Moreover, the curves contain a degree of freedom q which allows modification of their shapes.

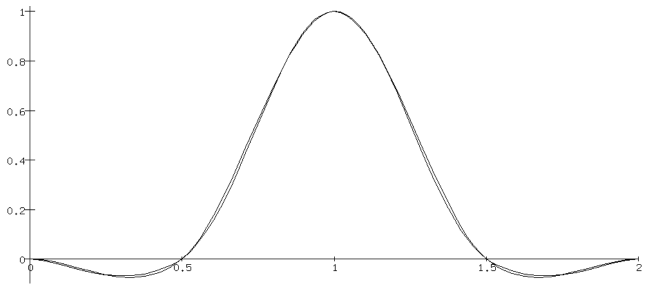

Figure 14: *Similarity of the cubic Catmull-Rom splines and the general X-splines blending functions (for $q = 1/2$)*

Two important remarks should be made about this model. First, as in every interpolation spline model, the regularity property is lost, thus the curve may have unwanted oscillations. We have observed experimentally that these oscillations can usually be avoided by limiting q to the range $[0, 1/2]$. Second, an interesting case is obtained for $q = 1/2$ because the blending functions are very close to the Catmull-Rom functions (see Figure 14). But it should be noticed that the new functions are C^2 continuous instead of C^1.

The final step of the construction of our new spline model will be to merge the parameter s of the approximation model and the parameter q of the interpolation one. Here again, the goal is to simplify the degrees of freedom manipulated by the end user. Practically, only one shape parameter s_k per control point P_k will be used. This is done with the following convention:

- When the user sets all s_k in the range $[0, 1]$, it means that he wants to manipulate approximation splines. In that case, s_k is the curve/lattice distance parameter defined in Section 4 (in particular, uniform cubic B-splines are approximated for $s_k = 1$).
- When the user sets all s_k in the range $[-1, 0]$, it means that he wants to manipulate interpolation splines. In that case, q_k is obtained from s_k by $q_k = -s_k/2$ (so, $s_k = -1$ provides $q = 1/2$ which approximates cubic Catmull-Rom splines).

The positive/negative distinction for s_k indicates clearly that there is a breaking point: for positive s_k, the convex hull property is fulfilled, for negative s_k, it is not the case anymore. On the other hand the intuitive notion of curve/lattice distance is preserved even for negative s_k. Indeed, as we will see below, the more s_k departs from zero, the more the curve departs from the control lattice.

5.2 Examples

We already know that a "sharp" (G^0 continuity) interpolation of the control lattice can be obtained by setting all s_k to zero (see Figure 12). If we want to realize a "smooth" (G^2 continuity) interpolation, the only thing to do is to set these parameters to negative values. For instance, by setting all s_k to -1, an interpolation spline almost identical to the Catmull-Rom spline is obtained (compare Figure 15 and Figure 2). As expected, the blending functions become partly negative, and thus the convex hull property is lost.

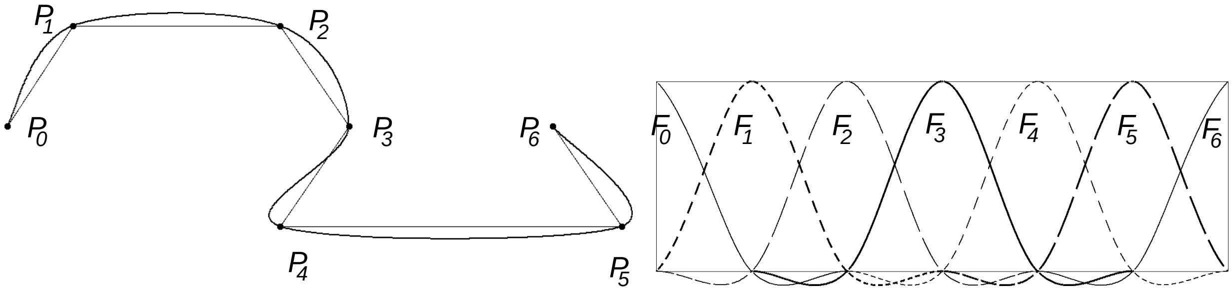

Figure 15: *Smooth interpolation of every control point*
$s_0 = s_1 = s_2 = s_3 = s_4 = s_5 = s_6 = -1$

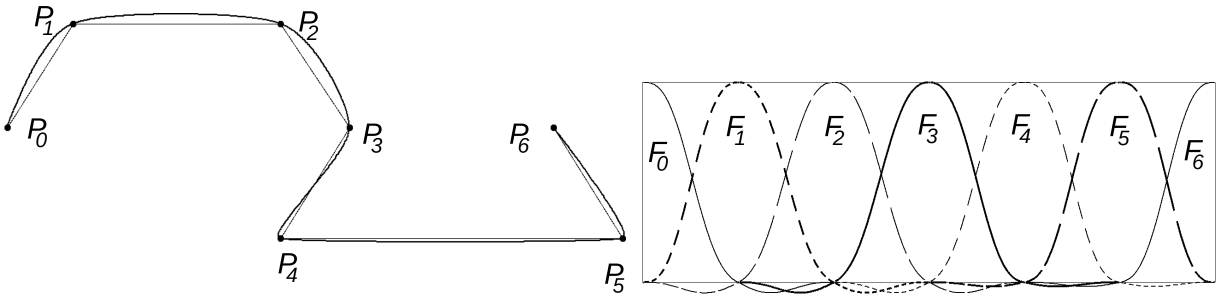

Figure 16: *Modification of the interpolation*
$s_0 = 0, s_1 = s_2 = s_3 = -1, s_4 = s_5 = -0.5, s_6 = 0$

By providing different values for the parameter s_k, the shape of the interpolation curve can be controlled precisely. For instance, one can enable very slack interpolation for a specific zone of the lattice and a much tighter interpolation for another zone (see Figure 16). And finally, what is perhaps the most interesting feature of the X-spline model, one can combine without any restriction, positive and negative shape parameters s_k in order to create approximation zones and interpolation ones in the same curve (see Figure 17).

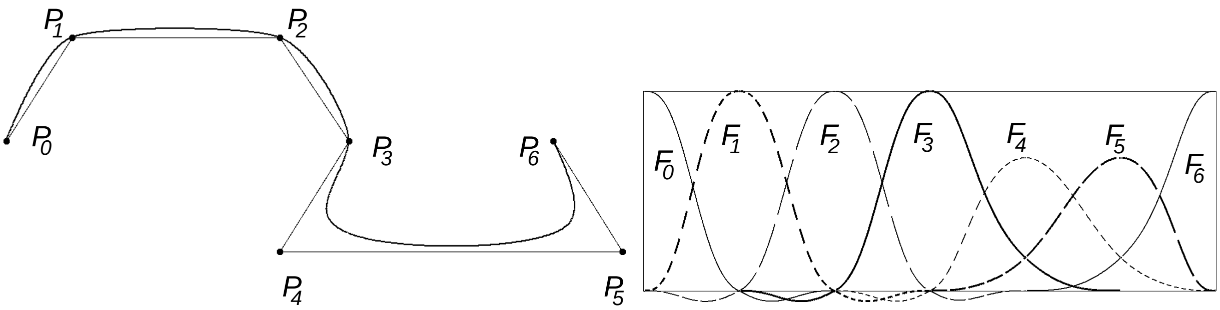

Figure 17: *Approximation zones and interpolation zones*
$s_0 = 0, s_1 = s_2 = s_3 = -1, s_4 = s_5 = 1, s_6 = 0$

6 Surfaces

The extension of the new model from curves to surfaces is straightforward. The only thing to do is to compute the tensor product of two non-normalized X-spline curves and then to apply the normalization step[5]. The characteristic of the X-splines to create all possible geometric effects by using only uniform knot vectors is vital here because, as we have recalled in Section 1, effects due to knot manipulations (e.g. sharp edges for B-splines) are propagated along the whole isoparametric curves. On the contrary, the shape parameters of the X-spline model are directly related to the control points and thus can be localized precisely on a given zone of the surface.

Because of the tensor product, two shape parameters r_k and s_k are provided for each control point P_k where r_k acts in the u direction of the surface and s_k acts in the v direction. A nice consequence is that non-isotropic manipulations are allowed (for instance, creating sharpness in one direction and smoothness in the other one). As a counterpart, the behaviour of these parameters is a bit more subtle than previously:

- $r_k > 0,\ s_k > 0$: P_k is a C^2/G^2 approximation point
- $r_k = 0,\ s_k = 0$: P_k is a C^2/G^0 interpolation point
- $r_k < 0,\ s_k < 0$: P_k is a C^2/G^2 interpolation point
- $r_k = 0,\ s_k > 0$: P_k is an approximation point providing C^2/G^0 continuity in u and C^2/G^2 continuity in v

[5] This process is sometimes called *generalized tensor product* [11]

- $r_k = 0$, $s_k < 0$: P_k is an interpolation point providing C^2/G^0 continuity in u and C^2/G^2 continuity in v

Figure 18 and Figure 19 shows some examples of X-spline surfaces. You should notice the ability to create interpolation of adjacent control points, localized sharp edges as well as soft transitions between sharp and smooth zones; three features that are impossible (or at best, only possible in specific cases) with any existing spline model.

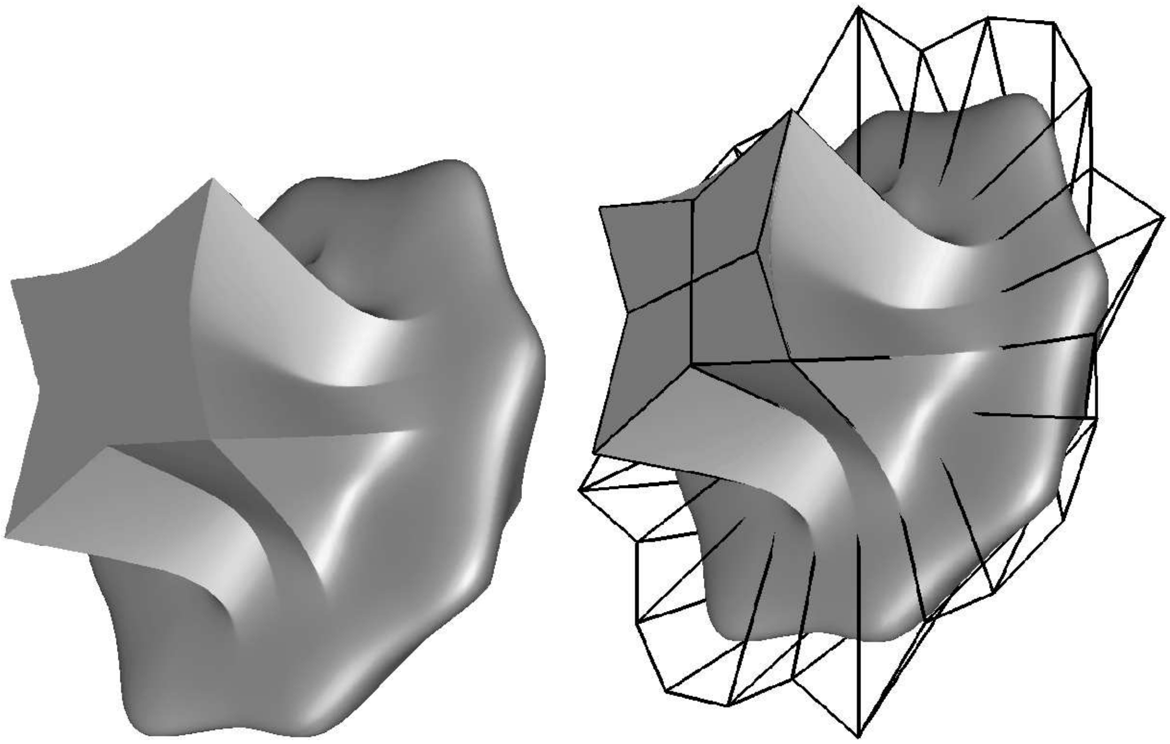

Figure 18: *Sharp extrusion from a smooth object*

Note that the star-shaped flat face on the top of the object is composed of two sides with straight edges (left and bottom) and two sides with rounded edges (top and right). Straight sides create sharp edges that are propagated all along the extrusion whereas the sharp edges smoothly vanish when they come near the rounded sides of the top face.

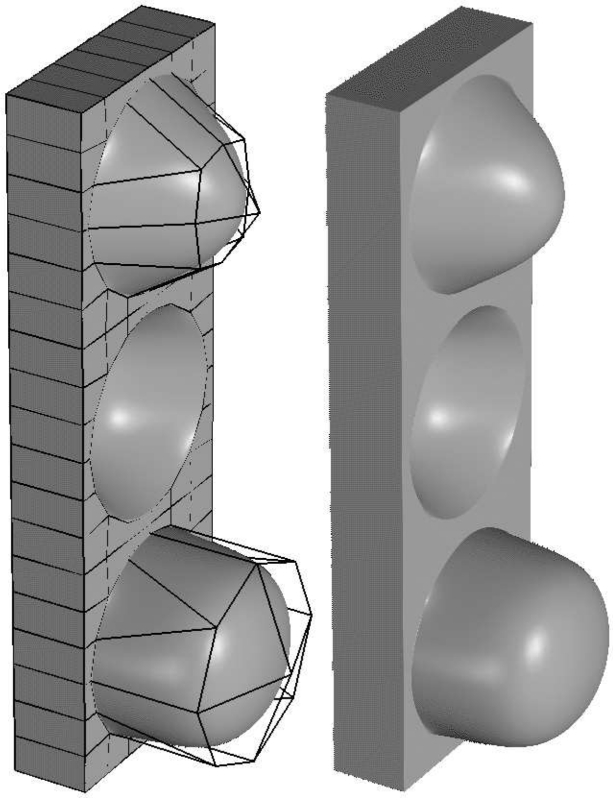

Figure 19: *Smooth extrusion from a sharp object*

7 Conclusion

In this paper, we have presented a new model of spline curves and surfaces. This model includes many classical properties such as affine and perpective invariance, convex hull, variation diminution, local control and C^2/G^2 or C^2/G^0 continuity, as well as some original features such as a continum between (an approximation of) B-splines and (an approximation of) Catmull-Rom splines, or the ability to define approximation zones and interpolation zones in the same curve or surface. These properties have been obtained by defining a new family of blending functions that are quintic rational polynomials and introducing an original shape parameter that provides, for each control point, a smooth transition between approximation, sharp interpolation and smooth interpolation.

This paper is only intended as an initial presentation of X-splines. For space limitations, several topics could not be included here. We propose some additional results in [5] which should be considered as the companion paper of this one. More precisely, the following topics are discussed in it:

- *Some precisions on efficient implementation of X-splines:* For instance, one can show that even if they are quintic, rational and provide more geometrical effects, uniform X-splines are less expensive to compute than non-uniform cubic B-splines).
- *Lower order and higher order X-splines:* Quintic polynomials have been chosen here because we sought for C^2/G^2 continuity, but in fact a similar construction process can be used for any polynomial of degree $2k+1$ providing splines with C^k/G^k continuity.
- *Extension to non-uniform knot vectors:* Geometrical effects generated by non-uniformity in classical splines can be created by the shape parameters, so this extension is not that vital. Nevertheless, non-uniform knots may be useful for key-frame animation or data-fitting.
- *Refinement algorithms:* This is clearly a much harder task. For the moment, we propose only some preliminary results on a kind of De Casteljau subdivision algorithm.

Acknowledgements

We wish to thank all the anonymous reviewers for many helpful comments and suggestions. We also thank Brian Smith (from Lawrence Berkeley Laboratory), the maintainer of the `xfig` package, who gave us the permission to include the X-splines model in his software and to use it for public demonstration. Finally, special thanks to C. Feuille, S. Grobois, L. Mazière and L. Minihot who modified `xfig` for us and discovered the L^2 (rather than L^4) locality of the shape parameters.

Bibliography

[1] B. Barsky, *The Beta-Spline: a Local Representation based on Shape Parameters and Fundamental Geometric Measures*, PhD Thesis, University of Utah, 1981.

[2] R. Bartels, J. Beatty, B. Barsky, *An Introduction to Splines for Computer Graphics and Geometric Modeling*, Morgan Kaufmann, 1987.

[3] C. Blanc, *Techniques de Modélisation et de Déformation de Surfaces pour la Synthèse d'Images*, PhD Thesis, Université Bordeaux I, 1994 (in french).

[4] C. Blanc, C. Schlick, *More Accurate Representation of Conics by NURBS*, Technical Report, LaBRI, 1995 (submitted for publication).

[5] C. Blanc, C. Schlick, *X-Splines: Some Additional Results*, Technical Report, LaBRI, 1995 (available by HTTP at `www.labri.u-bordeaux.fr/LaBRI/People/schlick`).

[6] E. Catmull, R. Rom, *A Class of Interpolating Splines*, in Computer Aided Geometric Design, p317-326, Academic Press.1974

[7] E. Cohen, T. Lyche, R. Riesenfeld, *Discrete B-Splines and Subdivision Techniques*, Computer Graphics & Image Processing, v14, p87-111, 1980.

[8] T. Duff, *Splines in Animation and Modelling*, SIGGRAPH Course Notes, 1986.

[9] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, 1990.

[10] D. Forsey, R. Bartels, *Hierarchical B-Spline Refinement*, Computer Graphics, v22, n4, p205-212, 1988.

[11] L. Piegl, *On NURBS: a Survey*, Computer Graphics & Applications, v11, n1, p55-71, 1991.

[12] R. Riesenfeld, *Applications of B-Spline Approximation to Geometric Problems of Computer Aided Design*, PhD Thesis, University Syracuse, 1973.