Chapter 2

Bézier Curves

Bézier curves are named after their inventor, Dr. Pierre Bézier, an engineer with the Renault car company who set out in the early 1960's to develop a curve formulation for use in shape design that would be intuitive enough for designers and artists to use, without requiring a background in mathematics.

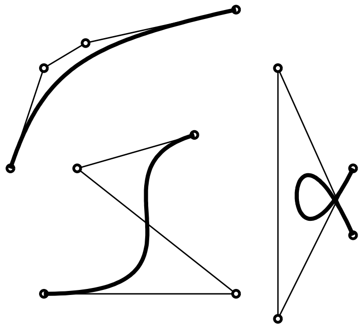


Figure 2.1: Examples of cubic Bézier curves.

Figure 2.1 shows three different Bézier curves, with their corresponding *control polygons.* Each control polygon is comprised of four *control points* that are connect with line segments. (These control polygons are not closed, and might more properly be called polylines.) The beauty of the Bézier representation is that a Bézier curve mimics the shape of its control polygon. A Bézier curve passes through its first and last control points, and is tangent to the control polygon at those endpoints. An artist can quickly master the process of designing shapes using Bézier curves by moving the control points, and most 2D drawing systems like Adobe Illustrator use Bézier curves.

Complicated shapes can be created by using a sequence of Bézier curves. Since Bézier curves are tangent to their control polygons, it is easy to join together two Bézier curves such that they are tangent continuous. Figure 2.2 shows the outline of a letter "g" created using several Bézier curves. All PostScript font outlines are defined using Bézier curves. Hence, as you read these notes, you are gazing upon Bézier curves!

Although Bézier curves can be used productively by artists who have little mathematical training,

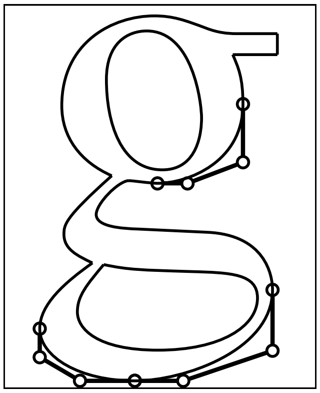
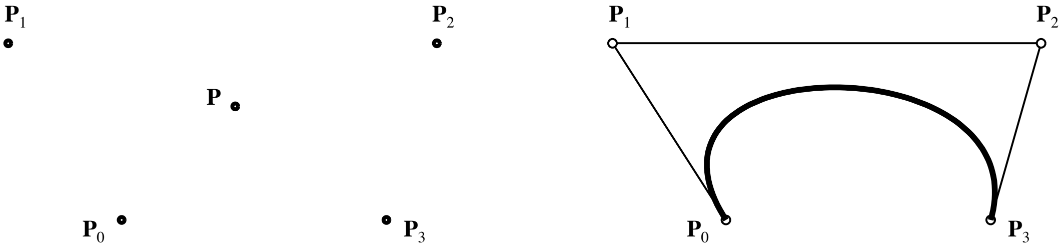


Figure 2.2: Font definition using Bézier curves.

one of the main objectives in this course is to study the underlying mathematics. These notes attempt to show that the power and elegance of Bézier curves are matched by the beauty of the underlying mathematics.

2.1 The Equation of a Bézier Curve

The equation of a Bézier curve is similar to the equation for the center of mass of a set of point masses. Consider the four masses m_0, m_1, m_2, and m_3 in Figure 2.3.a located at points $\mathbf{P}_0$, $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$.



(a) Center of mass of four points.

(b) Cubic Bézier Curve.

Figure 2.3: Bézier Curves in Terms of Center of Mass.

The equation for the center of mass is

$$\mathbf{P} \;=\; \frac{m_0\mathbf{P}_0 \;+\; m_1\mathbf{P}_1 \;+\; m_2\mathbf{P}_2 \;+\; m_3\mathbf{P}_3}{m_0 \;+\; m_1 \;+\; m_2 \;+\; m_3}.$$

Next, imagine that instead of being fixed, constant values, each mass varies as a function of a parameter t. Specifically, let

$$m_0(t) \;=\; (1-t)^3, \quad m_1(t) \;=\; 3t(1-t)^2, \quad m_2(t) \;=\; 3t^2(1-t), \quad m_3(t) \;=\; t^3. \tag{2.1}$$

Now, for each value of t, the masses assume different weights and their center of mass changes continuously. As t varies between 0 and 1, a curve is swept out by the moving center of mass.

Figure 2.3.b shows the Bézier curve that results when the point masses in Figure 2.3.a are taken as control points. This curve is a cubic Bézier curve — *cubic* because the mass equations are degree three polynomials in t.

Notice that the mass equations in (2.1) sum identically to one:

$$(1-t)^3 + 3t(1-t)^2 + 3t^2(1-t) + t^3 = [(1-t)+t]^3 = 1^3 \equiv 1,$$

and so we can write the equation of this Bézier curve as $\mathbf{P}(t) = m_0(t)\mathbf{P}_0 + m_1(t)\mathbf{P}_1 + m_2(t)\mathbf{P}_2 + m_3(t)\mathbf{P}_3$.

The mass functions are plotted in the graph in Figure 2.4. Note that when $t = 0$, $m_0 = 1$ and $m_1 = m_2 = m_3 = 0$. This explains why the curve passes through $\mathbf{P}_0$. When $t = 1$, $m_3 = 1$ and $m_0 = m_1 = m_2 = 0$, and the curve passes through point $\mathbf{P}_3$.

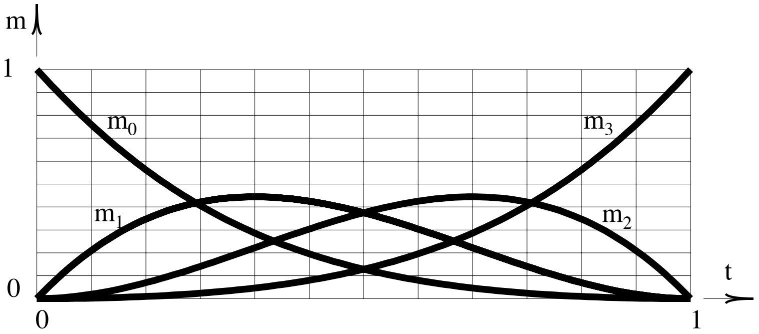


Figure 2.4: Cubic Bézier blending functions.

The variable masses $m_i(t)$ are usually called *blending functions* and, as noted before, the locations $\mathbf{P}_i$ are known as *control points*. The blending functions, in the case of Bézier curves, are known as *Bernstein polynomials*. We will later look at other curves formed with different blending functions.

Bézier curves of any degree can be defined. Figure 2.5 shows sample curves of degree one through four. A degree n Bézier curve has $n+1$ control points whose blending functions are denoted $B_i^n(t)$,

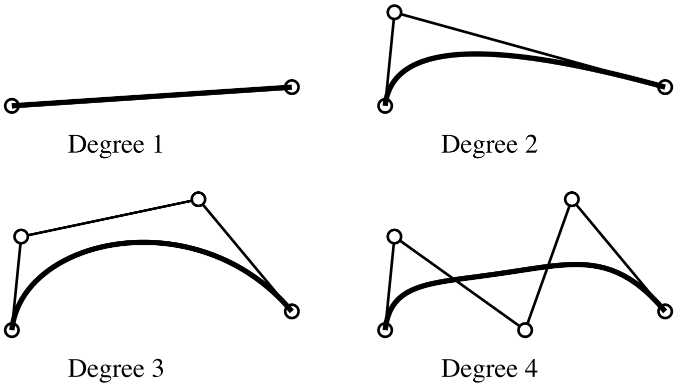


Figure 2.5: Bézier curves of various degree.

where

$$B_i^n(t) \;=\; \binom{n}{i}(1-t)^{n-i}t^i, \quad i = 0, 1, ..., n.$$

Recall that

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}.$$

$\binom{n}{i}$ is spoken "n – *choose* – i" and is called a *binomial coefficient* because it arises in the binomial expansion

$$(a+b)^n = \sum_{i=0}^{n}\binom{n}{i}a^i b^{n-i}.$$

In the degree three case, $n = 3$ and $B_0^3 = (1-t)^3$, $B_1^3 = 3t(1-t)^2$, $B_2^3 = 3t^2(1-t)$ and $B_3^3 = t^3$. $B_i^n(t)$ is also referred to as the *ith* Bernstein polynomial of degree n. The equation of a Bézier curve is thus:

$$\mathbf{P}(t) \;=\; \sum_{i=0}^{n}\binom{n}{i}(1-t)^{n-i}t^i\mathbf{P}_i. \tag{2.2}$$

2.2 Bézier Curves over Arbitrary Parameter Intervals

Equation 2.2 gives the equation of a Bézier curve which starts at $t = 0$ and ends at $t = 1$. It is useful, especially when fitting together a string of Bézier curves, to allow an arbitrary parameter interval:

$$t \in [t_0, t_1]$$

such that $\mathbf{P}(t_0) = \mathbf{P}_0$ and $\mathbf{P}(t_1) = \mathbf{P}_n$. We will denote the Bézier curve defined over an arbitrary parameter interval by $\mathbf{P}_{[t_0,t_1]}(t)$. It's equation is a modification of (2.2):

$$\mathbf{P}_{[t_0,t_1]}(t) \;=\; \frac{\sum_{i=0}^{n}\binom{n}{i}(t_1-t)^{n-i}(t-t_0)^i\mathbf{P}_i}{(t_1-t_0)^n} = \sum_{i=0}^{n}\binom{n}{i}(\frac{t_1-t}{t_1-t_0})^{n-i}(\frac{t-t_0}{t_1-t_0})^i\mathbf{P_i}. \tag{2.3}$$

If no parameter interval is specified, it is assumed to be $[0, 1]$. That is,

$$\mathbf{P}(t) = \mathbf{P}_{[0,1]}(t).$$

2.3 The de Casteljau Algorithm

The de Casteljau algorithm describes how to subdivide a Bézier curve $\mathbf{P}_{[t_0,t_2]}$ into two segments $\mathbf{P}_{[t_0,t_1]}$ and $\mathbf{P}_{[t_1,t_2]}$ whose union is equivalent to $\mathbf{P}_{[t_0,t_2]}$. This algorithm was devised in 1959 by Paul de Casteljau, a French mathematician at the Citroen automobile company. It is an example of a *geometric construction algorithm.*

To facilitate our description of the algorithm, we label the control points of a cubic Bézier curve $\mathbf{P}_{[t_0,t_2]}$ with $\mathbf{P}_0^0$, $\mathbf{P}_1^0$, $\mathbf{P}_2^0$, and $\mathbf{P}_3^0$ as illustrated in Figure 2.6. The algorithm involves computing the sequence of points

$$\mathbf{P}_i^j = (1-\tau)\mathbf{P}_i^{j-1} + \tau\mathbf{P}_{i+1}^{j-1}; \quad j = 1, \ldots, n; \quad i = 0, \ldots, n-j. \tag{2.4}$$

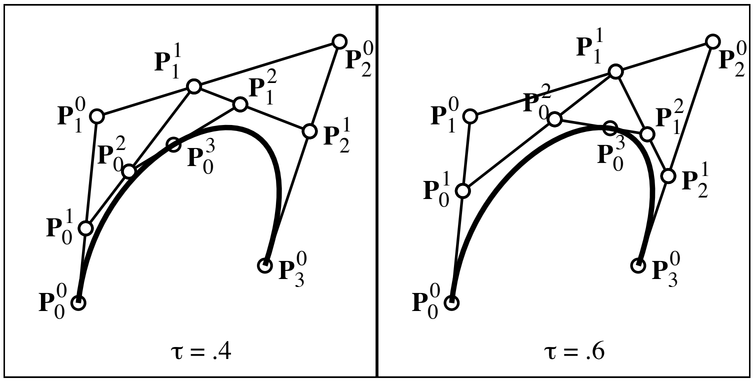


Figure 2.6: Subdividing a cubic Bézier curve.

where $\tau = \frac{t_1 - t_0}{t_2 - t_0}$. Then, the control points for $\mathbf{P}_{[t_0,t_1]}(t)$ are $\mathbf{P}_0^0, \mathbf{P}_0^1, \mathbf{P}_0^2, \ldots, \mathbf{P}_0^n$ and the control points for $\mathbf{P}_{[t_1,t_2]}(t)$ are $\mathbf{P}_0^n, \mathbf{P}_1^{n-1}, \mathbf{P}_2^{n-2}, \ldots, \mathbf{P}_n^0$. Although our example is for a cubic Bézier curve $(n = 3)$, the algorithm works for any degree.

A practical application of the de Casteljau algorithm is that it provides a numerically stable means of computing the coordinates and tangent vector of any point along the curve, since $\mathbf{P}(t_1) = \mathbf{P}_0^n$ and the tangent vector is $\mathbf{P}_1^{n-1} - \mathbf{P}_0^{n-1}$.

Figure 2.7 shows that when a Bézier curve is repeatedly subdivided, the collection of control polygons converge to the curve. Thus, one way of plotting a Bézier curve is to simply subdivide it an appropriate number of times and plot the control polygons (although a more efficient way is to use Horner's algorithm (see Chapter 3) or forward differencing (see Chapter 4)).

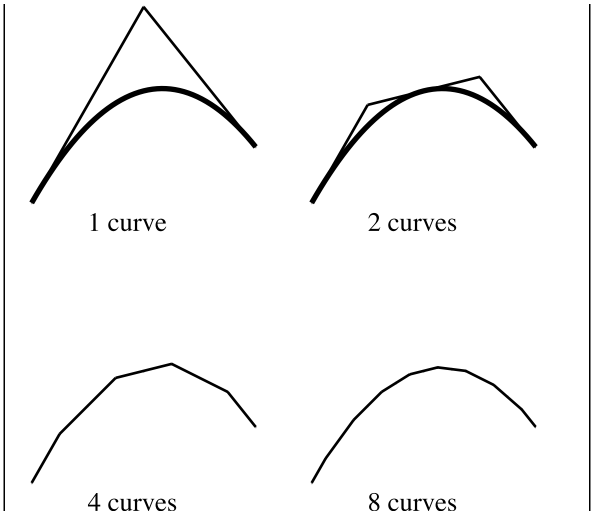


Figure 2.7: Recursively subdividing a quadratic Bézier curve.

The de Casteljau algorithm works even if $\tau \notin [0, 1]$, which is equivalent to $t_1 \notin [t_0, t_2]$. Fig. 2.8 shows a quadratic Bézier curve "subdivided" at $\tau = 2$. The de Castelau algorithm is numerically stable as long as the parameter subdivision parameter is within the parameter domain of the curve.

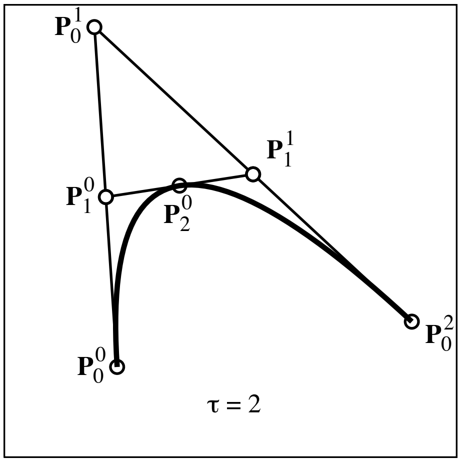


Figure 2.8: Subdividing a quadratic Bézier curve.

2.4 Degree Elevation

Any degree n Bézier curve can be exactly represented as a Bézier curve of degree $n+1$ (and hence, as a curve of any degree $m > n$). Given a degree n Bézier curve, the procedure for computing the control points for the equivalent degree $n+1$ Bézier curve is called *degree elevation*.

We now derive the degree elevation formula, using the identities:

$$\begin{aligned}
(1-t)B_i^n(t) &= (1-t)\binom{n}{i}(1-t)^{n-i}t^i \\
&= \binom{n}{i}(1-t)^{n-i+1}t^i \\
&= \frac{\binom{n}{i}}{\binom{n+1}{i}}\binom{n+1}{i}(1-t)^{n-i+1}t^i \\
&= \frac{\binom{n}{i}}{\binom{n+1}{i}}B_i^{n+1} = \frac{\frac{n!}{i!(n-i)!}}{\frac{(n+1)!}{i!(n+1-i)!}}B_i^{n+1} \\
&= \frac{n!i!(n+1-i)!}{i!(n-i)!(n+1)!}B_i^{n+1} = \frac{n!i!(n+1-i)(n-i)!}{i!(n-i)!(n+1)n!}B_i^{n+1} \\
&= \frac{n+1-i}{n+1}B_i^{n+1} \qquad (2.5)
\end{aligned}$$

and

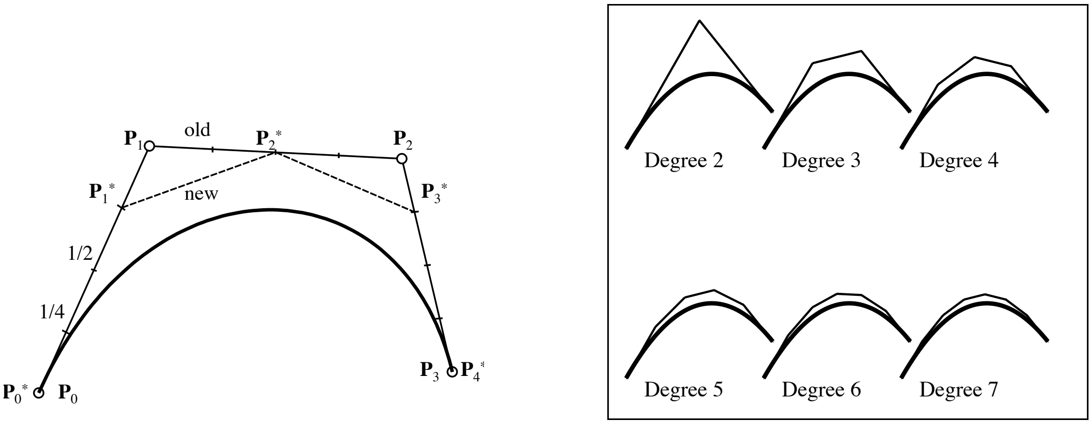
$$tB_i^n(t) = \frac{i+1}{n+1}B_{i+1}^{n+1}(t). \qquad (2.6)$$

Degree elevation is accomplished by simply multiplying the equation of the degree n Bézier curve by $[(1-t)+t]=1$:

$$
\begin{aligned}
\mathbf{P}(t) &= [(1-t)+t]\mathbf{P}(t) \\
&= [(1-t)+t]\sum_{i=0}^{n}\mathbf{P}_i B_i^n(t) \\
&= \sum_{i=0}^{n}\mathbf{P}_i[(1+t)B_i^n(t)+tB_i^n(t)] \\
&= \sum_{i=0}^{n}\mathbf{P}_i[\frac{n+1-i}{n+1}B_i^{n+1}+\frac{i+1}{n+1}B_{i+1}^{n+1}(t)] \\
&= \sum_{i=0}^{n}\frac{n+1-i}{n+1}\mathbf{P}_i B_i^{n+1}+\sum_{i=0}^{n}\frac{i+1}{n+1}\mathbf{P}_i B_{i+1}^{n+1}(t) \\
&= \sum_{i=0}^{n}\frac{n+1-i}{n+1}\mathbf{P}_i B_i^{n+1}+\sum_{i=1}^{n+1}\frac{i}{n+1}\mathbf{P}_{i-1} B_{i}^{n+1}(t) \\
&= \sum_{i=0}^{n+1}\frac{n+1-i}{n+1}\mathbf{P}_i B_i^{n+1}+\sum_{i=0}^{n+1}\frac{i}{n+1}\mathbf{P}_{i-1} B_{i}^{n+1}(t) \\
&= \sum_{i=0}^{n+1}[\frac{(n+1-i)\mathbf{P}_i+i\mathbf{P}_{i-1}}{n+1}]B_i^{n+1}(t) \\
&= \sum_{i=0}^{n+1}\mathbf{P}_i^* B_i^{n+1}(t) \qquad (2.7)
\end{aligned}
$$

where

$$\mathbf{P}_i^* \;=\; \alpha_i\mathbf{P}_{i-1} \;+\; (1-\alpha_i)\mathbf{P}_i, \qquad \alpha_i \;=\; \frac{i}{n+1}. \qquad (2.8)$$



(a) Degree Elevation of a Cubic Bézier Curve.

(b) Repeated degree elevation.

Figure 2.9: Degree Elevation of a Bézier Curve.

For the case $n = 3$, the new control points $\mathbf{P}_i^*$ are:

$$\begin{aligned}
\mathbf{P}_0^* &= \mathbf{P}_0 \\
\mathbf{P}_1^* &= \frac{1}{4}\mathbf{P}_0 + \frac{3}{4}\mathbf{P}_1 \\
\mathbf{P}_2^* &= \frac{2}{4}\mathbf{P}_1 + \frac{2}{4}\mathbf{P}_2 \\
\mathbf{P}_3^* &= \frac{3}{4}\mathbf{P}_2 + \frac{1}{4}\mathbf{P}_3 \\
\mathbf{P}_4^* &= \mathbf{P}_3
\end{aligned}$$

Figure 2.9.a illustrates.

If degree elevation is applied repeatedly, as shown in Figure 2.9.b, the control polygon converges to the curve itself.

2.5 The Convex Hull Property of Bézier Curves

An important property of Bézier curves is that they always lie within the convex hull of their control points. The convex hull can be envisioned by pounding an imaginary nail into each control point, stretching an imaginary rubber band so that it surrounds the group of nails, and then collapsing that rubber band around the nails. The polygon created by that imaginary rubber band is the convex hull. Figure 2.10 illustrates.

The fact that Bézier curves obey the convex hull property is assured by the center-of-mass definition of Bézier curves in Section 2.1. Since all of the control points lie on one side of an edge of the convex hull, it is impossible for the center of mass of those control points to lie on the other side of the line,

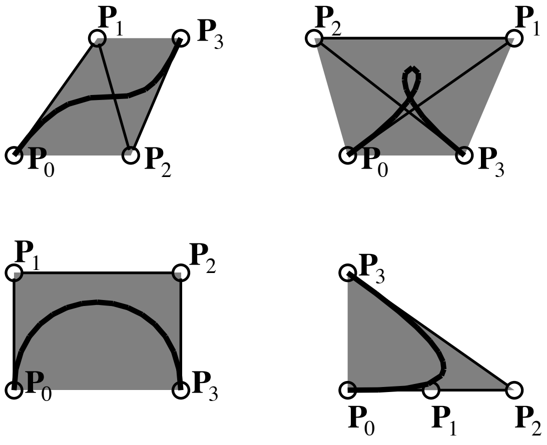


Figure 2.10: Convex Hull Property

2.6 Distance between Two Bézier Curves

The problem often arises of determining how closely a given Bézier curve is approximated by a second Bézier curve. For example, if a given cubic curve can be adequately represented by a degree elevated quadratic curve, it might be advantageous to replace the cubic curve with the quadratic.

Given two Bézier curves

$$\mathbf{P}(t) = \sum_{i=0}^{n} \mathbf{P}_i B_i^n(t); \qquad \mathbf{Q}(t) = \sum_{i=0}^{n} \mathbf{Q}_i B_i^n(t)$$

the vector $\mathbf{P}(t) - \mathbf{Q}(t)$ between points of equal parameter value on the two curves can itself be expressed as a Bézier curve

$$\mathbf{D}(t) = \mathbf{P}(t) - \mathbf{Q}(t) = \sum_{i=0}^{n} (\mathbf{P}_i - \mathbf{Q}_i) B_i^n(t)$$

whose control points are $\mathbf{D}_i = \mathbf{P}_i - \mathbf{Q}_i$. The vector from the origin to the point $\mathbf{D}(t)$ is $\mathbf{P}(t) - \mathbf{Q}(t)$. The convex hull property guarantees that the distance between the two curves is bounded by the largest distance from the origin to any of the control points $\mathbf{D}_i$.

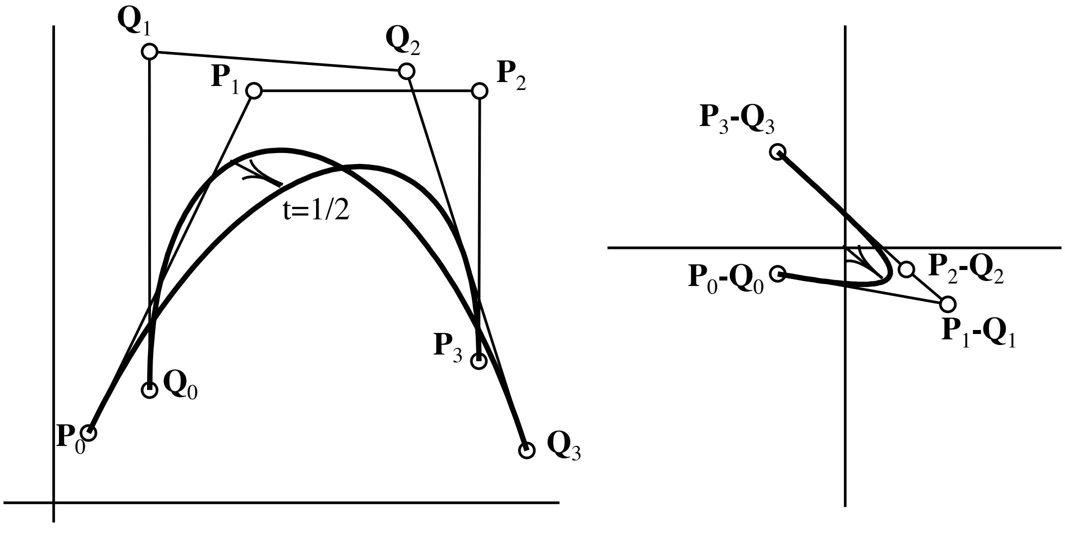


Figure 2.11: Difference curve.

This error bound is attractive because it is very easy to compute. However, it is not always a very tight bound because it is dependent on parametrization.

A more precise statement of the distance between two curves is the *Hausdorff* distance. Given two curves $\mathbf{P}_{[s_0,s_1]}(s)$ and $\mathbf{Q}_{[t_0,t_1]}(t)$. The Hausdorff distance $d(\mathbf{P}, \mathbf{Q})$ is defined

$$d(\mathbf{P}, \mathbf{Q}) = \max_{s \in [s_0,s_1]} \left\{ \min_{t \in [t_0,t_1]} |\mathbf{P}(s) - \mathbf{Q}(t)| \right\} \tag{2.9}$$

where $|\mathbf{P}(s) - \mathbf{Q}(t)|$ is the Euclidean distance between a point $\mathbf{P}(s)$ and the point $\mathbf{Q}(t)$. In practice, it is much more expensive to compute the Hausdorff distance than to compute a bound using the difference curve in Figure 2.11.

2.7 Derivatives

The parametric derivatives of a Bézier curve can be determined geometrically from its control points. For a polynomial Bézier curve $\mathbf{P}_{[t_0,t_1]}(t)$ of degree n with control points $\mathbf{P}_i$, the first parametric derivative can be expressed as a polynomial Bézier curve of degree $n-1$ with control points $\mathbf{D}_i$ where

$$\mathbf{D}_i \;=\; \frac{n}{t_1-t_0}(\mathbf{P}_{i+1} \;-\; \mathbf{P}_i).$$

For example, the cubic Bézier curve in Fig. 2.12 has a first derivative curve as shown. The first derivative vector $\mathbf{P}'(\tau)$ (that is, the first derivative $\mathbf{P}(t)$ evaluated at $t=\tau$) is tangent to the curve curve $\mathbf{P}(t)$ at the point $\mathbf{P}(\tau)$. This is illustrated in Fig. 2.12 for the point $\tau \;=\; .3$. In this example, $t_0=0$ and $t_1=1$.

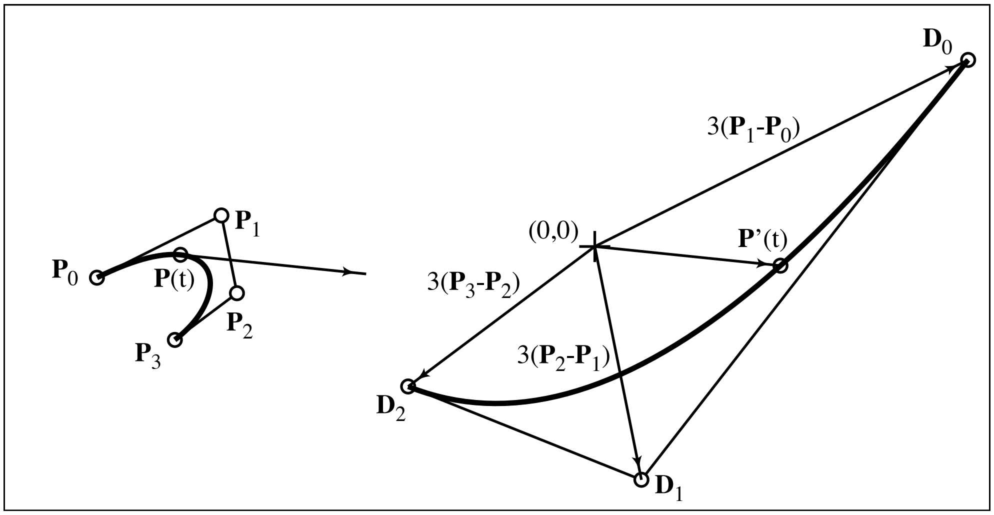


Figure 2.12: Hodograph.

The first derivative curve is known as a *hodograph.* We can compute the second derivative of a Bézier curve by taking the hodograph of the hodograph (or, the second hodograph), etc. It is convenient to compute hodographs in tabular form. For example, consider a cubic Bézier curve $\mathbf{P}_{[0,\frac{1}{2}]}(t)$ with control points

$$\mathbf{P}_0=(2,1),\quad \mathbf{P}_1=(4,5),\quad \mathbf{P}_2=(8,6),\quad \mathbf{P}_3=(9,2).$$

The control points of the first, second, and third hodographs are the rows of the following table:

Curve	Control Points			
$\mathbf{P}(t)$	$(2,1)$	$(4,5)$	$(8,6)$	$(9,2)$
$\mathbf{P}'(t)$	$(12,24)$	$(24,6)$	$(6,-24)$	
$\mathbf{P}''(t)$	$(48,-72)$	$(-72,-120)$		
$\mathbf{P}'''(t)$	$(-240,-96)$			

From this table, we can obtain the values of all derivatives at $t=t_0$ and $t=t_1$. For example, $p'(1/2)=(6,-24)$, $\mathbf{P}''(0)=(48,-72)$, the the third derivative vector is a constant $\mathbf{P}'''(t)\equiv(-240,-96)$.

It is noteworthy that if the hodograph passes through the origin, there is a cusp corresponding to that point on the original curve!

Note that the hodograph we have just described relates only to polynomial Bézier curves, *not* to rational Bézier curves or any other curve that we will study. The derivative of any other curve must be computed by differentiation. For a rational Bézier curve, that differentiation will involve the quotient rule. Consequently, the derivative of a degree n rational Bézier curve can be expressed as a rational Bézier curve of degree $2n$. As it turns out, the equations for those control points are rather messy.

2.8 Three Dimensional Bézier Curves

If Bézier control points are defined in three dimensional space, the resulting Bézier curve is three dimensional. Such a curve is sometimes called a *space* curve, and a two dimensional curve is called a *planar* curve. Our discussion of the de Casteljau algorithm, degree elevation, and hodographs extend to 3D without modification.

Since a degree two Bézier curve is defined using three control points, every degree two curve is planar, even if the control points are in a three dimensional coordinate system.

2.9 Rational Bézier Curves

A rational Bézier curve is one for which each control point $\mathbf{P}_i$ is assigned a scalar *weight*. The equation of a rational Bézier curve is

$$\frac{\sum_{i=0}^{n} w_i B_i^n(t)\mathbf{P}_i}{\sum_{i=0}^{n} w_i B_i^n(t)}.$$

The effect of changing a control point weight is illustrated in Fig. 2.13. This type of curve is known as a *rational Bézier curve*, because the blending functions are rational polynomials, or the *ratio* of two polynomials. Note that if all weights are 1 (or if all weights are simply the same), a rational Bézier curve reduces to a *polynomial* Bézier curve.

Rational Bézier curves have several advantages over polynomial Bézier curves. Clearly, rational Bézier curves provide more control over the shape of a curve than does a polynomial Bézier curve. In addition, a perspective drawing of a 3D Bézier curve (polynomial or rational) is a rational Bézier curve, not a polynomial Bézier curve. Also, rational Bézier curves are needed to exactly express all conic sections. A degree two polynomial Bézier curve can only represent a parabola. Exact representation of circles requires rational degree two Bézier curves.

A rational Bézier curve can be interpreted as the perspective projection of a 3-D polynomial curve. Fig. 2.14 shows two curves: a 3-D curve and a 2-D curve. The 2-D curve lies on the plane $z = 1$ and it is defined as the projection of the 3-D curve onto the plane $z = 1$. One way to consider this is to imagine a funny looking cone whose vertex is at the origin and which contains the 3-D curve. In other words, this cone is the collection of all lines which contain the origin and a point on the curve. Then, the 2-D rational Bézier curve is the intersection of the cone with the plane $z = 1$.

What we see here can be viewed as the geometric interpretation of homogeneous coordinates discussed in Section 7.1.1. The 3D Bézier control points $w_i(x_i, y_i, 1) = (x_i w_i, y_i w_i, w_i)$ can be thought of as homogeneous coordinates (X, Y, Z) that correspond to 2D Cartesian coordinates $(\frac{X}{Z}, \frac{Y}{Z})$. If the 2-D rational Bézier curve has control points (x_i, y_i) with corresponding weights w_i, then the homogeneous (X, Y, Z) coordinates of the 3-D control points are $w_i(x_i, y_i, 1) = (x_i w_i, y_i w_i, w_i)$. Denote

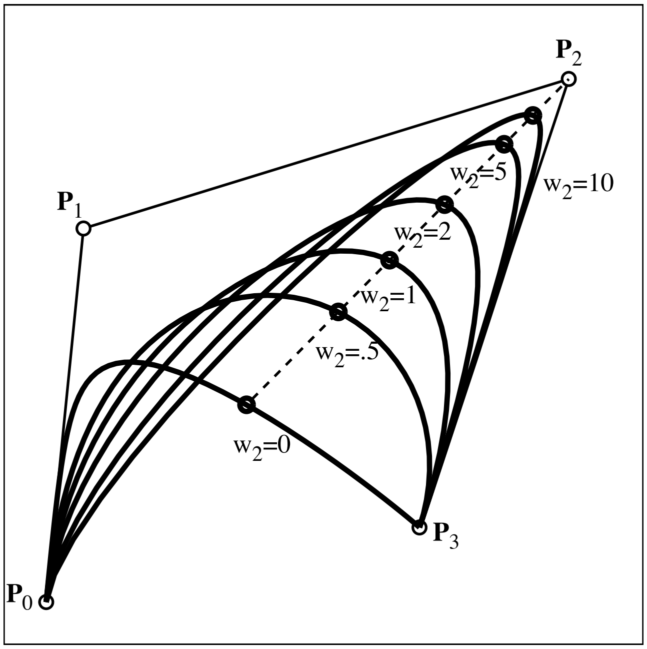


Figure 2.13: Rational Bézier curve.

points on the 3-D curve using upper case variables $(X(t), Y(t), Z(t))$ and on the 2-D curve using lower case variables $(x(t), y(t))$. Then, any point on the 2-D rational Bézier curve can be computed by computing the corresponding point on the 3-D curve, $(X(t), Y(t), Z(t))$, and projecting it to the plane $z = 1$ by setting

$$x(t) = \frac{X(t)}{Z(t)}, \quad y(t) = \frac{Y(t)}{Z(t)}.$$

2.9.1 De Casteljau Algorithm and Degree Elevation on Rational Bézier Curves

The de Casteljau algorithm and degree elevation algorithms for polynomial Bézier curves extend easily to rational Bézier curves as follows. First, convert the rational Bézier curve into its corresponding polynomial 3D Bézier curve as discussed in Section 2.9. Next, perform the de Casteljau algorithm or degree elevation on the 3D polynomial Bézier curve. Finally, map the resulting 3D Bézier curve back to 2D. The Z coordinates of the control points end up as the weights of the control points for the 2D rational Bézier curve.

This procedure does not work for hodographs, since the derivative of a rational Bézier curve requires the use of the quotient rule for differentiation. However, Section 2.9.2 describes how to compute the first derivative at the endpoint of a rational Bézier curve and Section 2.9.3 describes how to compute the curvature at the endpoint of a rational Bézier curve. Used in conjunction with the de Casteljau algorithm, this enable us to compute the first derivative or curvature at any point on a rational Bézier curve.

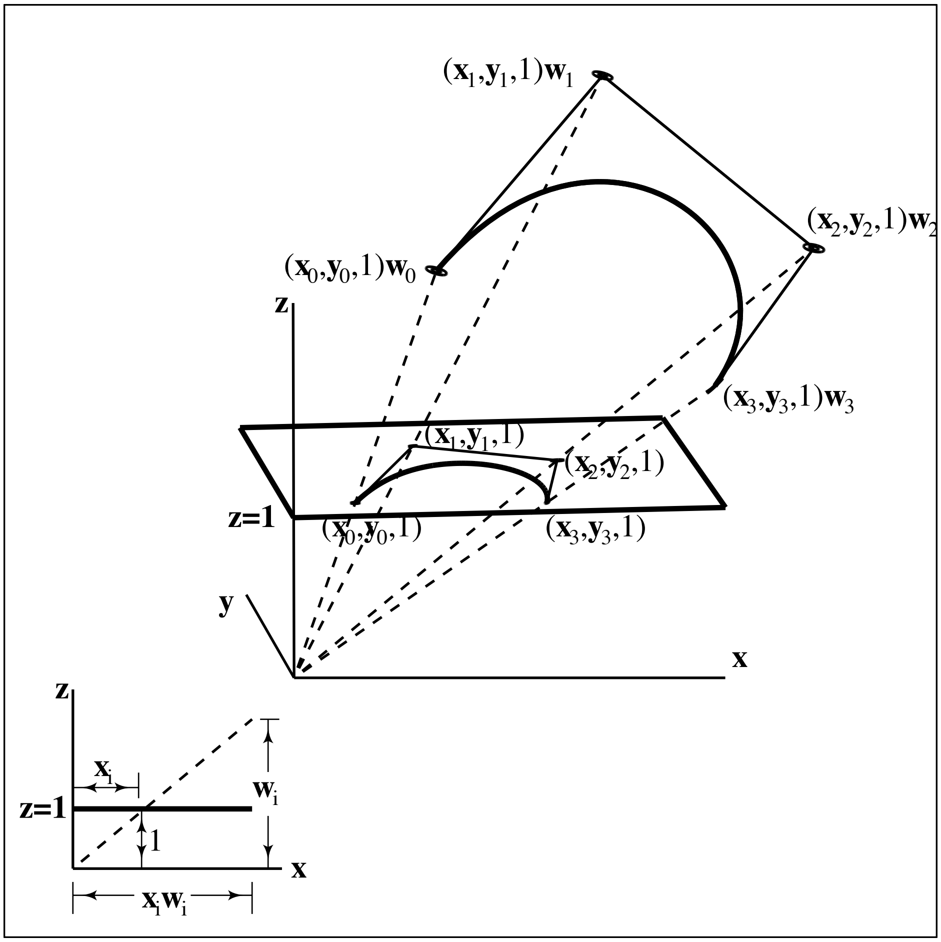


Figure 2.14: Rational curve as the projection of a 3-D curve.

2.9.2 First Derivative at the Endpoint of a Rational Bézier Curve

The derivative equation for a rational Bézier involves the quotient rule for derivatives — it does not work to simply compute the tangent vector and curvature for the three dimensional non-rational Bézier curve and then project that value to the (x, y) plane. For a degree n rational Bézier curve $\mathbf{P}_{[0,1]}(t)$,

$$x(t) = \frac{x_n(t)}{d(t)} =$$

$$\frac{w_0 x_0 \binom{n}{0}(1-t)^n + w_1 x_1 \binom{n}{1}(1-t)^{n-1}t + w_2 x_2 \binom{n}{2}(1-t)^{n-2}t^2 + \dots}{w_0 \binom{n}{0}(1-t)^n + w_1 \binom{n}{1}(1-t)^{n-1}t + w_2 \binom{n}{2}(1-t)^{n-2}t^2 + \dots};$$

$$y(t) = \frac{y_n(t)}{d(t)} =$$

$$\frac{w_0 y_0 \binom{n}{0}(1-t)^n + w_1 y_1 \binom{n}{1}(1-t)^{n-1}t + w_2 y_2 \binom{n}{2}(1-t)^{n-2}t^2 + \dots}{w_0 \binom{n}{0}(1-t)^n + w_1 \binom{n}{1}(1-t)^{n-1}t + w_2 \binom{n}{2}(1-t)^{n-2}t^2 + \dots}$$

the equation for the first derivative vector at $t = 0$ must be found by evaluating the following equations:

$$\dot{x}(0) = \frac{d(0)\dot{x}_n(0) - \dot{d}(0)x_n(0)}{d^2(0)}; \qquad \dot{y}(0) = \frac{d(0)\dot{y}_n(0) - \dot{d}(0)y_n(0)}{d^2(0)}$$

from which

$$\mathbf{P}'(0) = \frac{w_1}{w_0} n(\mathbf{P}_1 - \mathbf{P}_0). \tag{2.10}$$

For a rational curve $\mathbf{P}_{[t_0,t_1]}(t)$, the first derivative at $t = t_0$ is:

$$\mathbf{P}'(t_0) = \frac{w_1}{w_0}\frac{n}{t_1 - t_0}(\mathbf{P}_1 - \mathbf{P}_0). \tag{2.11}$$

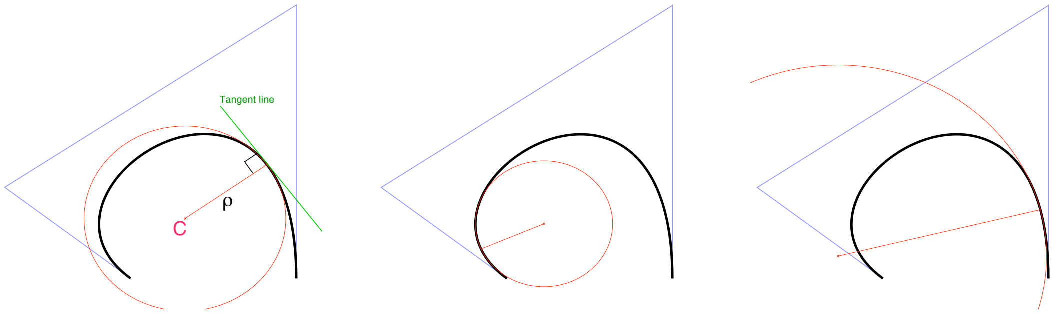
The second derivative of a rational Bézier curve at its endpoint is

$$\mathbf{P}''(t_0) = \frac{n(n-1)}{(t_1-t_0)^2}\frac{w_2}{w_0}(\mathbf{P}_2 - \mathbf{P}_0) - \frac{2n}{(t_1-t_0)^2}\frac{w_1}{w_0}\frac{nw_1 - w_0}{w_0}(\mathbf{P}_1 - \mathbf{P}_0) \tag{2.12}$$

2.9.3 Curvature at an Endpoint of a Rational Bézier Curve

The curvature (denoted by κ) of a curve is a measure of how sharply it curves. The curvature of a circle of radius ρ is defined to be $\kappa = 1/\rho$. A straight line does not curve at all, and its curvature is zero. For other curves, the curvature is generally different at each point on the curve. The meaning of curvature in this general case is illustrated by Figure 2.15 which shows a Bézier curve and the *osculating circle* with respect to a point on the curve. The word "osculate" means to kiss, and the osculating circle is the circle that "best fits" the curve at a particular point in the following sense. An arbitrary line that passes through a point on a curve intersects the curve with multiplicity one; the tangent line at that point intersects the curve with multiplicity two. An osculating circle is a circle of radius ρ that intersects the curve with multiplicity at least three, and the curvature κ is defined to be $\kappa = 1/\rho$. (See Section 7.3.1 for a description of intersection multiplicity.) ρ is called the radius of curvature for the curve at that point.

In Figure 2.15, the center of the circle, $\mathbf{C}$, is located along the normal line a distance ρ from the point of contact.



(a) At $t = .7$

(b) At $t = .1$

(c) At $t = .8$

Figure 2.15: Osculating Circles.

The equation for the curvature of a parametric curve is

$$\kappa = \frac{|\dot{x}\ddot{y} - \dot{y}\ddot{x}|}{(\dot{x}^2 + \dot{y}^2)^{\frac{3}{2}}}$$

where the dots denote differentiation with respect to the curve parameter. If this equation is applied to an endpoint of a rational Bézier curve, a beautifully simple and geometrically meaningful equation results:

$$\kappa(t_0) = \frac{w_0 w_2}{w_1^2} \frac{n-1}{n} \frac{h}{a^2} \tag{2.13}$$

where n is the degree of the curve and a and h are as shown in Fig. 2.16 (a is the length of the first leg of the control polygon, and h is the perpendicular distance from $\mathbf{P}_2$ to the first leg of the control polygon). Curvature is independent of $[t_0, t_1]$.

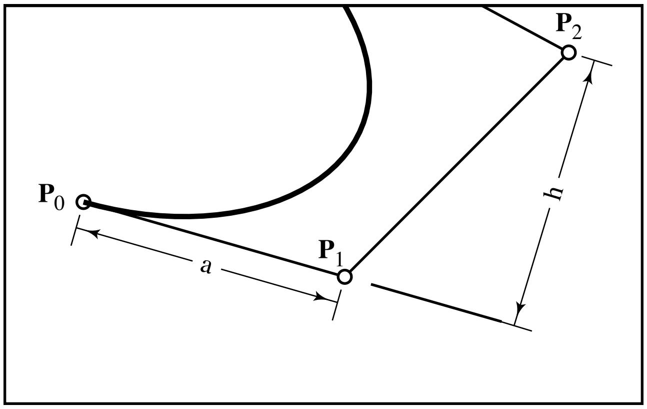


Figure 2.16: Endpoint curvature.

While is is mathematically possible to write down an equation for the first derivative vector and for the curvature of a rational Bézier curve as a function of t, such equations would be extremely complicated. The best way to find the curvature or first derivative vector at an arbitrary point along a rational Bézier curve is to first perform the de Casteljau algorithm (Section 2.3) to make the desired point an endpoint of the curve, and then apply (2.11) or (2.13).

Example A rational quadratic Bézier curve has control points and weights:

$$\mathbf{P}_0 = (0,0),\ w_0 = 1; \quad \mathbf{P}_1 = (4,3),\ w_1 = 2; \quad \mathbf{P}_2 = (0,5),\ w_2 = 4.$$

What is the curvature at $t = 0$?

Solution
The distance equation for the line $\mathbf{P}_0\mathbf{P}_1$ is $\frac{3}{5}x - \frac{4}{5}y$, so plugging the Cartesian coordinates of $\mathbf{P}_2$ into this distance equation give $h = 4$. The value of a is found as the distance from $\mathbf{P}_0$ to $\mathbf{P}_1$, which is $\sqrt{4^2+3^2} = 5$. So, the curvature is

$$\kappa(0) = \frac{w_0 w_2}{w_1^2}\frac{n-1}{n}\frac{h}{a^2} = \frac{1\cdot 4}{2^2}\frac{1}{2}\frac{4}{5^2} = \frac{2}{25}$$

2.10 Continuity

Two curve segments $\mathbf{P}_{[t_0,t_1]}$ and $\mathbf{Q}_{[t_1,t_2]}$ are said to be C^k continuous (or, to have k^{th} order parametric continuity) if

$$\mathbf{P}(t_1) = \mathbf{Q}(t_1),\ \mathbf{P}'(t_1) = \mathbf{Q}'(t_1),\ \ldots,\ \mathbf{P}^{(k)}(t_1) = \mathbf{Q}^{(k)}(t_1). \tag{2.14}$$

Thus, C^0 means simply that the two adjacent curves share a common endpoint. C^1 means that the two curves not only share the same endpoint, but also that they have the same tangent vector at their shared endpoint, in magnitude as well as in direction. C^2 means that two curves are C^1 and in addition that they have the same second order parametric derivatives at their shared endpoint, both in magnitude and in direction.

In Fig. 2.17, the two curves $\mathbf{P}_{[t_0,t_1]}(t)$ and $\mathbf{Q}_{[t_1,t_2]}(t)$ are at least C^0 because $\mathbf{p}_3 \equiv \mathbf{q}_0$. Furthermore, they are C^1 if

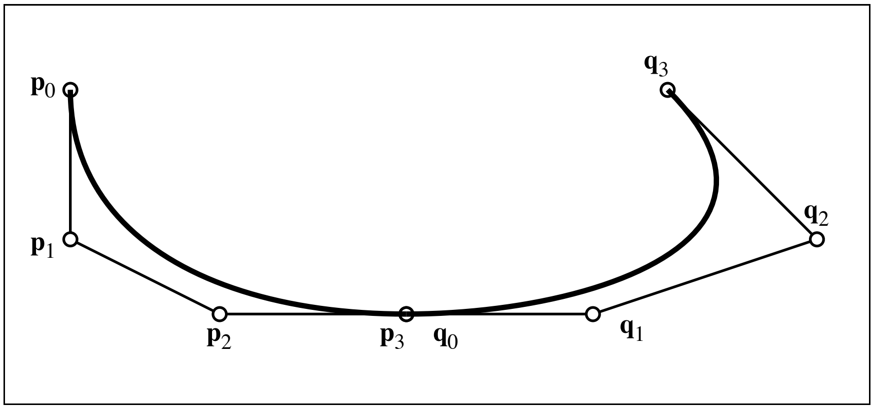


Figure 2.17: C^2 Bézier curves.

$$\frac{3}{t_1 - t_0}(\mathbf{p}_3 - \mathbf{p}_2) = \frac{3}{t_2 - t_1}(\mathbf{q}_1 - \mathbf{q}_0).$$

They are C^2 if they are C^1 and

$$\frac{6}{(t_1-t_0)^2}(\mathbf{p}_3 - 2\mathbf{p}_2 + \mathbf{p}_1) = \frac{6}{(t_2-t_1)^2}(\mathbf{q}_2 - 2\mathbf{q}_1 + \mathbf{q}_0).$$

A second method for describing the continuity of two curves, that is independent of their parametrization, is called geometric continuity and is denoted G^k. The conditions for geometric continuity (also known as *visual* continuity) are less strict than for parametric continuity. For G^0 continuity, we simply require that the two curves have a common endpoint, but we do not require that they have the same parameter value at that common point. For G^1, we require that line segments $\mathbf{p}_2 - \mathbf{p}_3$ and $\mathbf{q}_0 - \mathbf{q}_1$ are collinear, but they need not be of equal length. This means that they have a common tangent line, though the magnitude of the tangent vector may be different. G^2 (second order geometric continuity) means that the two neighboring curves have the same tangent line and also the same center of curvature at their common boundary. A general definition of G^n is that two curves that are G^n can be reparametrized (see Section 2.13) to make them be C^n. Two curves which are C^n are also G^n, as long as (2.15) holds.

$$\mathbf{P}(t_1) = \mathbf{Q}(t_1),\ \mathbf{P}'(t_1) = \mathbf{Q}'(t_1) \neq 0,\ \ldots,\ \mathbf{P}^{(k)}(t_1) = \mathbf{Q}^{(k)}(t_1) \neq 0. \qquad (2.15)$$

Second order continuity (C^2 or G^2) is often desirable for both physical and aesthetic reasons. One reason that cubic NURBS curves and surfaces are an industry standard in CAD is that they are C^2. Two surfaces that are G^1 but not G^2 do not have smooth reflection lines. Thus, a car body made up of G^1 surfaces will not look smooth in a show room. Railroad cars will jerk where G^1 curves meet.

2.11 Curvature Combs

A curvature comb is a display mode used in commercial CAD software. Artists use curvature combs as feedback in designing aesthetically pleasing curves. The comb is a series of line segments (call them *teeth*) that are drawn perpendicular to the curve, and whose lengths are proportional to the curvature. The longest tooth will correspond to the point on the curve with the largest curvature.

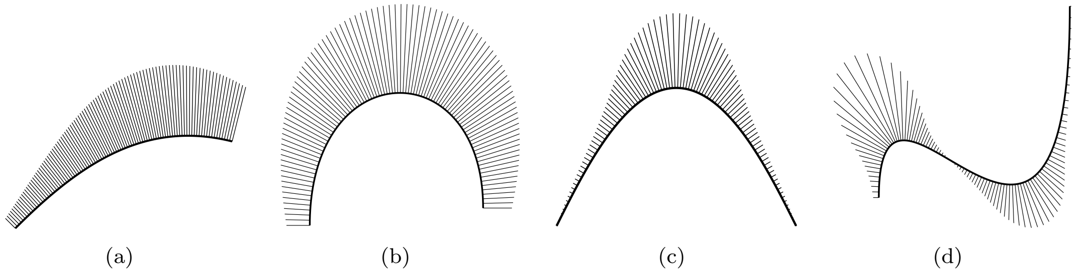
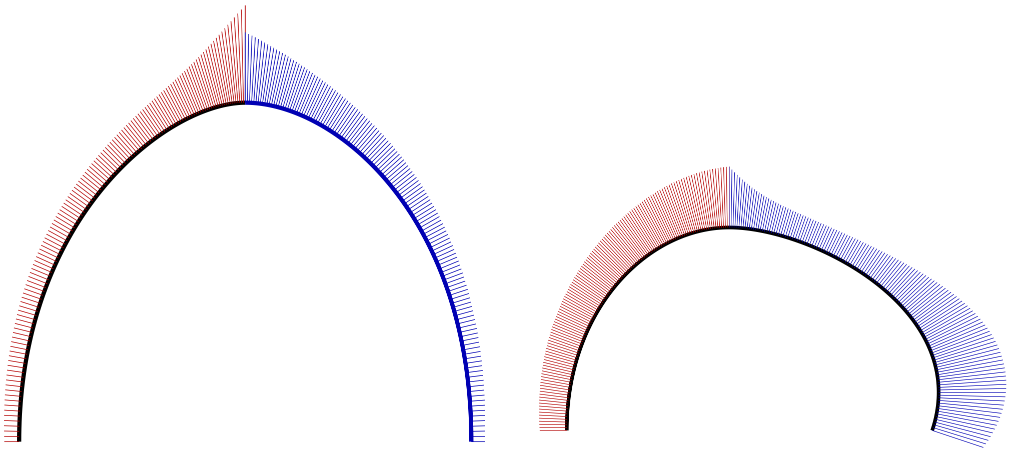

(a) (b) (c) (d)

Figure 2.18: Curvature combs.

To implement the drawing of a curvature comb that has n teeth, first compute the curvature $\kappa_i,\ i = 1, \ldots, n$ for the n points on the curve corresponding to the teeth. Set κ_{max} to the largest of the κ_i. The length of the longest tooth is usually a user-specified value, call it d. Then, the length of tooth i is simply $d\frac{\kappa_i}{\kappa_{max}}$.

As a general rule, artists consider that the most aesthetically pleasing curves have curvature combs whose length changes monotonically and smoothly. The line segments are drawn in the direction away from the center of curvature, as illustrated in Figure 2.18.

The continuity of two curves $\mathbf{P}$ and $\mathbf{Q}$ can be visualized in terms of their respective curvature combs. Define a *comb curve* to be the set of endpoints of a comb. Denote by $\mathbf{P}_c$ the curve comb of $\mathbf{P}$ and by $\mathbf{Q}_c$ the curve comb of $\mathbf{Q}$. Then, if $\mathbf{P}$ and $\mathbf{Q}$ are G_1, $\mathbf{P}_c$ and $\mathbf{Q}_c$ are discontinuous, as shown in Figure 2.19.a. If $\mathbf{P}$ and $\mathbf{Q}$ are G^2, $\mathbf{P}_c$ and $\mathbf{Q}_c$ are G^0, as shown in Figure 2.19.b.



(a) Original curves are G^1. Comb curves are discontinuous.

(b) Original curves are G^2. Comb curves are G^0.

Figure 2.19: Curvature combs for two piecewise curves.

To get comb curves that are mathematically G^1, $\mathbf{P}$ and $\mathbf{Q}$ must be G^3. In practice, some commercial software advertises that their curves and surfaces are G^3 or C^3. This is often marketing jargon; what they really mean is that the curvature combs look like they are G^1 when in reality, $\mathbf{P}$ and $\mathbf{Q}$ may not even be exactly G^1 or even G^0.

2.12 Circular Arcs

Circular arcs can be exactly represented using rational Bézier curves. Figure 2.20 shows a circular arc as both a degree two and a degree three rational Bézier curve. Of course, the control polygons are tangent to the circle. The degree three case is a simple degree elevation of the degree two case. The length e is given by

$$e = \frac{2\sin\frac{\theta}{2}}{1 + 2\cos\frac{\theta}{2}} r.$$

The degree two case has trouble when θ approaches 180° since $\mathbf{P}_1$ moves to infinity, although this can be remedied by just using homogeneous coordinates. The degree three case has the advantage that it can represent a larger arc, but even here the length e goes to infinity as θ approaches 240°. For large arcs, a simple solution is to just cut the arc in half and use two cubic Bézier curves. A complete circle can be represented as a degree five Bézier curve as shown in Figure 2.21. Here, the weights are $w_0 = w_5 = 1$ and $w_1 = w_2 = w_3 = w_4 = \frac{1}{5}$.

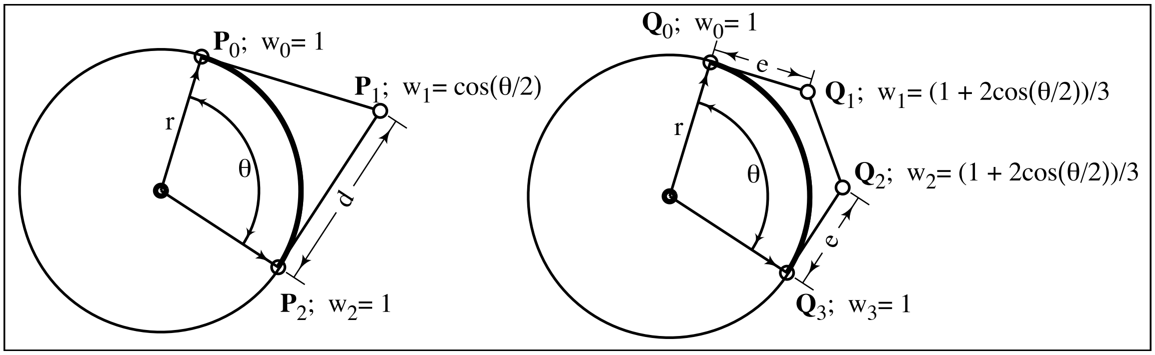

$\mathbf{P}_0$; w_0= 1
$\mathbf{P}_1$; w_1= cos(θ/2)
$\mathbf{P}_2$; w_2= 1
r
θ
d
$\mathbf{Q}_0$; w_0= 1
e
$\mathbf{Q}_1$; w_1= (1 + 2cos(θ/2))/3
$\mathbf{Q}_2$; w_2= (1 + 2cos(θ/2))/3
$\mathbf{Q}_3$; w_3= 1
r
θ
e

Figure 2.20: Circular arcs.

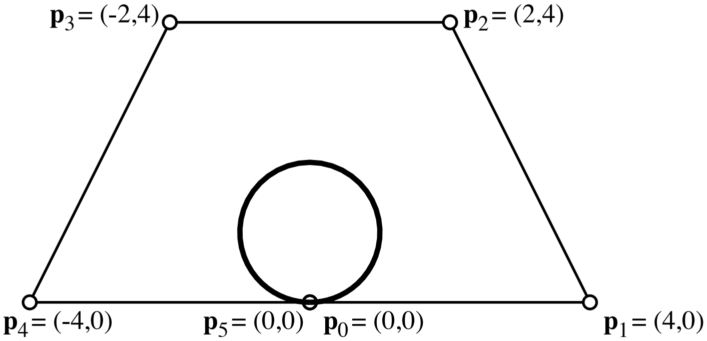


Figure 2.21: Circle as Degree 5 Rational Bézier Curve.

2.13 Reparametrization of Bézier Curves

Reparametrization of a curve means to change how a curve is parametrized, i.e., to change which parameter value is assigned to each point on the curve. Reparametrization can be performed by a parameter substitution. Given a parametric curve $\mathbf{P}(t)$, if we make the substitution $t = f(s)$, we end up with a curve $\mathbf{Q}(s)$ which is geometrically identical to $\mathbf{P}(t)$. For example, let

$$\mathbf{P}(t) = (t^2 + t - 1, 2t^2 - t + 3), \quad t = 2s + 1$$

yields

$$\mathbf{Q}(s) = ((2s+1)^2 + (2s+1) - 1, 2(2s+1)^2 - (2s+1) + 3) = (4s^2 + 6s + 1, 8s^2 + 6s + 4).$$

The curves $\mathbf{P}(t)$ and $\mathbf{Q}(s)$ consist of the same points (if the domains are infinite) but the correspondence between parameter values and points on the curve are different.

A simple (and obvious) way to reparametrize a Bézier curve $\mathbf{P}_{[t_0,t_1]}(t)$ is to simply change the values of t_0 and t_1 without changing the control points. Thus, two curves $\mathbf{P}_{[t_0,t_1]}(t)$ and $\mathbf{Q}_{[s_0,s_1]}(s)$ that have the same degree, control point positions, and weights, but for which $t_0 \neq s_0$ and/or $t_1 \neq s_1$ are reparametrizations of each other. The reparametrization function is

$$t = \frac{t_0 s_1 - s_0 t_1}{s_1 - s_0} + \frac{t_1 - t_0}{s_1 - s_0} s, \quad \text{or } s = \frac{s_0 t_1 - t_0 s_1}{t_1 - t_0} + \frac{s_1 - s_0}{t_1 - t_0} t.$$

Example
Bézier curves $\mathbf{P}_{[0,1]}(t)$ and $\mathbf{Q}_{[5,9]}(s)$ have the same degree, control point positions, and weights. The curves look the same, but have different parametrizations. For example,

$$\mathbf{P}(0) = \mathbf{Q}(5), \quad \mathbf{P}(1) = \mathbf{Q}(9), \quad \mathbf{P}(.5) = \mathbf{Q}(7).$$

In general,

$$\mathbf{P}(t) \equiv \mathbf{Q}(5 + 4t) \text{ and } \mathbf{Q}(s) \equiv \mathbf{P}(-\frac{5}{4} + \frac{1}{4}s).$$

A rational parametric curve can be reparametrized with the substitution $t = f(u)/g(u)$. In this case, it is possible to perform a rational-linear reparametrization which does not change the endpoints of the curve segment. If we let

$$t = \frac{a(1-u) + bu}{c(1-u) + du}$$

and want $u = 0$ when $t = 0$ and $u = 1$ when $t = 1$, then $a = 0$ and $b = d$. Since we can scale then numerator and denominator without affecting the reparametrization, set $c = 1$ and we are left with

$$t = \frac{bu}{(1-u) + bu}$$

A rational Bézier curve

$$\mathbf{X}(t) = \frac{\binom{n}{0} w_0 \mathbf{P}_0 (1-t)^n + \binom{n}{1} w_1 \mathbf{P}_1 (1-t)^{n-1} t + \ldots + \binom{n}{n} w_n \mathbf{P}_n t^n}{\binom{n}{0} w_0 (1-t)^n + \binom{n}{1} w_1 (1-t)^{n-1} t + \ldots + \binom{n}{n} w_n t^n}$$

can be raparametrized without changing its endpoints by making the substitutions

$$t = \frac{bu}{(1-u) + bu}, \quad (1-t) = \frac{(1-u)}{(1-u) + bu}.$$

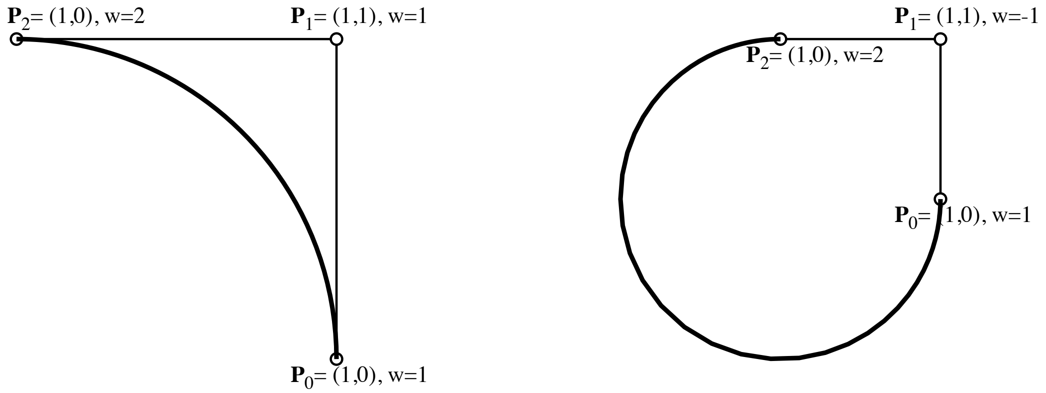
After multiplying numerator and denominator by $((1-u) \ + \ bu)^n$, we obtain

$$\mathbf{X}(u) \ = \ \frac{\binom{n}{0}(b^0 w_0)\mathbf{P}_0(1-u)^n \ + \ \binom{n}{1}(b^1 w_1)\mathbf{P}_1(1-u)^{n-1}u \ + \ ... \ + \ \binom{n}{n}(b^n w_n)\mathbf{P}_n u^n}{\binom{n}{0}b^0 w_0(1-u)^n \ + \ \binom{n}{1}b^1 w_1(1-u)^{n-1}u \ + \ ... \ + \ \binom{n}{n}b^n w_n u^n}$$

In other words, if we scale the weights w_i by b^i, the curve will not be changed!

Negative Weights

Normally, negative weights are discouraged for Bézier curves because they can cause a violation of the convex hull property. However, an interesting situation arises if we choose $b = -1$ for the rational reparametrization: the signs of the weights alternate, and the curve for $u \in [0,1]$ is the curve $t \in (\infty, 0] \cup [1, \infty)$. If the original curve is a quarter circle, the reparametrized curve will be the rest of the circle, as shown in Figure 2.22.



(a) Quarter circle.

(b) After reparametrization with $b = -1$.

Figure 2.22: Circle with negative weight.

2.14 Advantages of Rational Bézier Curves

As we have seen, rational Bézier curves have several advantages over polynomial Bézier curves. We here emphasize them.

1. Rational Bézier curves can represent conic sections exactly.
2. If you perform a perspective projection of a Bézier curve (either a polynomial or rational), the result is a rational Bézier curve.
3. Rational Bézier curves permit some additional shape control: by changing the weights, you change the shape of the curve.
4. It is possible to reparametrize the curve by simply changing the weights in a specific manner.

2.15 Explicit Bézier Curves

An explicit Bézier curve $\mathbf{P}_{[t_0,t_1]}(t)$ is one for which $x \equiv t$. For a polynomial Bézier curve, this happens when the x-coordinates of the control points are evenly spaced between t_0 and t_1. That is, $\mathbf{P}_i = (\frac{t_0(n-i)+t_1 i}{n}, y_i),\ i = 0, \ldots, n$. Such a Bézier curve takes on the form of a function

$$\begin{aligned} x &= t \\ y &= f(t) \end{aligned}$$

or simply

$$y = f(x).$$

An explicit Bézier curve is sometimes called a non-parametric Bézier curve. It is just a polynomial function expressed in the Bernstein polynomial basis. Figure 2.23 shows a degree five explicit Bézier curve over the domain $[t_0, t_1] = [0, 1]$.

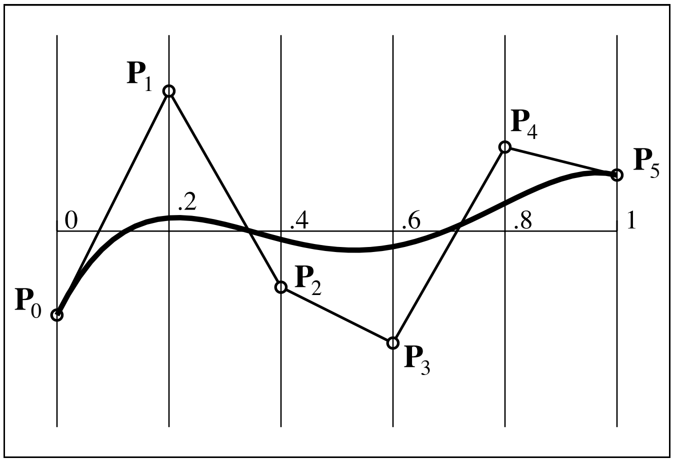


Figure 2.23: Explicit Bézier curve.

It is also possible to create an explicit rational Bézier curve. However, the representation of a degree-n rational function

$$x = t; \quad y = \frac{\sum_{i=0}^{n} w_i y_i B_i^n(t)}{\sum_{i=0}^{n} w_i B_i^n(t)}$$

as an explicit Bézier curve is degree $n + 1$ because we must have

$$x = t = t\frac{\sum_{i=0}^{n} w_i B_i^n(t)}{\sum_{i=0}^{n} w_i B_i^n(t)}$$

which is degree $n + 1$. Since $x(t)$ is degree $n + 1$, we must likewise degree elevate $y(t)$.

2.16 Integrating Bernstein polynomials

Recall that the hodograph (first derivative) of a Bézier curve is easily found by simply differencing adjacent control points (Section 2.7). It is equally simple to compute the integral of a Bernstein polynomial. Since the integral of a polynomial in Bernstein form

$$p(t) = \sum_{i=0}^{n} p_i B_i^n(t) \tag{2.16}$$

is that polynomial whose derivative is $p(t)$. If the desired integral is a degree $n+1$ polynomial in Bernstein form

$$q(t) = \sum_{i=0}^{n+1} q_i B_i^{n+1}(t), \tag{2.17}$$

we have

$$p_i = (n+1)(q_{i+1} - q_i). \tag{2.18}$$

Hence, $q_0 = 0$ and

$$q_i = \frac{\sum_{j=0}^{i-1} p_j}{n+1}, \quad i = 1, n+1. \tag{2.19}$$

Note that if $p(t)$ is expressed as an explicit Bézier curve, $q(t)$ can be interpreted as the area under $p(t)$ between the lines $x = 0$ and $x = t$. Thus, the entire area under an explicit Bézier curve can be computed as simply the average of the control points! This is so because

$$q(1) = q_{n+1} = \frac{\sum_{j=0}^{n} p_j}{n+1}. \tag{2.20}$$