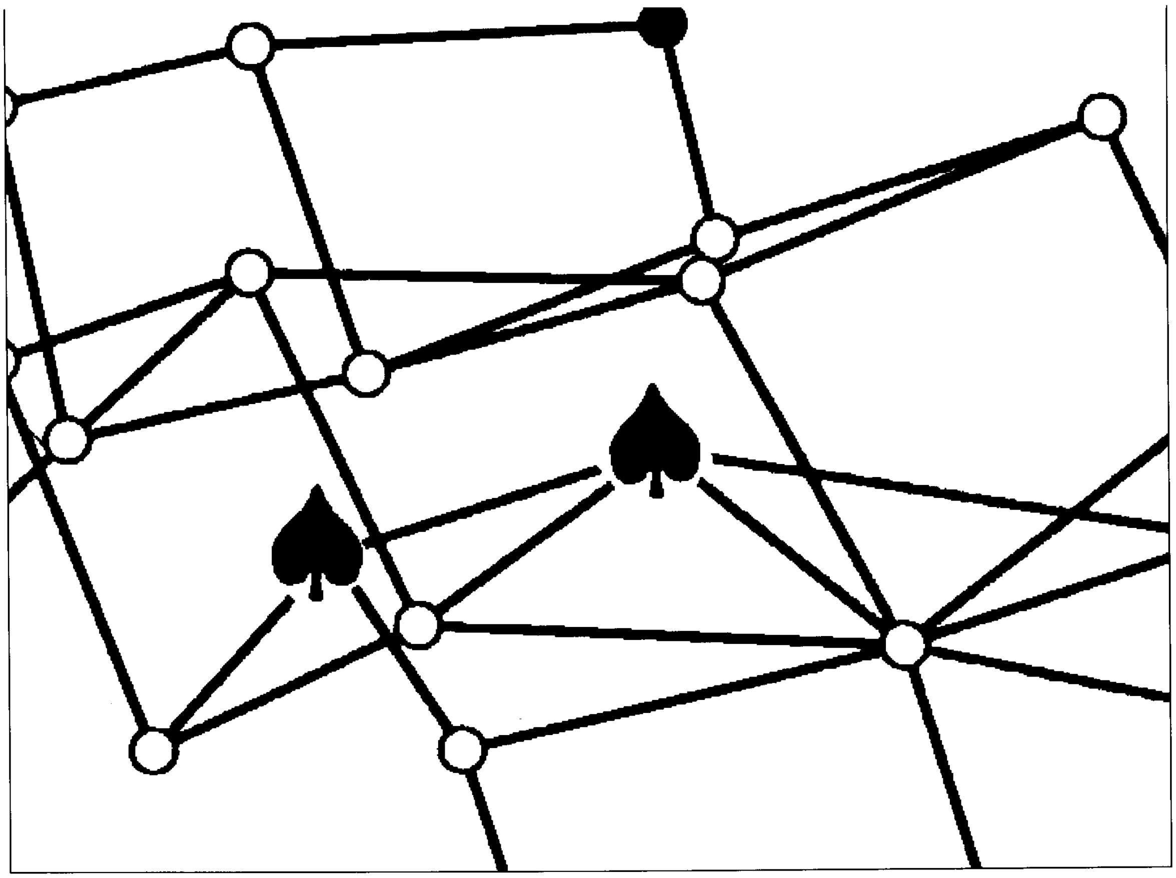


# From Conics to NURBS: A Tutorial and Survey

**Gerald Farin**
**Arizona State University**

*Conics exhibit many important geometric properties of NURBS. This article surveys conic geometry and shows how it carries over to NURBS.*

Ask anyone in the CAD/CAM industry or graphics about the most promising curve or surface form. The answer is invariably "NURBS"—nonuniform rational B-splines. In this article, I describe the main geometric features of this curve and surface representation. Surprisingly, most of these features are already exhibited by conics, which are a special case of NURBS. I discuss properties typical of NURBS—that is, I don't dwell on properties already present in polynomial curves. I adopt the definition that a NURB curve or surface is a piecewise rational polynomial curve or surface. Also, I assume familiarity with the concepts of integral (that is, nonrational) Bezier and B-spline curves and surfaces.[1-3] For more advanced material on NURBS, see *NURBS for Curve and Surface Design.*[4]

## Conics

Conic sections, the oldest known curve form, are still essential to many CAD systems. In 1944, R. Liming[5] used conics as the basis for the first


0272-17-16/92/0900-0078$03.00 © 1992 IEEE

CAD system: He based the design of airplane fuselages on *calculating* with conics, as opposed to the traditional *drafting* with conics. (Because of the tools available to Liming, we might now call his method "calculator-aided design.") There are a number of equivalent ways to define a conic section; for our purposes the following one is very useful:

> A conic section in two space is the perspective projection of a parabola in Euclidean three space into a plane.

To formulate conics as rational curves, we typically choose the projection's center to be the origin $\mathbf{0}$ of a 3D Cartesian coordinate system. The plane into which we project the parabola is the plane $z = 1$. Since we study planar curves in this section, we can think of this plane as a copy of two space, thus identifying points $[x\ y]^T$ with $[x\ y\ 1]^T$. Our special projection is characterized by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow \begin{bmatrix} x/z \\ y/z \\ 1 \end{bmatrix}$$

A point $[x\ y]^T$ is the projection of a whole family of points: Every point on the straight line $[wx\ wy\ w]^T$ projects to $[x\ y]^T$. In the following, I use the shorthand notation $[w\mathbf{x}\ w]^T$ with $\mathbf{x} \in \mathbb{E}^2$ for $[wx\ wy\ w]^T$. (Sometimes the set of all points $[wx\ wy\ w]^T$ is called the *homogeneous form* or *homogeneous coordinates* of $[x\ y]^T$.)

Let $\mathbf{c}(t) \in \mathbb{E}^2$ be a point on a conic. Then there are numbers $w_0, w_1, w_2 \in \mathbb{R}$ and points $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2 \in \mathbb{E}^2$ such that

$$\mathbf{c}(t) = \frac{w_0\mathbf{b}_0B_0^2(t) + w_1\mathbf{b}_1B_1^2(t) + w_2\mathbf{b}_2B_2^2(t)}{w_0B_0^2(t) + w_1B_1^2(t) + w_2B_2^2(t)} \qquad \textbf{(1)}$$

that is, we can express $\mathbf{c}$ as a *parametric rational quadratic curve*. Here the terms $B_i^2(t)$ refer to quadratic Bernstein polynomials.[2]

We call the points $\mathbf{b}_i$ the *control polygon* of the conic $\mathbf{c}$; the numbers $w_i$ are *weights* of the corresponding control polygon vertices. Thus, the conic control polygon is the projection of the control polygon with vertices $[w_i\mathbf{b}_i\ w_i]^T$, which is the control polygon of the 3D parabola that we projected onto the conic $\mathbf{c}$.

The form in Equation 1 is called the *rational quadratic form* of a conic section. If all weights are equal, we recover nonrational quadratics—that is, parabolas. As $w_1$ becomes larger—in other words, as $[w_1\mathbf{b}_1, w_1]$ moves "up" parallel to the $z$ axis—the conic is "pulled" toward $\mathbf{b}_1$.

Conics have many interesting geometric properties, such as eccentricity and axes.[6]

## Weight points

For any conic we can compute the *shoulder tangent:* the tangent at $t = 1/2$. It will intersect the two polygon legs at points $\mathbf{q}_0$ and $\mathbf{q}_1$, respectively. But we don't have to compute the actual intersections to determine $\mathbf{q}_0$ and $\mathbf{q}_1$; we can express them as

$$\mathbf{q}_0 = \frac{w_0\mathbf{b}_0 + w_1\mathbf{b}_1}{w_0 + w_1}, \quad \mathbf{q}_1 = \frac{w_1\mathbf{b}_1 + w_2\mathbf{b}_2}{w_1 + w_2}$$

The points $\mathbf{q}_0$ and $\mathbf{q}_1$ are called *weight points.*[7] Instead of prescribing the weights of a conic, we might as well prescribe the position of the weight points, or, equivalently, the shoulder tangent. Figure 1 shows the weight points of a hyperbolic segment.

If the control points and the weight points (that is, the shoulder tangent) are given, the point $\mathbf{p}$ on the shoulder tangent is

$$\mathbf{p} = \frac{\mathbf{b}_0 + 2w_1\mathbf{b}_1 + w_2\mathbf{b}_2}{1 + 2w_1 + w_2} \qquad \textbf{(2)}$$

where $w_1 = \text{ratio}(\mathbf{b}_0, \mathbf{q}_0, \mathbf{b}_1)$ and $w_2 = w_1/\text{ratio}(\mathbf{b}_1, \mathbf{q}_1, \mathbf{b}_2)$.

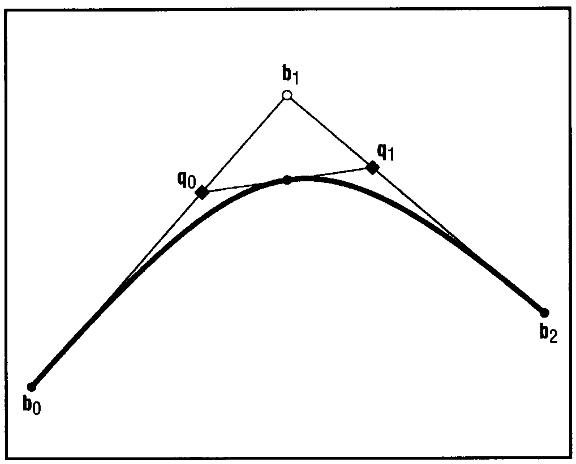



**Figure 1. The weight points of a hyperbolic segment.**

## Reparameterization

A common nonzero factor in the $w_i$ does not affect the conic at all. If $w_0 \neq 0$, we can therefore always achieve $w_0 = 1$ by a simple scaling of all $w_i$. But other weight changes might also leave the curve shape unchanged: These correspond to *rational linear parameter transformations* $t = t(\hat{t})$. Let's set

$$t = \frac{\hat{t}}{\hat{p}(1-\hat{t}) + \hat{t}}, \quad (1-t) = \frac{\hat{p}(1-\hat{t})}{\hat{p}(1-\hat{t}) + \hat{t}}$$

for some $\hat{\rho} \in \mathbb{R}$. We can insert this into Equation 1 and get

$$c(\hat{t}) = \frac{\hat{\rho}^2 w_0 b_0 B_0^2(\hat{t}) + \hat{\rho} w_1 b_1 B_1^2(\hat{t}) + w_2 b_2 B_2^2(\hat{t})}{\hat{\rho}^2 w_0 B_0^2(\hat{t}) + \hat{\rho} w_1 B_1^2(\hat{t}) + w_2 B_2^2(\hat{t})} \quad \textbf{(3)}$$

Thus, the curve shape does not change if we replace each weight $w_i$ by $\hat{w}_i = \hat{\rho}^{2-i} w_i$. If, for a given set of weights $w_i$, we select

$$\hat{\rho} = \sqrt{\frac{w_2}{w_0}}$$

then we get $\hat{w}_0 = w_2$. After dividing all three weights through by $w_2$, we even have $\hat{w}_0 = \hat{w}_2 = 1$. A conic that satisfies this condition is in *standard form*. We can rewrite all conics with $w_0, w_2 \neq 0$ in standard form with the above choice of $\hat{\rho}$. If the conic is in standard form, the shoulder tangent is parallel to $\mathbf{b}_0\mathbf{b}_2$.

Let $\hat{\mathbf{q}}_i$ be the weight points after reparameterization. On each polygon leg, we now have four points: $\mathbf{b}_0$, $\mathbf{q}_0$, $\hat{\mathbf{q}}_0$, $\mathbf{b}_1$ on the first one and $\mathbf{b}_1$, $\mathbf{q}_1$, $\hat{\mathbf{q}}_1$, $\mathbf{b}_2$ on the second one. Their *cross ratios* are equal:[8]

$$\mathrm{cr}(\mathbf{b}_0, \mathbf{q}_0, \hat{\mathbf{q}}_0, \mathbf{b}_1) = \mathrm{cr}(\mathbf{b}_1, \mathbf{q}_1, \hat{\mathbf{q}}_1, \mathbf{b}_2) \quad \textbf{(4)}$$

The cross ratio cr is the fundamental invariant of projective geometry:

$$\mathrm{cr}(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}) = \mathrm{ratio}(\mathbf{a}, \mathbf{b}, \mathbf{d}) \,/\, \mathrm{ratio}(\mathbf{a}, \mathbf{c}, \mathbf{d})$$

Since all the lines $\mathbf{b}_0\mathbf{b}_1$, $\mathbf{b}_1\mathbf{b}_2$, $\mathbf{q}_0\mathbf{q}_1$, and $\hat{\mathbf{q}}_0\hat{\mathbf{q}}_1$ are tangent to the conic, we can interpret Equation 4 as a statement about any four tangents to a conic—I shall refer to the equation as the *four-tangent theorem.*

## Derivatives

We write a conic section as a rational function, so derivatives now call for the quotient rule—at least at first sight. Life is actually easier than that, with a little trick.[1] We denote the derivative by $\dot{\mathbf{c}}$ and have

$$\dot{\mathbf{c}}(t) = \frac{1}{w(t)}\left[\dot{\mathbf{p}}(t) - \dot{w}(t)\mathbf{c}(t)\right] \quad \textbf{(5)}$$

We can easily compute higher derivatives by applying this formula recursively.[2]

We now consider two conics, one defined over the interval $[u_0, u_1]$ with control polygon $\mathbf{b}_0$, $\mathbf{b}_1$, $\mathbf{b}_2$ and weights $w_0$, $w_1$, $w_2$, and the other defined over the interval $[u_1, u_2]$ with control polygon $\mathbf{b}_2$, $\mathbf{b}_3$, $\mathbf{b}_4$ and weights $w_2$, $w_3$, $w_4$. Both segments form a $C^1$ curve if

$$\frac{w_1}{u_1 - u_0}\Delta\mathbf{b}_1 = \frac{w_3}{u_2 - u_1}\Delta\mathbf{b}_2 \quad \textbf{(6)}$$

The interval lengths appear because of the chain rule, which we must apply since we now consider a composite curve with a global parameter $u$. But notice the absence of the weight $w_2$ in the $C^1$ equation. This means there are $C^1$ piecewise conics (in the plane) that are not the projections of $C^1$ piecewise parabolas (in space). In fact, we can project two *discontinuous* parabolas in three dimensions onto a smooth piecewise conic in two dimensions.

## Curvature and $G^2$ continuity

The curvature of a conic involves first and second derivatives, and is somewhat messy to write down. However, if we restrict ourselves to the endpoints, things work out much more simply: The curvatures $\kappa_0$ and $\kappa_1$ at $\mathbf{b}_0$ and $\mathbf{b}_2$ are given by

$$\kappa_0 = \frac{w_0 w_2}{w_1^2}\frac{A}{l_0^3}, \quad \kappa_1 = \frac{w_0 w_2}{w_1^2}\frac{A}{l_1^3} \quad \textbf{(7)}$$

with $A$ denoting the area of the triangle formed by the control polygon. Here $l_0$ denotes the length of the first control polygon leg, and $l_1$, that of the second. If we are interested in curvatures at other points on the conic, we just perform subdivision and apply Equation 7. (As an interesting aside, note that the ratio $\kappa_0/\kappa_1$ does not depend on the weights.)

Suppose again that we have two conics, one with control polygon $\mathbf{b}_0$, $\mathbf{b}_1$, $\mathbf{b}_2$, and the other with control polygon $\mathbf{b}_2$, $\mathbf{b}_3$, $\mathbf{b}_4$. Let's assume that $\mathbf{b}_1$, $\mathbf{b}_2$, and $\mathbf{b}_3$ are collinear, making the two conics tangent continuous. They are also curvature continuous if

$$\frac{A_1}{A_2}\frac{l_2^3}{l_1^3} = \frac{w_1^3}{w_3^3}$$

assuming that both conics are in standard form and setting $l_1 = \|\mathbf{b}_2 - \mathbf{b}_1\|$, $l_2 = \|\mathbf{b}_3 - \mathbf{b}_2\|$. Planar curves that are tangent and curvature continuous are called $G^2$ continuous.

Recently, Pottmann[9] used $G^2$ conics to solve the following interpolation problem: Given two points $\mathbf{b}_0$ and $\mathbf{b}_4$, plus a tangent direction and a curvature at each point, find a curve that interpolates to these data. This is too much information for one conic to interpolate to, so Pottmann uses two conics instead. Each interpolates to the information at one of the given points, and both meet with $G^2$ continuity. The same problem is also solvable with one rational cubic.[2]

## Control vectors

In principle, we can write any arc of a conic as a rational quadratic curve segment (possibly with negative weights). But what happens when the tangents at $\mathbf{b}_0$ and $\mathbf{b}_2$ are parallel? Intuitively, this would send $\mathbf{b}_1$ to infinity. With a little analysis, we can overcome this problem, as the following example shows.

Let a conic be given by $\mathbf{b}_0 = [-1, 0]^T$, $\mathbf{b}_2 = [1, 0]^T$, and $\mathbf{b}_1 = [0, \tan\alpha]^T$ and a weight $w_1 = c\cos\alpha$ (we assume standard form). The angle $\alpha$ is formed by $\mathbf{b}_0\mathbf{b}_1$ and $\mathbf{b}_0\mathbf{b}_2$ at $\mathbf{b}_0$. For $c = 1$, we get a circular arc, as illustrated in Figure 2.

The equation of our conic is

$$c(t)=\frac{(1-t)^2\begin{bmatrix}-1\\0\end{bmatrix}+2c\ \cos\ \alpha\cdot t(1-t)\begin{bmatrix}0\\\tan\ \alpha\end{bmatrix}+t^2\begin{bmatrix}1\\0\end{bmatrix}}{(1-t)^2+2ct(1-t)\cos\ \alpha+t^2}$$

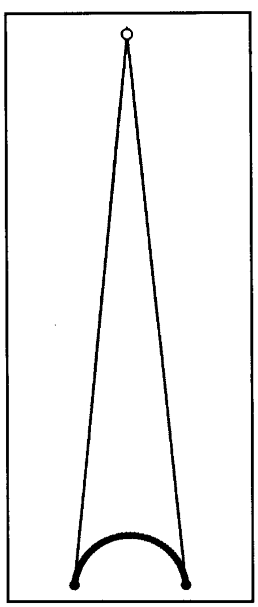
What happens as $\alpha$ tends to $\pi/2$? For the limiting conic, we get the equation

$$\mathbf{c}(t)=\frac{(1-t)^2\begin{bmatrix}-1\\0\end{bmatrix}+2t(1-t)\begin{bmatrix}0\\c\end{bmatrix}+t^2\begin{bmatrix}1\\0\end{bmatrix}}{(1-t)^2+t^2} \qquad \textbf{(8)}$$

Thus, we have resolved the problem of a weight tending to zero and a control point tending to infinity. For $c = 1$, we get a semicircle; other values of $c$ give different conics. For $c = -1$, we get the "lower" half of the unit circle.

We have overcome possible problems with parallel end tangents, but for a price: The factors of $\mathbf{b}_0$ and $\mathbf{b}_2$ sum to 1 identically; hence $[0, c]^T$ must be interpreted as a *vector*. Thus, Equation 8 contains both control points and control vectors. We thus lose an important property of Bezier curves, namely, the convex hull property. It is defined only for point sets, not for a potpourri of points and vectors.

In projective geometry, vectors are sometimes called "points at infinity." As a consequence, Vesprille[10] and later Piegl[11] use the term "infinite control points." The term "control vector" seems more appropriate because it lets us distinguish between $[0, c]^T$ and $[0, -c]^T$.



Control vectors provide a very compact form for writing a semicircle. But, in my opinion, two disadvantages argue against their practical use:

1. The loss of the convex hull property.
2. The need for a special-case treatment to write the control vector form in the context of "normal" rational quadratics.

Other researchers have written more extensively about control vectors.[12]

**Figure 2. A 168-degree arc of a circle; $\alpha$ is close to 90 degrees.**

## Rational Bezier curves

So far, we obtained a conic section in Euclidean two space as the projection of a parabola (a quadratic) in three space. We can express conic sections as rational quadratic Bezier curves, and their generalization to higher degree rational curves is quite straightforward: A rational Bezier curve of degree $n$ in three space is the projection of an $n$th-degree Bezier curve in four space into the hyperplane $w = 1$. We can view this 4D hyperplane as a copy of three space; we assume that a point in four space is given by its coordinates $[x\ y\ z\ w]^T$. Proceeding in exactly the same way as we did for conics, we can show that an $n$th-degree rational Bezier curve is given by

$$\mathbf{x}(t)=\frac{w_0\mathbf{b}_0B_0^n(t)+\cdots+w_n\mathbf{b}_nB_n^n(t)}{w_0B_0^n(t)+\cdots+w_nB_n^n(t)};\quad \mathbf{x}(t),\mathbf{b}_i\in I\!E^3 \qquad \textbf{(9)}$$

The $B_i^n$ are *Bernstein polynomials* of degree $n$, the $w_i$ are again *weights*, and the $\mathbf{b}_i$ form the control polygon. It is the projection of the 4D control polygon $[w_i\mathbf{b}_i\ w_i]^T$ of the nonrational 4D preimage of $\mathbf{x}(t)$.

If all weights equal 1, we obtain the standard nonrational Bezier curve; then the denominator is identically equal to 1. If some $w_i$ are negative, singularities might occur; we therefore deal only with nonnegative $w_i$. Then rational Bezier curves enjoy all the properties that their nonrational counterparts possess; for example, they are affinely invariant. If all $w_i$ are nonnegative, we have the convex hull property.

### Bezier curves' weight points

We use the control points and weights to define weight points $\mathbf{q}_i$:

$$\mathbf{q}_i=\frac{w_i\mathbf{b}_i+w_{i+1}\mathbf{b}_{i+1}}{w_i+w_{i+1}}$$

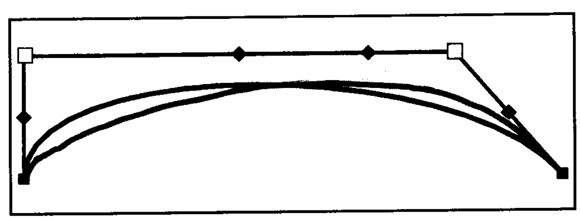


**Figure 3. The change of one weight point (from black to gray) "shifts" the black curve to the gray curve.**

Although weight points[13] are determined by the weights, the reverse statement also holds: If the weight points are given, we can compute a set of weights that generate them. As Figure 3 shows, we can use the weight points as a design tool: Changing one weight point has a predictable effect on the new curve. If we change one weight point as in Figure 3, all weights to its left are multiplied by a common factor, and all weights to its right are multiplied by a different factor. Such operations are too tedious for a designer to manipulate directly.

The weight points also give some insight into the behavior of rational Bezier curves under projective maps (such as viewing transformations in graphics). We subject the polygon *together with its weight points* to a projective map. Before the map, the weight points and weights were related by

$$w_{i+1} = w_i \text{ratio}(\mathbf{b}_i, \mathbf{q}_i, \mathbf{b}_{i+1}); \qquad i = 0, \ldots, n-1$$

A projective map does not leave ratios invariant. Thus mapping changes the above ratios, giving rise to different weights of the mapped curve.

## Reparameterizing Bezier curves

Arguing exactly as in the conic case (see the earlier section on conics), we might *reparameterize* a rational Bezier curve by changing the weights according to

$$\hat{w}_i = c^i w_i; \qquad i = 0, \ldots, n$$

where $c$ is any nonzero constant. Figure 4 shows how the reparameterization affects the parameter spacing on the curve: The curve shape remains the same.

The old and new weight points are again related by the cross ratio condition:

$$\text{cr}(\mathbf{b}_i, \mathbf{q}_i, \hat{\mathbf{q}}_i, b_{i+1}) = \text{const.} \qquad i = 0, \ldots, n-1$$

We can transform a rational Bezier curve to standard form by using the rational linear parameter transformation resulting from the choice

$$c = \sqrt[n]{\frac{w_0}{w_n}}$$

This results in $\hat{w}_n = w_0$. After dividing all weights through by $w_0$, we have the standard form $\hat{w}_0 = \hat{w}_n = 1$. Of course, the root must exist. Patterson[14] gave a different derivation of this result.

How can rational Bezier curves in nonstandard form arise? A common case occurs with rational Bezier surfaces (as discussed in the later section on rational Bezier and B-spline surfaces): The end weights of an isoparametric curve generally will not be unity. Such curves are often "extracted" from a surface and then treated as entities in their own right.

A final note: Even after standardization, a rational curve will not be parameterized with respect to arc length. As we trace out the interval [0, 1] in equal parameter increments, the curve is not traced out in steps of equal length. Farouki and Sakkalis[15] pointed out this "defect" of rational curves: In particular, a circle in rational form will not be traced out as we are used to from the sin/cos parameterization.

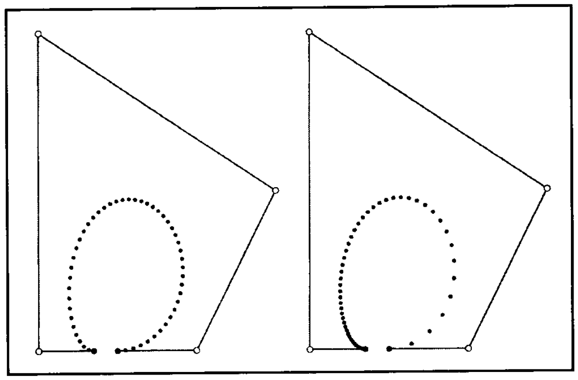


**Figure 4. (a) A rational Bezier curve evaluated at parameter values 0, 0.02, . . . , 1. (b) The same curve and parameter values after a reparameterization with $c = 3$.**

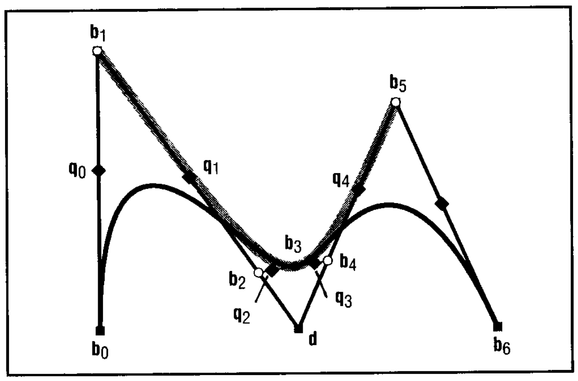



**Figure 5. A rational $G^2$ condition: The left and right rational cubics are $G^2$ if they share the same (gray) conic osculant.**

## Bezier curvature and $G^2$ continuity

We find the curvature at the first control point of a rational cubic Bezier curve following the conic recipe:

$$\kappa_0 = \frac{4}{3} \frac{w_0 w_2}{w_1^2} \frac{A}{l_0^3}$$

Here, $A$ denotes the area of the triangle $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2$, and $l_0 = \|\mathbf{b}_1 - \mathbf{b}_0\|$. The curvature at $\mathbf{b}_0$ thus involves the first three control points and weights only. These define a conic $\mathbf{c}_0$, and we call $\mathbf{c}_0$ the *conic osculant* at $\mathbf{b}_0$. The curvature at $\mathbf{b}_0$ of the cubic and its conic osculant differ by a factor of 4/3.

The plane spanned by $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2$ is the *osculating plane* at $\mathbf{b}_0$. Two rational Bezier curves are $G^2$ if they share a common tangent direction, osculating plane, and curvature at their common point. Thus, they are certainly $G^2$ if they share the same conic osculant at their common point. This is illustrated in Figure 5: We know that the left osculant is defined by $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$ and weight points $\mathbf{q}_1$, $\mathbf{q}_2$. The straight line $\mathbf{q}_1\mathbf{q}_2$ is tangent to this conic. The right osculant is defined by control points $\mathbf{b}_3$, $\mathbf{b}_4$, $\mathbf{b}_5$ and weight points $\mathbf{q}_3$, $\mathbf{q}_4$. Again, the straight line $\mathbf{q}_3\mathbf{q}_4$ is a tangent. Of course, $\mathbf{b}_2\mathbf{b}_4$ is tangent to both osculants.

If both osculants are part of the same conic, then that conic must be defined by the control polygon $\mathbf{b}_1$, $\mathbf{d}$, $\mathbf{b}_5$ and weight points $\mathbf{b}_2$, $\mathbf{b}_4$. This conic also has tangents $\mathbf{b}_1\mathbf{d}$ and $\mathbf{d}\mathbf{b}_5$. We can now use the four-tangent theorem and formulate a $G^2$ condition for two rational cubics:

$$\mathrm{cr}(\mathbf{b}_1, \mathbf{q}_1, \mathbf{b}_2, \mathbf{d}) = \mathrm{cr}(\mathbf{b}_2, \mathbf{q}_2, \mathbf{b}_3, \mathbf{b}_4) \tag{10}$$

and

$$\mathrm{cr}(\mathbf{b}_2, \mathbf{b}_3, \mathbf{q}_3, \mathbf{b}_4) = \mathrm{cr}(\mathbf{d}, \mathbf{b}_4, \mathbf{q}_4, \mathbf{b}_5) \tag{11}$$

Pottmann[16] suggested a similar (less symmetrical but more general) $G^2$ condition.

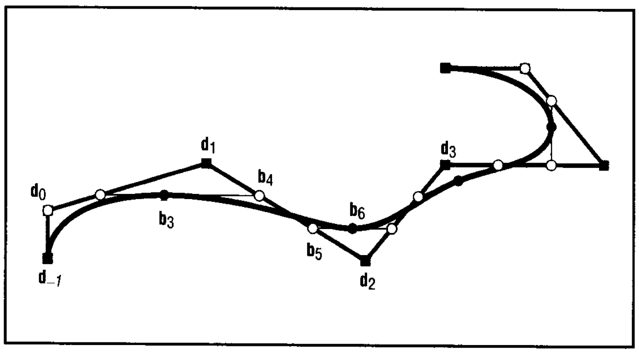



**Figure 6. Cubic NURBS: the NURB and Bezier control points. NURB control points are squares. Bezier control points are circles.**

## Cubic NURB curves

A $C^2$ cubic NURB curve in three dimensions is the projection through the origin of a 4D nonrational $C^2$ cubic B-spline curve into the hyperplane $w = 1$. The NURB curve's control polygon is given by vertices $\mathbf{d}_{-1}, \ldots, \mathbf{d}_{L+1}$. Each vertex $\mathbf{d}_i \in \mathbb{E}^3$ has a corresponding weight $w_i$. The knot sequence is $u_0, \ldots, u_L$, and we set $\Delta_i = u_{i+1} - u_i$. The NURB curve has a piecewise rational cubic Bezier representation. We can obtain it by projecting the corresponding 4D Bezier points into the hyperplane $w = 1$.

Thus, referring to Figure 6, we get

$$\mathbf{b}_{3i-2} = \frac{w_{i-1}(1-\alpha_i)\mathbf{d}_{i-1} + w_i\alpha_i\mathbf{d}_i}{v_{3i-2}} \tag{12}$$

$$\mathbf{b}_{3i-1} = \frac{w_{i-1}\beta_i\mathbf{d}_{i-1} + w_i(1-\beta_i)\mathbf{d}_i}{v_{3i-1}} \tag{13}$$

where all points $\mathbf{b}_j$, $\mathbf{d}_k$ are in $\mathbb{E}^3$ and

$$\Delta = \Delta_{i-2} + \Delta_{i-1} + \Delta_i,$$

$$\alpha_i = \frac{\Delta_{i-2}}{\Delta},$$

$$\beta_i = \frac{\Delta_i}{\Delta}$$

The weights of these Bezier points are

$$v_{3i-2} = w_{i-1}(1 - \alpha_i) + w_i\alpha_i \tag{14}$$

$$v_{3i-1} = w_{i-1}\beta_i + w_i(1 - \beta_i) \tag{15}$$

For the junction points, we get

$$\mathbf{b}_{3i} = \frac{\gamma_i v_{3i-1}\mathbf{b}_{3i-1} + (1-\gamma_i)v_{3i+1}\mathbf{b}_{3i+1}}{v_{3i}} \tag{16}$$

where

$$\gamma_i = \frac{\Delta_i}{\Delta_{i-1} + \Delta_i}$$

and

$$v_{3i} = \gamma_i v_{3i-1} + (1 - \gamma_i)v_{3i+1}$$

is the weight of the junction point $\mathbf{b}_{3i}$.

Designing with cubic NURB curves is not very different from designing with their nonrational counterparts. We now have the added freedom of being able to change weights. A change of only one weight affects a rational B-spline curve only locally, as is amply demonstrated in the literature.[2,17,18] We can also reparameterize rational B-splines, as Lee and Lucian show.[19]

We should not overlook a problem with this curve form: The added "freedom" of weights is potentially more a nuisance than a real help. A designer might simply be overburdened by having to specify control points, parameter values, and weights. We need more methods to automate this process. I describe one in the next section.

## Geometric rational splines

$G^2$ polynomial splines are well established in the literature.[20,21] Of course, they can be generalized to the rational case.[22,23] More theoretical discussions of NURB $G^2$ continuity are also available.[24-27]

I now briefly describe a class of $G^2$ rational cubics that do not rely on the specification of weights; instead, I use more geometric input. Figure 6 shows the NURB control points (squares). We can use Equations 12 to 16 to compute the Bezier points of each rational cubic piece from the known weights and parameters. Here we try a different approach: Using just the NURB control polygon and the Bezier points

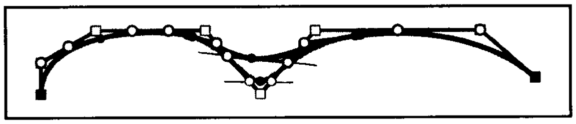


**Figure 7. An example of the scheme for $G^2$ NURBS. One tangent has been changed.**

that lie on the polygon legs (these are the points of the form $\mathbf{b}_{3i\pm1}$), we can define a piecewise cubic $G^2$ NURB curve.

As a first step, recall that each $\mathbf{b}_{3i}$ must lie on a conic with control polygon $\mathbf{b}_{3i-2}$, $\mathbf{d}_i$, $\mathbf{b}_{3i+2}$ and weight points $\mathbf{b}_{3i-1}$, $\mathbf{b}_{3i+1}$. Application of Equation 2 thus yields the junction points $\mathbf{b}_{3i}$. We will use the $G^2$ conditions Equations 10 and 11, so we need to look at the weight points of each cubic piece. We prescribe all weight points $\mathbf{q}_{3i+1}$—that is, the ones on the control legs—to be the midpoints of $\mathbf{b}_{3i+1}$ and $\mathbf{b}_{3i+2}$. Now all remaining weight points follow from application of the $G^2$ conditions formulated in Equations 10 and 11. Having found all weight points and already having all control points of each rational cubic segment, we have determined our $G^2$ NURB curve. I have described a nonrational version of this method elsewhere.[28]

The advantage of this kind of curve description is that it does not need abstract constraints such as parameter values and weights. Instead, it uses, in addition to the control polygon, *tangents* to the curve: After all, the lines $\mathbf{b}_{3i-1}$, $\mathbf{b}_{3i+1}$ are tangents at $\mathbf{b}_{3i}$. Figure 7 gives an example.

While we were happy to produce a $G^2$ curve without referring to a knot sequence, it is in fact possible to produce a knot sequence with respect to which the described curves will be twice differentiable, or $C^2$.[29]

As a next step, we might try to automate tangent selection. I have experimented with this using a method that aims at reducing the eccentricity of the osculants.[30]

## Rational Bezier, B-spline surfaces

We can generalize Bezier and B-spline surfaces to their rational counterparts in much the same way as we did for curves. In other words, we define a rational Bezier or B-spline surface as the projection of a 4D tensor product Bezier or B-spline surface. Thus, the rational Bezier patch takes the form

$$\mathbf{x}(u,v)=\frac{\sum_i\sum_j w_{i,j}\mathbf{b}_{i,j}B_i^m(u)B_j^n(v)}{\sum_i\sum_j w_{i,j}B_i^m(u)B_j^n(v)} \tag{17}$$

and a rational B-spline surface is written as

$$\mathbf{s}(u,v)=\frac{\sum_i\sum_j w_{i,j}\mathbf{d}_{i,j}N_i^m(u)N_j^n(v)}{\sum_i\sum_j w_{i,j}N_i^m(u)N_j^n(v)} \tag{18}$$

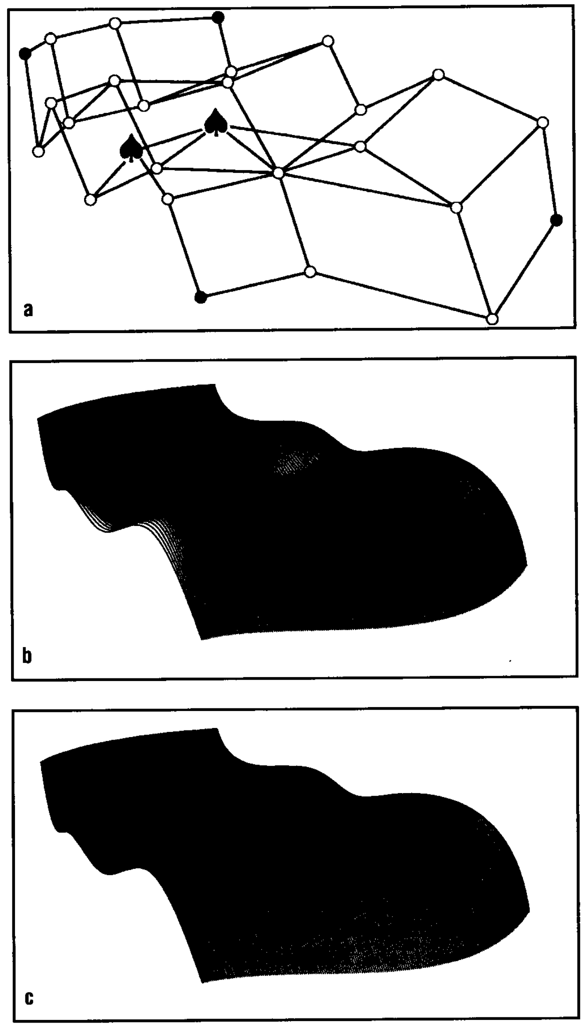



**Figure 8. Two rational bicubic B-spline surfaces: (a) the control net for both surfaces; (b) all weights equal to 1 except for the indicated ones, which have weight 5; (c) the weights indicated by solid circles are 0.5.**

Figure 8 shows two rational B-spline surfaces. Their control nets are the same; the weight changes are only local.

We obtain rational surfaces as the projections of tensor product patches, but they are not tensor product patches themselves. Recall that a tensor product surface has the form $\mathbf{x}(u, v) = \sum_i\sum_j\mathbf{c}_{i,j}F_{i,j}(u, v)$, where we can express the basis functions $F_{i,j}$ as products $F_{i,j}(u, v) = A_i(u)B_j(v)$. The basis functions for Equation 18 have the form

$$F_{i,j}(u,v)=\frac{w_{i,j}N_i^m(u)N_j^n(v)}{\sum_i\sum_j w_{i,j}N_i^m(u)N_j^n(v)}$$

### Rational de Casteljau algorithm

**Given:** A triangular array of points $\mathbf{b_i} \in \mathbb{E}^3$, $|\mathbf{i}| = n$, corresponding weights $w_\mathbf{i}$, and a point in a domain triangle with barycentric coordinates $\mathbf{u}$.

**Set (with eu** = (1, 0, 0), **ev** = (0, 1, 0), **ew** = (0, 0, 1)):

$$\mathbf{b}_\mathbf{i}^r(\mathbf{u}) = \frac{u w_{\mathbf{i}+\mathbf{eu}}^{r-1}(\mathbf{u})\mathbf{b}_{\mathbf{i}+\mathbf{eu}}^{r-1}(\mathbf{u}) + v w_{\mathbf{i}+\mathbf{ev}}^{r-1}(\mathbf{u})\mathbf{b}_{\mathbf{i}+\mathbf{ev}}^{r-1}(\mathbf{u}) + w w_{\mathbf{i}+\mathbf{ew}}^{r-1}(\mathbf{u})\mathbf{b}_{\mathbf{i}+\mathbf{ew}}^{r-1}(\mathbf{u})}{w_\mathbf{i}^r(\mathbf{u})}$$

where

$$w_\mathbf{i}^r(\mathbf{u}) = u w_{\mathbf{i}+\mathbf{eu}}^{r-1}(\mathbf{u}) + v w_{\mathbf{i}+\mathbf{ev}}^{r-1}(\mathbf{u}) + w w_{\mathbf{i}+\mathbf{ew}}^{r-1}(\mathbf{u})$$

and

$$r = 1, \ldots, n \quad \text{and} \quad |\mathbf{i}| = n - r$$

and $\mathbf{b}_\mathbf{i}^0(\mathbf{u}) = \mathbf{b_i}$, $w_\mathbf{i}^0 = w_\mathbf{i}$. Then $\mathbf{b}_\mathbf{0}^n(\mathbf{u})$ is the point with parameter value $\mathbf{u}$ on the rational Bezier triangle $\mathbf{b}^n$.

Because of the denominator's structure, this generally cannot be factored into the required form $F_{i,j}(u, v) = A_i(u)B_j(v)$.

But even though rational surfaces do not possess a tensor product structure, we can use many tensor product algorithms for their manipulation. Consider, for example, the problem of finding the piecewise rational bicubic Bezier form of a rational bicubic B-spline surface. All we do is convert each row of the B-spline control net into piecewise rational Bezier cubics (see the earlier section on cubic NURB curves). Then we repeat this process for each column of the resulting net (and the resulting weights), simply following the standard recipe for tensor product surfaces.[2,31]

As another example, consider the problem of extracting an isoparametric curve from a rational Bezier surface. Suppose the curve corresponds to $v = \hat{v}$. We simply interpret all control net columns as control polygons and evaluate each at $\hat{v}$, using, for example, a rational version of the de Casteljau algorithm (see sidebar). We also have to compute a weight in each case. We can now interpret all obtained points together with their weights as the Bezier control polygon of the desired isoparametric curve. In general, its end weights will not be unity: The curve will not be in standard form (as described in the section on rational Bezier curves). We can remedy this situation by using the reparameterization algorithm also described in that section.

## Rational Bezier triangles

Following the familiar theme of generating rational curve and surface schemes, we define a rational Bezier triangle to be the projection of a nonrational 4D Bezier triangle.[2] We thus have

$$\mathbf{b}^n(\mathbf{u}) = \mathbf{b}_0^n(\mathbf{u}) = \frac{\sum_{|\mathbf{i}|=n} w_\mathbf{i}\mathbf{b}_\mathbf{i}B_\mathbf{i}^n(\mathbf{u})}{\sum_{|\mathbf{i}|=n} w_\mathbf{i}B_\mathbf{i}^n(\mathbf{u})} \tag{19}$$

where, as usual, the $w_\mathbf{i}$ are the weights associated with the control vertices $\mathbf{b_i}$, and $\mathbf{i} = (i, j, k)$. For positive weights we have the convex hull property, and we have affine and projective invariance.

We can evaluate rational Bezier triangles using a de Cas-teljau algorithm (see sidebar). This algorithm works because we can interpret each intermediate $\mathbf{b}_\mathbf{i}^r$ as the projection of the corresponding point in the de Casteljau algorithm of the nonrational 4D preimage of our patch. The algorithm also produces the *normal vector* to the surface: The points $\mathbf{b}_\mathbf{eu}^{n-1}$, $\mathbf{b}_\mathbf{ev}^{n-1}$, and $\mathbf{b}_\mathbf{ew}^{n-1}$ span the tangent plane at $\mathbf{b}_\mathbf{0}^n$.

### Derivatives of Bezier triangles

I now give a formula for the *directional derivative* of a rational Bezier triangular patch. Let $\mathbf{d}$ denote a direction in the domain triangle, expressed in barycentric coordinates. We are interested in the directional derivative $D_\mathbf{d}$ of a rational triangular Bezier patch $\mathbf{b}_n(\mathbf{u})$. Proceeding exactly as in the curve case (see the section on conics), we get

$$D_\mathbf{d}\mathbf{b}^n(\mathbf{u}) = \frac{1}{w(\mathbf{u})}\left[\dot{\mathbf{p}}(\mathbf{u}) - D_\mathbf{d}(\mathbf{u})\mathbf{b}^n(\mathbf{u})\right]$$

where we have set

$$\mathbf{p}(\mathbf{u}) = w(\mathbf{u})\mathbf{b}^n(\mathbf{u}) = \sum_{|\mathbf{i}|=n} w_\mathbf{i}\mathbf{b}_\mathbf{i}B_\mathbf{i}^n(\mathbf{u})$$

Higher derivatives follow the pattern outlined for conics derivatives:

$$D_\mathbf{d}^r\mathbf{b}^n(\mathbf{u}) = \frac{1}{w(\mathbf{u})}\left[D_\mathbf{d}^r\mathbf{p}(\mathbf{u}) - \sum_{j=1}^{r} D_\mathbf{d}^j w(\mathbf{u})D_\mathbf{d}^{r-j}(\mathbf{u})\right]$$

### The sphere

We can use rational Bezier triangles to represent an *octant of a sphere*. As it turns out, this representation has to be rational quartic and not, as we might guess, rational quadratic.[32] To represent the whole sphere, we assemble eight copies of this octant patch. Other representations are possible: We can write each octant as a rational biquadratic patch (introducing singularities at the north and south poles).[17] A representation of the whole sphere as two rational bicubics[33] turned out to be incorrect.[34] Cobb presented a different way of representing the sphere: He covered it with six rational bicubics having a cubelike connectivity.[35]

### Quadrics

There are (at least) two motivations for the use of rational Bezier curves: They are projectively invariant, and they allow us to represent conics—namely, in the form of rational quadratics. While the first argument holds trivially for rational Bezier triangles, the second one does not carry over immediately.

The proper generalization of a conic to surfaces is a *quadric surface*—"quadric" for short. A conic (curve) has the implicit equation $q(x, y) = 0$, where $q$ is a quadratic polynomial in $x$ and $y$. Similarly, a quadric has the implicit equation $q(x, y, z) = 0$, where $q$ is quadratic in $x$, $y$, and $z$.

Given a rational quadratic Bezier triangle, what is the condition that it represents a quadric patch? First, recall that every rational quadratic Bezier triangle is completely defined by its boundary curves, as there are no interior Bezier points. Each boundary curve is a segment of a conic. These conics are defined beyond the confines of their control polygons; for example, they might be ellipses. Our condition is now that these three conics *intersect at one point*.[36]

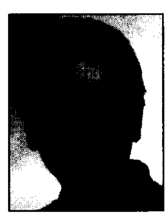
So while it is true that we can use NURBS to represent quadrics,[37] it is far from trivial (and really unresolved at this point) to decide whether a given patch is quadric. This applies to rectangular patches as well as to triangular ones. ❑

## Acknowledgments

This research was supported by NSF grant DMC-8807747 and by Department of Energy grant DE-FG02-87ER25041 to Arizona State University.

## References

1. I. Faux and M. Pratt, *Computational Geometry for Design and Manufacture*, Ellis Horwood, Chichester, U.K., 1979.
2. G. Farin, *Curves and Surfaces for Computer-Aided Geometric Design*, Academic Press, New York, 1992.
3. J. Hoschek and D. Lasser, *Grundlagen der Geometrischen Datenverarbeitung*, B.G. Teubner, Stuttgart, Germany, 1989. (Trans.: *Fundamentals of Computer Aided Geometric Design*), Jones and Bartlett, Boston, 1992.
4. *NURBS for Curve and Surface Design*, G. Farin, ed., SIAM, Philadelphia, 1991.
5. R. Liming, *Practical Analytical Geometry with Applications to Aircraft*, Macmillan, New York, 1944.
6. E. Lee, "The Rational Bezier Representation for Conics," in *Geometric Modeling: Algorithms and New Trends*, G. Farin, ed., SIAM, Philadelphia, 1987, pp. 3-19.
7. G. Geise and B. Juettler, "The Constructive Geometry of Bezier Curves," to appear in *Computer-Aided Geometric Design*.
8. G. Farin and A. Worsey, "Reparametrization and Degree Elevation of Rational Bezier Curves," in *NURBS for Curve and Surface Design*, G. Farin, ed., SIAM, Philadelphia, 1991, pp. 47-58.
9. H. Pottmann, "Locally Controllable Conic Splines with Curvature Continuity," *ACM Trans. Graphics*, Vol. 10, No. 4, Oct. 1991, pp. 366-377.
10. K. Vesprille, *Computer-Aided Design Applications of the Rational B-Spline Approximation Form*, doctoral dissertation, Syracuse Univ., Syracuse, N.Y., 1975.
11. L. Piegl, "On the Use of Infinite Control Points in CAGD," *Computer-Aided Geometric Design*, Vol. 4, Nos. 1-2, Jan. 1987, pp. 155-166.
12. J. Fiorot and P. Jeannin, *Courbes et Surfaces Rationelles*, Masson, Paris, 1989.
13. G. Farin, "Algorithms for Rational Bezier Curves, *Computer-Aided Design*, Vol. 15, No. 2, Feb. 1983, pp. 73-77.
14. R. Patterson, "Projective Transformations of the Parameter of a Rational Bernstein-Bezier Curve," *ACM Trans. Graphics*, Vol. 4, No. 3, July 1986, pp. 276-290.
15. R. Farouki and T. Sakkalis, "Real Rational Curves Are Not `Unit Speed,'" *Computer-Aided Geometric Design*, Vol. 8, No. 2, Mar. 1991, pp. 151-158.
16. H. Pottmann, "A Projectively Invariant Characterization of $G^2$ Continuity for Rational Curves," in *NURBS for Curve and Surface Design*, G. Farin, ed., SIAM, Philadelphia, 1991, pp. 141-148.
17. L. Piegl and W. Tiller, "Curve and Surface Constructions Using Rational B-Splines," *Computer-Aided Design*, Vol. 19, No. 9, 1987, pp. 485-498.
18. L. Piegl, "Modifying the Shape of Rational B-Splines, Part 1: Curves," *Computer-Aided Design*, Vol. 21, No. 8, Sept. 1989, pp. 509-518.
19. E. Lee and M. Lucian, "Moebius Reparametrizations of Rational B-Splines, *Computer-Aided Geometric Design*, Vol. 8, No. 3, Aug. 1991, pp. 213-216.
20. J. Gregory, "Geometric Continuity," in *Mathematical Methods in Computer Aided Geometric Design*, T. Lyche and L. Schumaker, eds., Academic Press, New York, 1989, pp. 353-372.
21. W. Boehm, "Visual Continuity," *Computer-Aided Design*, Vol. 20, No. 6, July 1988, pp. 307-311.
22. B. Barsky, "Introducing the Rational Beta-Spline," in *Proc. Third Int'l Conf. Engineering Graphics and Descriptive Geometry*, Vienna, 1988.
23. W. Boehm, "Rational Geometric Splines," *Computer-Aided Geometric Design*, Vol. 4, Nos. 1-2, Jan. 1987, pp. 67-77.
24. W. Boehm, "Smooth Rational Curves," *Computer-Aided Design*, Vol. 22, No. 1, Jan. 1990, p. 70, (letter to ed.).
25. M. Hohmeyer and B. Barsky, "Rational Continuity: Parametric and Geometric Continuity of Rational Polynomial Curves, *ACM Trans. Graphics*, Vol. 8, No. 4, Oct. 1989, pp. 335-359.
26. H. Pottmann, "Projectively Invariant Classes of Geometric Continuity for CAGD," *Computer-Aided Geometric Design*, Vol. 6, No. 4, Oct. 1989, pp. 307-322.
27. T. Goodman, "Constructing Piecewise Rational Curves with Frenet Frame Continuity," *Computer-Aided Geometric Design*, Vol. 7, Nos. 1-4, June 1990, pp. 15-32.
28. G. Farin, "Visually $C^2$ Cubic Splines, *Computer-Aided Design*, Vol. 14, No. 3, June 1982, pp. 137-139.
29. W. Degen, "Some Remarks on Bezier Curves," *Computer-Aided Geometric Design*, Vol. 5, No. 3, June 1988, pp. 259-268.
30. G. Farin, "Weightless NURBS," to appear in *Computer-Aided Geometric Design.*
31. C. de Boor, *A Practical Guide to Splines*, Springer-Verlag, Berlin, 1978.
32. G. Farin, B. Piper, and A. Worsey, "The Octant of a Sphere as a Non-Degenerate Triangular Bezier Patch," *Computer-Aided Geometric Design*, Vol. 4, No. 4, Oct. 1988, pp. 329-332.
33. L. Piegl, "The Sphere as a Rational Bezier Surface," *Computer-Aided Geometric Design*, Vol. 3, No. 1, Jan. 1986, pp. 45-52.
34. J. Cobb, "Concerning Piegl's Sphere Approximation," *Computer-Aided Geometric Design*, Vol. 6, No. 1, Jan. 1989, p. 85 (letter to ed.).
35. J. Cobb, "A Rational Bicubic Representation of the Sphere," tech. report, Computer Science Dept., Univ. of Utah, Salt Lake City, 1988.
36. W. Boehm and D. Hansford, "Bezier Patches on Quadrics," in *NURBS for Curve and Surface Design*, G. Farin, ed., SIAM, Philadelphia, 1991, pp. 1-14.
37. L. Piegl, "On NURBS: A Survey," *IEEE CG&A*, Vol. 11, No. 1, Jan. 1991, pp. 55-71.



**Gerald Farin** is a professor of computer science at Arizona State University, where his research interests are NURBS and curve and surface smoothing. Before joining ASU, he was an assistant professor at the University of Utah, and he has worked in the CAD/CAM development group at Mercedes-Benz, Germany, where he implemented B-spline curves and surfaces. His book *Curves and Surfaces for Computer Aided Geometric Design* is now in its third edition. He received his PhD from the Technical University of Braunschweig, Germany in 1979.

Farin can be reached at Arizona State University, Department of Computer Science, Tempe, AZ 85287.