Constraint-Based LN-Curves

Young Joon Ahn[*]
Department of Mathematics Education
Chosun University, Gwangju
501-759, Korea
ahn@chosun.ac.kr

Christoph M. Hoffmann [†]
Purdue University
West Lafayette, IN
47907, USA
cmh@purdue.edu

ABSTRACT

We consider the design of parametric curves from geometric constraints such as distance from lines or points and tangency to lines or circles. We solve the Hermite problem with such additional geometric constraints. We use a family of curves with linearly varying normals, LN curves, over the parameter interval $[0, u]$. The nonlinear equations that arise can be of algebraic degree 60. We solve them using the GPU on commodity graphics cards and achieve interactive performance. The family of curves considered has the additional property that the convolution of two curves in the family is again a curve in the family, assuming common Gauss maps, making the class more useful to applications. We also remark on the larger class of LN curves and how it relates to Bézier curves.

Keywords

Geometric constraints, LN-curves, MCAD, CAGD, GPU programming, convolution.

1. INTRODUCTION

Constraint-based sketching is a major design paradigm in mechanical computer-aided design (MCAD): A rough sketch is prepared by the user and is annotated with geometric constraints such as distance, angle, tangency, concentricity, etc. The sketch is then instantiated to the precise specifications implied by the constraints, and is interpreted as a profile. The instantiation, by geometric constraint solving, enables generic design, feature libraries, convenient redesign, and design variation. Constraint solving is therefore of fundamental importance. Literature reviews include [17,11].

In Computer-Aided Geometric Design (CAGD), on the other hand, curves are designed subject to constraints of interpolating points, curve segments meeting with tangent or higher-order continuity, and shape design subject to fairness criteria. For an introduction see, e.g., [7,16]. This different way of constraint-based design of curves and surfaces has a markedly different vocabulary with little or no overlap of constraint operations familiar from MCAD. Algorithms for approximation, matching continuity to various degree, etc., constitute a very different form of constraint solving and with a different shape vocabulary.

MCAD style constraint solving typically restricts to the shape vocabulary of points lines and circles, and to geometric and dimensional constraints between them. This restriction is rooted in part in the absence of suitable techniques for solving the algebraic problems that underlie more complicated specifications. Extending the vocabulary of MCAD constraint solvers, therefore, follows one of two strategies: identifying a tractable class of shapes, or extending the algebraic techniques needed to solve more complex sub problems. Our paper seeks to narrow the gap between MCAD and CAGD employing both strategies: We focus on the class of LN curves [20,22] that play an important role in convolution, and we develop novel approaches to solving the algebraic equation systems that arise, exploiting the GPU. Elsewhere, we have shown how to include, in the solver vocabulary, conic arcs [8] and Tschirnhaus cubic Bézier curves [14]. There is also work by others on extending the vocabulary, such as [5] that designs rational cubic Bézier curves with monotone curvature. We refer the reader to the survey [4] for additional information.

Minkowski sums have many applications, for example in motion planning, NC machining, and in offset computations. Curve and surface convolutions are often employed to compute Minkowski sums. However, as noted in [19], the convolution of two rational curves is, in general, not a rational curve. Therefore, the subclass of LN curves and surfaces has received attention in the literature because they allow an exact parameterization of their convolutions; [21]. This is another motivation for our work.

We begin our paper with the usual preliminaries and definitions in Section 2, where we also review a characterization of LN curves from the literature. Based on this characterization, we explain how to solve the Hermite interpolation problem with G^1-continuity between consecutive interpolants. For the Hermite problem, cubic LN curves suffice when no additional constraints are imposed and continuity is G^1. But when we stipulate that the interpolating arcs be tangent to a given line or a circle, to be discussed in Sections 3 and 4, quartic interpolants will be needed working with the normal form. This extended constraint problem is linear for an additional line tangency, owing to the

[*]Tel: +82-62-230-7331
[†]Tel: 1-765-494-6185

SAC'10 March 22-26, 2010, Sierre, Switzerland.

nature of the interpolants. When tangency to circles is required, however, the problem is strongly nonlinear. Nonlinear problems pose practical difficulties which we address by exploiting the native graphics hardware, so allowing efficient solutions to problems that otherwise would demand time-consuming iteration. We discuss this GPU approach to solving the equation system and, in Section 5, report the resulting performance. Section 6 shows how to use our curve class for convolutions that result again in LN curves. Section 7 finally discusses how we can relax the requirements that the parameter interval for the normal form be the interval $[0, u]$.

2. PRELIMINARIES AND THE HERMITE PROBLEM

We consider polynomial parametric plane curves $K(t) = [x(t), y(t)]$ where the coordinate functions are polynomials in t with real coefficients. We consider a Hermite interpolation problem that asks to interpolate a sequence of points P_k in the plane with a set of parametric curve arcs C_k such that consecutive arcs meet with tangent continuity at the interpolated points. The tangent directions at the points are prescribed. We require G^1-continuity, but not C^1-continuity.

A parametric curve $K(t')$ is LN if there is a parameterization $t = s(t')$ such that the curve normal at $K(t)$ is $\overrightarrow{q}t + \overrightarrow{p}$, where $\overrightarrow{q}$ and $\overrightarrow{p}$ are vectors. In [20] it is proved that such curves can be characterized by a rational function $f(t)$ with the following properties:

1. The curve is given by $K(t) = [-f_t, -f + tf_t]$.
2. The curve tangent, at $K(t)$ has the equation $f(t) + tx + y = 0$.
3. The curve normal at $K(t)$ is $(t, 1)$.

In the following, we will work with LN curves for which the function f is not rational.

Proposition 1. *A non-rational degree three LN curve $K(t)$ solves the Hermite interpolation problem between the origin $[0, 0]$ with normal $(0, 1)$ and the point $[x_1, y_1]$ with normal $(u, 1)$ and is not singular in the closed interval $[0, u]$ if and only if*

$$-\frac{3y_1}{2x_1} < u < -\frac{3y_1}{x_1}.$$

Proof. Let $f(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$. The curve $K(t)$ interpolates the end points with the required tangents if and only if

$$\begin{aligned} a_0 &= 0 \\ a_1 &= 0 \\ -2a_2 u - 3a_3 u^2 &= x_1 \\ a_2 u^2 + 2a_3 u^3 &= y_1 \end{aligned} \quad (1)$$

The system (1) is linear since the normal $(u, 1)$ is given at the end point $[x_1, y_1]$. By algebra,

$$\begin{aligned} a_2 &= (-2ux_1 - 3y_1)/u^2 \\ a_3 &= (ux_1 + 2y_1)/u^3 \end{aligned}$$

$K(t)$ is singular at t^* when $f_{tt}(t^*) = 2a_2 + 6a_3 t^* = 0$. Hence $t^* = -a_2/3a_3$ and

$$t^* = \frac{u}{3} \cdot \frac{2ux_1 + 3y_1}{ux_1 + 2y_1}.$$

We want to establish $t^* < 0$ and $t^* > u$ to ensure that $K(t)$ has no singularity in $[0, u]$. The first inequality is established from $J = (2ux_1 + 3y_1)(ux_1 + 2y_1) < 0$. With $s = -y_1/x_1$, we obtain $J = (u - 3/2s)(u - 2s)$ so that

$$J < 0 \quad \text{iff} \quad \frac{3}{2}s < u < 2s.$$

The inequality $t^* > u$ is established from

$$\frac{u}{3} \cdot \frac{2ux_1 + 3y_1}{ux_1 + 2y_1} > u$$

or, equivalently, $(2ux_1 + 3y_1)(ux_1 + 2y_1) > 3(ux_1 + 2y_1)^2$. This simplifies to $(u - 3s)(u - 2s) < 0$ which means

$$2s < u < 3s.$$

Now for $u = 2s$ we obtain $a_3 = 0$; i.e., the curve is quadratic and has no singularity. Thus the nonsingular range is given by

$$-\frac{3y_1}{2x_1} < u < -\frac{3y_1}{x_1}.$$

QED. $\square$

Remark. Geometrically, the bound relates the tangent of the turn angle α to the tangent of angle β. See also Fig.1.

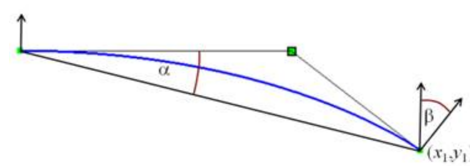

Figure 1: Cubic Nonsingularity Condition: $1.5\tan(\alpha) < \tan(\beta) < 3\tan(\alpha)$.

3. THE LINE TANGENCY PROBLEM

3.1 Line Tangency

We consider the Hermite interpolation problem for quartic LN curves where, as additional constraint, a line $L : mx + y + b = 0$ is given and the interpolant is to be tangent to L. Since the (polynomial) cubic Hermite interpolant has no additional degrees of freedom, a quartic LN curve is needed. Again we choose the coordinate system such that the two end points are $[0, 0]$ and $[x, y]$, and the respective normals $(0, 1)$ and $(u, 1)$. The normal at the tangency to L is $(m, 1)$, and since the interpolant is LN, tangency must be at the point $K(m)$. The equations on the coefficients are therefore the system (H1) augmented by the equation:

$$f(m) = b \quad (2)$$

The equations are linear and determine the LN interpolant. Note that the normal of line L, $(m, 1)$, determines the parameter value m at which the curve must be tangent. The correctness of the equation follows immediately from the canonical tangent representation $f(m) + mx + y = 0$.

3.2 Conditions for Nonsingularity

We will analyze when the quartic LN interpolant is nonsingular. Let $[x_1, y_1]$ be the end point. From the equation systems (1) and (2) we obtain

$$a_3 = \frac{-(2a_2u^2 + 3ux_1 + 4y_1)}{u^3}, a_4 = \frac{a_2u^2 + 2ux_1 + 3y_1}{u^4} \quad (3)$$

So the derivative can be written as

$$K'(t) = \left(\frac{-2g_4(t)}{u^4}, \frac{2tg_4(t)}{u^4} \right)$$

Where $g_4(t) = 6t^2(a_2u^2 + 2ux_1 + 3y_1) - 3t(2u^3a_2 + 3u^2x_1 + 4uy_1) + a_2u^4$. If the discriminant D of the quadratic equation $g_4(t) = 0$ is negative, then the curve has no singularity in the range $[0, u]$.

PROPOSITION 2. *If $(3 - \sqrt{3})s < u < (3 + \sqrt{3})s$, where $s = (-y_1)/x_1$, then there exists at least one quartic LN curve that is not singular in the interval $[0, u]$.*

PROOF. Consider $D < 0$, that is, $4u^4a_2^2 + (20u^3x_1 + 24y_1u^2)a_2 + (27u^2x_1^2 + 72ux_1y_1 + 48y_1^2) < 0$. Using dD/da_2, the quadratic polynomial $D(a_2)$ has the minimum at $a_2 = -(5ux_1 + 6y_1)/2u^2$ with the value $6u^2x_1^2 + 36ux_1y_1 + 36y_1^2$. Let

$$B_0 = \frac{6s - 5u}{(2u^2)/x_1}$$

and take $a_2 = B_0$, then $D(a_2) < 0$ iff $6u_1^2x_1^2 + 36ux_1y_1 + 36y_1^2 < 0$, or equivalently, $u^2 - 6us + 6s^2 < 0$. Factoring the left-hand side, this means $D(a_2) < 0$ iff $(u - (3 + \sqrt{3})s)(u - (3 - \sqrt{3})s) < 0$; that is, iff $(3 - \sqrt{3})s < u < (3 + \sqrt{3})s$. So, if $D < 0$, then we may choose

$$a_2 \in \left(B_0 - \frac{\sqrt{D_1}}{2u^2/x_1}, B_0 + \frac{\sqrt{D_1}}{2u^2/x_1} \right). \tag{4}$$

where $D_1 = -2(u-(3-3)s)(u-(3+3)s)$ is the discriminant of the quadratic equation $D(a_2) = 0$. QED. $\square$

We can strengthen Proposition 2 to

PROPOSITION 3. *There is at least one quartic LN curve that is not singular in the interval $[0, u]$ iff $(3 - \sqrt{3})s < u < (3 + \sqrt{3})s$.*

The proof proceeds by analyzing when the curve has a singularity in the interval $[0, u]$ and is omitted.

3.3 Implementation

In our implementation, the user draws interactively the poly-arc $[P_0, Q_0, Q_1, \cdots Q_{n-1}, P_n]$. The segments $[Q_k, Q_{k+1}]$ are subdivided each by inserting a point P_k as explained later. Now each point triple $[P_k, Q_k, P_{k+1}]$, $0 \leq k < n$, defines a LN curve between P_k to P_{k+1} with the respective normals perpendicular to the line segments $[P_k, Q_k]$ and $[Q_k, P_{k+1}]$. For each segment the user defines, in addition, a line L to which the arc should be tangent.

The insertion of $P_2 \cdots P_{n-1}$ implies that consecutive LN arcs are G^1-continuous. Insertion can place P_k as midpoint between Q_k and Q_{k+1}, or by the ratio of turning angles. In the latter case, let u be the distance $u = d(Q_k, P_k)$ and $v = d(P_k, Q_{k+1})$. Let α_k be the complement of the angle $\alpha'_k = \angle P_k, Q_k, P_{k+1}$; i.e., $\alpha_k = \pi - \alpha'_k$. Then we require $v/u = |\alpha_k/\alpha_{k+1}|$. The user may also modify the partition of the segment $[Q_k, Q_{k+1}]$ manually. An example of the two schemata is shown in Figure 2.

Intuitively, line tangency can be used to manipulate the parameter speed and with it the turning rate of of the curve normal. Alternatively, it may represent an additional contact constraint.

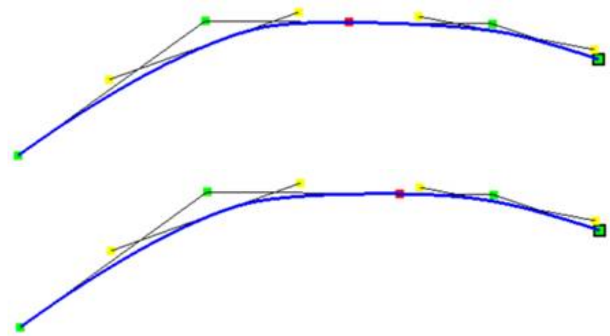

Figure 2: Quartic LN spline; midpoint division above, angle ratio below.

4. THE TANGENCY TO A CIRCLE PROBLEM

4.1 Defining Equations

We exploited the LN property to find a linear equation for the tangency to the given line. When tangency to a circle is required, there is no a-priori parameter value at which tangency would be achieved, thus the equations become nonlinear. Let $O = (O_x, O_y)$ be the center of the circle and r its radius. We consider the radius a signed quantity, indicating whether tangency should be on the convex or the concave side of the arc. Let m be the parameter value at which the interpolating LN curve is tangent to the circle, and let $K(m) = [x(m), y(m)]$. Since m is unknown, we need to formulate two equations. The first equation states that the point $K(m)$ is on the circle:

$$(x(m) - O_x)^2 + (y(m) - O_y)^2 - r^2 = 0. \tag{5}$$

Now the normal at $K(m)$ is $(m, 1)$, and it is either parallel or anti-parallel to the radius $(O, K(m))$, depending on which side the circle should lie. So

$$x(m) - O_x = m(y(m) - O_y). \tag{6}$$

Recall the expressions for a_3 and a_4 from equation (3). Substitution into equations (5) and (6) yields two equations with unknowns m and a_2. The algebraic degree of equation (5) is ten and of equation (6) is six, after this substitution. Note that these are linear substitutions, so we do not raise the algebraic degree of the two nonlinear equations. We need to solve the system in order to satisfy the tangency constraint.

The algebraic problem is of degree 60 and therefore demanding. We expect that many of the roots of the bivariate system are complex or out of the range of interest: We are only interested in solutions where the circle touches the curve in the parameter range, so that $m \in [0, u]$. Moreover, from equation (4) we have an estimated range for the value of a_2 for a nonsingular solution. So, we seek an approach that makes use of this information.

4.2 Solving the Equations

The two equations define implicit algebraic curves. Since we have range estimates for the solution(s) of interest, we can evaluate both curves using a continuation method such as marching squares and so find approximate solutions. However, a sequential marching approach will be time-consuming, so we seek to parallelize the computation and use the GPU to carry it out. Given the algebraic degrees of the two curves, accurate evaluation of the points (m, a_2) requires some care. We choose to evaluate the polynomials by first evaluating the coefficients a_3 and a_4 followed by evaluating repeated and common sub expressions. Next, we sample the values of equation (6) on a grid in the domain of interest, by rendering

$$\Phi(m, a_2) = x(m) - O_x - m(y(m) - O_y) \tag{7}$$

on a raster of size 1K by 1K pixels. Positive values of Φ are rendered red, negative values black. Then, we render equation (5) in the same manner, with blue for positive and black for negative values. Note that the two colors use separate channels, so that the resulting raster has pixels that are red, blue, black or cyan. Pixels that are at the boundary of both curves and are therefore near the actual intersection of the two curves are discovered by applying a local 2×2 mask and selecting the center of the mask when the four pixels have more than two colors. All this is done in parallel on the graphics hardware. The resulting solution(s) can then be refined to actual intersections using, say, Newton iteration.

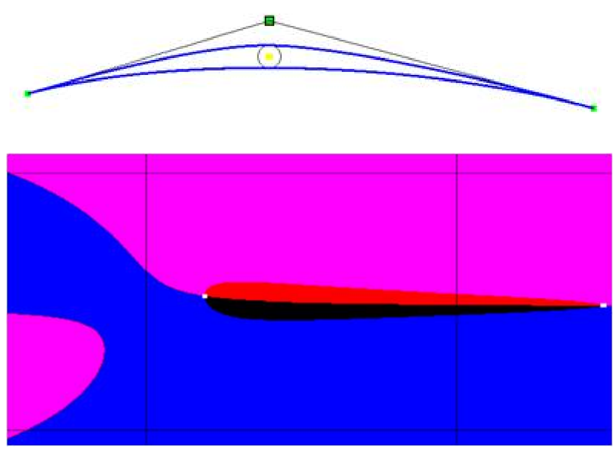

Figure 3: Two solutions of a circle tangency (above) and associated raster (below). The two solutions are shown as white dots. The raster image has been clipped.

5. IMPLEMENTATION AND RESULTS

We implemented our algorithms and experimented with the constraints. We also timed the performance on a desktop PC running Windows Vista (32bit) with the following configuration: Intel Xeon X5460 CPU at 3.16GHz, 4GB main memory, and an nVidia GeForce GTX 285 graphics card driving a display with 2560x1600 pixels. The program was run in release mode alongside other applications including Outlook and web browser windows.

Single curves (three control points) allow interactive performance at 60 frames per second (fps), both without additional constraints (cubic arc) as well as for line tangency (quartic arc). Since this is a linear problem, there is no GPU involvement. Performance stayed the same, at close to 60 fps, when circle constraints are imposed. The nonlinearity of the problem is compensated for by the rasterization in parallel, by the GPU, using an accuracy of 1K by 1K pixels. Newton iteration was not implemented. When interacting with multiple segments, performance did not change.

We believe that those performance numbers can be more than doubled in view of prior work in which we also extracted information from rasters of this size. To achieve higher frame rates, the interaction between CPU and GPU needs to be restructured in the implementation.

6. CONVOLUTION

So far, we have considered a subset of LN curves, not the full set of LN curves. This smaller family of LN curves has the following special property:

PROPOSITION 4. *For any two curves in the family of LN curves, if they have the same Gauss map, then their convolution curve is polynomial.*[1]

We illustrate the property of the family of LN curves. Consider two curves in the family, $\mathbf{c}_1(t)$ and $\mathbf{c}_2(t)$, satisfying that they start at $[-1.5, 1]$ and $[1, -0.1]$ with the common normals $(-\tan(\pi/3), 1)$. Assume that they end at [0.5,2.7] and [3.1,1], respectively, with the common end normal $(-\tan(\pi/12), 1)$, as shown in figure 4. Let $\theta = \pi/3$. Using rotation and translation, and Equation (1), we obtain the cubic LN curves $\mathbf{c}_1(t) = [x(t), y(t)]$ and $\mathbf{c}_2(t) = [X(t), Y(t)]$:

$$\begin{aligned}\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} &= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} -f_t(t) \\ -f(t) + tf_t(t) \end{pmatrix} + \begin{pmatrix} -1.5 \\ 1 \end{pmatrix} \\ \begin{pmatrix} X(t) \\ Y(t) \end{pmatrix} &= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} -F_t(t) \\ -F(t) + tF_t(t) \end{pmatrix} + \begin{pmatrix} 1 \\ -.1 \end{pmatrix}\end{aligned}$$

for $t \in [0, 1]$, where

$$\begin{aligned} f(t) &= -2.2983t^2 + .7081t^3 \\ F(t) &= -.1993t^2 - .5347t^3. \end{aligned}$$

Their derivatives can be obtained as

$$\begin{aligned}\begin{pmatrix} x'(t) \\ y'(t) \end{pmatrix} &= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} -f_{tt} \\ tf_{tt} \end{pmatrix} \\ \begin{pmatrix} X'(t) \\ Y'(t) \end{pmatrix} &= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} -F_{tt} \\ tF_{tt} \end{pmatrix}.\end{aligned}$$

The two cubic LN curves $[x(t), y(t)]$ and $[X(s), Y(s)]$ have the same normal at $t = s$, since

$$\begin{aligned}\frac{y'(t)}{x'(t)} = \frac{Y'(s)}{X'(s)} \quad &\Leftrightarrow \quad \frac{-\sin\theta + t\cos\theta}{-\cos\theta - t\sin\theta} = \frac{-\sin\theta + s\cos\theta}{-\cos\theta - s\sin\theta} \\ &\Leftrightarrow \quad t = s\end{aligned}$$

Thus the convolution curve is

$$\begin{aligned}(\mathbf{c}_1 * \mathbf{c}_2)(t) &= [x(t), y(t)] + [X(t), Y(t)] \\ &= \begin{bmatrix} -.5 + 2.4976t + 1.9028t^2 - .3004t^3 \\ .9 + 4.3260t - 1.6995t^2 + .17346t^3 \end{bmatrix}^T\end{aligned}$$

for $t \in [0, 1]$. Therefore we can see that the convolution of any two curves containing the family of LN curves is polynomial.

[1]For more about Gauss maps see [19].

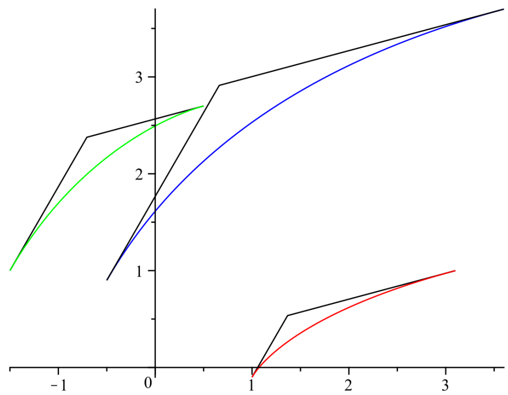

Figure 4: A family of Cubic LN curves: The convolution of two curves $\mathbf{c}_1(t)$(green) and $\mathbf{c}_2(s)$(red) in the family of LN curves in this paper is also polynomial curve $\mathbf{c}_1 * \mathbf{c}_2$(blue).

7. GENERAL PARAMETER INTERVALS

We have restricted to LN curve segments that are over the interval $[0, u]$, the customary choice for CAGD. However, due to the nature of linear normals and the normal form representation we adopted, as reviewed in Section 2, the theory so derived is less flexible than it could be. If we allow a general parameter interval, then we can work with a larger family of LN curves. That is, the choice of a coordinate system in which the curve begins at the origin with a tangent in the positive x-direction constitutes a restriction. We now remark on some of the consequences when relaxing this assumption.

Assume that the curve arc is over the interval $[u_0, u_1]$, where $u_0 < u_1$, and consider the Hermite problem where we are given the end points $P_0 = [x_0, y_0]$ and $P_2 = [x_2, y_2]$ and the end tangents t_0 and t_2. We extend the lines defined by the points and their tangents to obtain the intersection $P_1 = [x_1, y_1]$, as shown in Figure 5 (left).

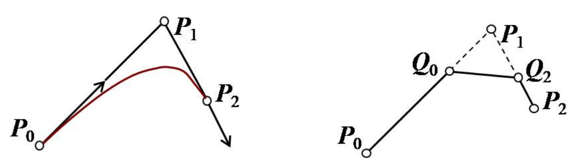

Figure 5: Left: Quadratic Bézier curve. Right: Cubic control polygon construction

The quadratic Bézier curve defined by the three points is, of course, an LN curve, although re-parameterization will be necessary to make that fact explicit. Obtaining the normal form of Section 2, moreover, requires that the parameter interval be larger than $[0, u_1]$ and that the x-coordinates satisfy $P_0^x < P_1^x < P_2^x$. For such (extended quadratic) LN curves, the slope $y'(t)/x'(t)$ will be linear. When extending to cubic LN-curves, again with the larger interval $[u_0, u_1]$, the slope $y'(t)/x'(t)$ of the nonquadratic curves will be fractional linear. We can prove conditions under which such cubic LN curves are nonsingular in the parameter interval.

Proposition 5. *Let $[P_0, P_1, P_2]$ be defined as in Figure 5, with $P_0^x < P_1^x < P_2^x$, and construct the cubic Bézier curve with control points $[P_0, Q_0, Q_2, P_2]$, where*

$$\begin{aligned} Q_0 &= (1 - r_0)P_0 + r_0 P_1 \\ \text{and} \quad Q_1 &= (1 - r_2)P_2 + r_2 P_1. \end{aligned}$$

We choose the partition values r_0 and r_2 such that, for some $t = \tau$, we have $x'(\tau) = y'(\tau) = 0$. This means that

$$r_0 = \frac{2\tau}{3\tau - 1} \qquad \text{and} \qquad r_2 = \frac{2(\tau - 1)}{3\tau - 2}$$

Under these conditions, the resulting cubic Bézier curve is LN. The curve degenerates to a quadratic curve when $\tau = \infty$, i.e., when $r_0 = r_2 = 2/3$. Moreover, the cubic LN curve has a singularity in the range $[0, 1]$ iff $\tau \in [0, 1]$.

Thus, using this construction and choosing $\tau \in (-\infty, -1) \cup (1, +\infty)$, we obtain nonsingular cubic LN curves for the basic Hermite problem. Working in this larger class of LN curves, we can also solve the constraint problems requiring tangencies to given lines and circles with cubic curves.

Acknowledgements: This work has been supported in part by NSF Grant CPATH CCF-0722210, DOE award DE-FG52-06NA26290, and by a gift from Intel Corp.

8. REFERENCES

[1] W. Bouma, I. Fudos, C. Hoffmann, J. Cai, R. Paige. "A Geometric Constraint Solver," CAD 27, 1995, 487-501.

[2] C.-S. Chiang, C. Hoffmann and Paul Rosen, "Hardware Assist for Constrained Circle Constructions I: Sequential Problems," submitted.

[3] C.-S. Chiang, C. Hoffmann, and P. Rosen, "Hardware Assist for Constrained Circle Constructions II: Cluster Merging Problems," submitted.

[4] V. Cheteut, M. Daniel, S. Hahmann, R. LaGreca, J. Léon, R. Maculet, B. Sauvage. "Constraint modeling for curves and surfaces in CAGD," Intl. J. of Shape Modeling 13, 159-199, 2007.

[5] D. Dietz, B. Piper, E. Sebe. "Rational Cubic Spirals," CAD 40, 2008, 3-12.

[6] C. B. Durand. Symbolic and numerical techniques for constraint solving. PhD thesis, Computer Science Department, Purdue University, 1998.

[7] G. Farin. Curves and Surfaces for Computer-Aided Geometric Design. Academic Press 1990.

[8] I. Fudos and C. Hoffmann. "Constraint based parametric conics for CAD," CAD 28, 1996, 91-100.

[9] X.-S. Gao, Q. Lin, and G.-F. Zhang. A c-tree decomposition algorithm for 2d and 3d geometric constraint solving,. Computer-Aided Design, 38(1):1-13, 2005.

[10] C. Hoffmann. "Constraint Based CAD," JCISE 5:3, 2005, 182-187.

[11] C.M. Hoffmann, R. Joan-Arinyo. "Parametric Modeling." Handbook of Computer Aided Geometric Design, G. Farin, J. Hoschek, M.-S. Kim, eds., Elsevier North Holland, Amsterdam, 2002.

[12] C. Hoffmann, A. Lomonosov, and M. Sitharam. Decomposition plans for geometric constraint systems,

Part I: Performance measures for CAD. Journal of Symbolic Computation, 31(4):367-408, 2001.

[13] C. Hoffmann, A. Lomonosov, and M. Sitharam. Decomposition plans for geometric constraint systems, Part II: New algorithms. Journal of Symbolic Computation, 31(4):409-427, Apr. 2001.

[14] C. Hoffmann and J. Peters. ”Geometric Constraints for CAGD,” in Mathematical Methods for Curves and Surfaces, M. Daehlen, T. Lyche, L. Schumaker, eds., Vanderbilt University Press, 1995, 237-254.

[15] K. Hoff III, T. Culver, J. Keyser, M. Lin, D. Manocha. ”Fast Computation of Generalized Voronoi Diagrams Using Graphics Hardware,” Siggraph ’99, Los Angeles, CA, 277-286.

[16] J. Hoschek and D. Lasser. Fundamentals of Computer Aided Geometric Design. A.K.Peters, Wellesley, MA 1993.

[17] C. Jermann, G. Trombettoni, B. Neveu, and P. Mathis. Decomposition of geometric constraints systems: A survey. International Journal of Computational Geometry and Applications, 23(7):1-35, 2006.

[18] H. Lamure and D. Michelucci. ”Solving geometric constraints by homotopy.” Proc. Symp. on Solid Modeling and Applications, ACM Press 1995: 263-269.

[19] I.-K. Lee, M.-S. Kim, G. Elber. ”Polynomial/Rational Approximation of Minkowski Sum Boundary Curves,” Graphical Models and Image Processing 60, 1998, 136-165.

[20] M. Peternell, B. Odehnal. ”Convolution surfaces of quadratic triangular Bézier surfaces,” CAGD 25, 2008, 116-129.

[21] M. Sampoli. ”Computing the convolution and the Minkowski sum of surfaces,” Proc. . 21st Spring Conference on Computer Graphics, Budmerice, Slovakia, 2005, 111 - 117.

[22] M. Sampoli, M. Peternell, B. Juttler. ”Exact parameterization of convolution surfaces and rational surfaces with linear normals.” CAGD 23, 2006, 179-192.