CHAPTER

SEVEN

Conics and Circles

7.1 Introduction

The conic sections and circles play a fundamental role in CADCAM applications. Undoubtedly one of the greatest advantages of NURBS is their capability of precisely representing conic sections and circles, as well as free-form curves and surfaces. We assume a knowledge of conics and circles; the purpose of this chapter is to study them in the framework of their representation as NURBS curves. In Section 7.2 we review various forms and properties of conics which are required in subsequent sections. Section 7.3 covers the quadratic rational Bézier representation of conic and circular arcs; Section 7.4 introduces infinite control points. In Sections 7.5 and 7.6 we present algorithms for constructing the NURBS representation of arbitrary circles and conics, respectively, including full circles and ellipses. Section 7.7 covers conversions between the various representation forms, and Section 7.8 gives examples of higher order circle representations.

7.2 Various Forms for Representing Conics

There are many ways to define and represent the conics. We start by giving a geometric definition, and then use it to derive the general implicit (algebraic) equations of the conics in the xy plane (see [Lawr72]). A conic is the locus of a point moving so that its distance from a fixed point (the *focus*) is proportional to its distance to a fixed line (the *directrix*), that is (see Figure 7.1)

$$\text{Conic} = \left\{ \mathbf{P} \,\middle|\, \frac{\mathbf{PF}}{\mathbf{PD}} = e \right\}$$

where e is a constant of proportionality called the *eccentricity*. The eccentricity

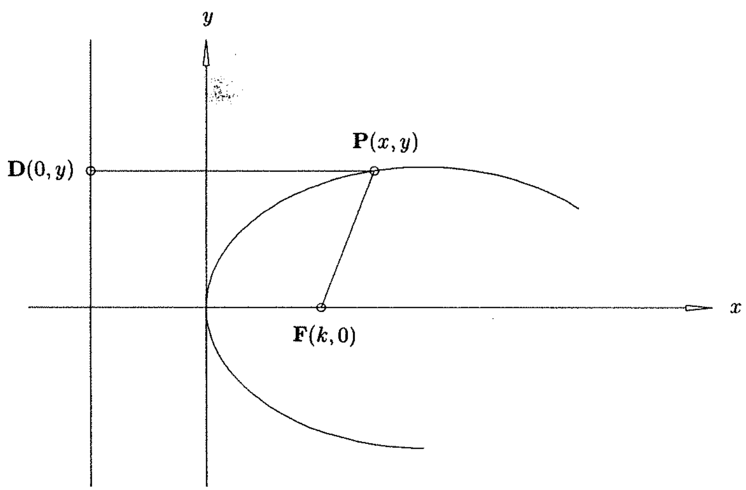


Figure 7.1. Conic definition via the focus and the directrix.

determines the type of conic, i.e.

$$e = \begin{cases} = 1 & \text{parabola} \\ < 1 & \text{ellipse} \\ > 1 & \text{hyperbola} \end{cases}$$

If the directrix is chosen to be the line $x = 0$ and the focus to be $\mathbf{F} = (k, 0)$, then

$$e = \frac{\sqrt{(x-k)^2 + y^2}}{|\, x \,|} \tag{7.1}$$

where $|\, x \,| = \mathbf{PD}$ and $\mathbf{D} = (0, y)$. Squaring and rearranging Eq. (7.1) yields

$$(1 - e^2)x^2 - 2kx + y^2 + k^2 = 0 \tag{7.2}$$

that is, a conic lying in the xy plane is represented by a second-degree algebraic equation. Conversely, it can be shown that any second-degree algebraic equation represents a conic. Let this equation be written as

$$ax^2 + by^2 + 2hxy + 2fx + 2gy + c = 0 \tag{7.3}$$

Let $\alpha = ab - h^2$, and let

$$D = \begin{vmatrix} a & h & f \\ h & b & g \\ f & g & c \end{vmatrix}$$

be the determinant of the 3×3 matrix formed from the coefficients of Eq. (7.3). Using α and D, a complete classification of the conics is

$\alpha = 0$	$D \neq 0$			parabola
	$D = 0$	$b \neq 0$	$g^2 - bc > 0$	2 parallel real lines
			$g^2 - bc = 0$	2 parallel coincident lines
			$g^2 - bc < 0$	2 parallel imaginary lines
		$b = h = 0$	$f^2 - ac > 0$	2 parallel real lines
			$f^2 - ac = 0$	2 parallel coincident lines
			$f^2 - ac < 0$	2 parallel imaginary lines
$\alpha > 0$	$D = 0$			point ellipse
	$D \neq 0$		$-bD > 0$	real ellipse
			$-bD < 0$	imaginary ellipse
$\alpha < 0$	$D \neq 0$			hyperbola
	$D = 0$			2 intersecting lines

By means of simple transformations, Eq. (7.2) can be restated in one of the standard forms. For $e \neq 1$ we use the transformation

$$x' = x - \frac{k}{1 - e^2} \qquad y' = y$$

Substituting into Eq. (7.2) and dropping the primes yields

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{7.4}$$

where
$$a = \frac{ke}{1 - e^2} \qquad b^2 = a^2(1 - e^2)$$

There are two cases:

- Ellipse (see Figure 7.2): $e < 1$ implies $a, b > 0$, and Eq. (7.4) becomes

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \qquad a > b > 0 \tag{7.5}$$

  In this position the origin is the *center* of the ellipse. The x-axis is the *major axis*, and the y-axis is the *minor axis*. The points $(-a, 0), (a, 0), (0, -b), (0, b)$ are the *vertices*. The distances a and b are the *major* and *minor radii*, respectively. If $a = b$ Eq. (7.5) represents a circle (this is the limiting case, $e \to 0$, $k \to \infty$).

- Hyperbola (Figure 7.3): $e > 1$ implies $a < 0$ and $b^2 < 0$ (b is imaginary). Setting $b = |\, b \,|$, Eq. (7.4) yields

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1 \qquad a, b > 0 \tag{7.6}$$

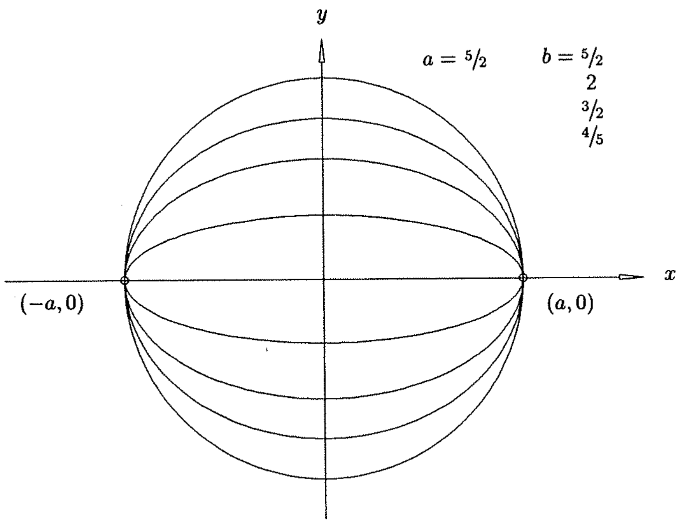


Figure 7.2. Ellipses defined by different parameters ($a = 5/2, b = \{5/2, 2, 3/2, 4/5\}$).

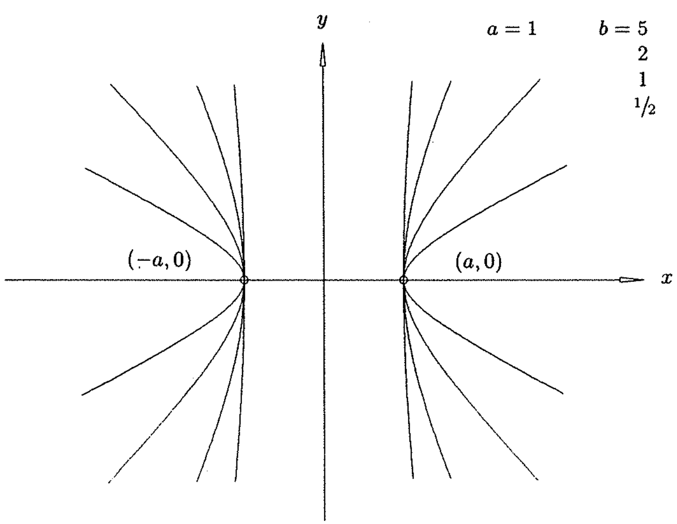


Figure 7.3. Hyperbolas defined by different parameters ($a = 1, b = \{5, 2, 1, 1/2\}$).

In this position the origin is the *center* of the hyperbola. The x-axis is the *transverse axis*, and the y-axis is called the *semiconjugate* or *imaginary axis*. The points $(-a, 0)$ and $(a, 0)$ are the *vertices*. The distances a and b are called the *major* and *minor* (or *imaginary*) *radii*, respectively. Note that the hyperbola has two branches separated by the imaginary axis.

Similarly, $e = 1$ and the transformations

$$x' = x - \frac{1}{2}k \qquad y' = y$$

yield the parabola

$$y^2 = 4ax \qquad a = \frac{1}{2}k > 0 \tag{7.7}$$

with focus $\mathbf{F} = (a, 0)$ and directrix $x + a = 0$ (see Figure 7.4). The parabola has no center. In standard position its *axis* is the x-axis, the origin is its *vertex*, and a is its *focal distance*.

Two parametric representations of the conics are important in CADCAM applications: rational and maximum inscribed area forms. We discuss the rational form first. The equations

$$x(u) = a\frac{1-u^2}{1+u^2}$$

$$y(u) = b\frac{2u}{1+u^2} \qquad -\infty < u < \infty \tag{7.8}$$

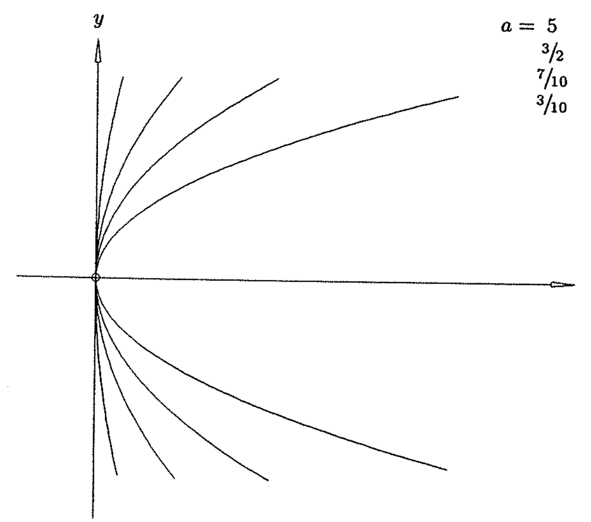


Figure 7.4. Parabolas defined by different parameters ($a = \{5, 3/2, 7/10, 3/10\}$).

represent an ellipse in standard position (Figure 7.5 shows points on the segment $0 \le u \le 1$). This can be seen by substituting Eq. (7.8) into Eq. (7.5)

$$\frac{a^2\left(\dfrac{1-u^2}{1+u^2}\right)^2}{a^2} + \frac{b^2\left(\dfrac{2u}{1+u^2}\right)^2}{b^2} = \frac{1-2u^2+u^4+4u^2}{1+2u^2+u^4} = 1$$

Note that $\big(x(0), y(0)\big) = (a, 0)$, $\big(x(1), y(1)\big) = (0, b)$, and the vertex $(-a, 0)$ is approached in the limit as $u \to -\infty$ or $u \to \infty$. The equations

$$\begin{aligned} x(u) &= a\frac{1+u^2}{1-u^2} \\ y(u) &= b\frac{2u}{1-u^2} \qquad -\infty < u < \infty \end{aligned} \tag{7.9}$$

represent a hyperbola in standard position (Figure 7.6). The interval $u \in (-1, 1)$ corresponds to the right branch; the left branch is traced out by $u \in (-\infty, -1)$ and $u \in (1, \infty)$. $\mathbf{C}(-1)$ and $\mathbf{C}(1)$ represent points at infinity. Finally

$$\begin{aligned} x(u) &= au^2 \\ y(u) &= 2au \qquad -\infty < u < \infty \end{aligned} \tag{7.10}$$

parameterizes the parabola in standard position (see Figure 7.7). Let $\mathbf{C}(u) = \big(x(u), y(u), 0\big)$ be a conic in standard position, embedded in three-dimensional Euclidean space. By applying an arbitrary 3×3 rotation matrix and an arbitrary

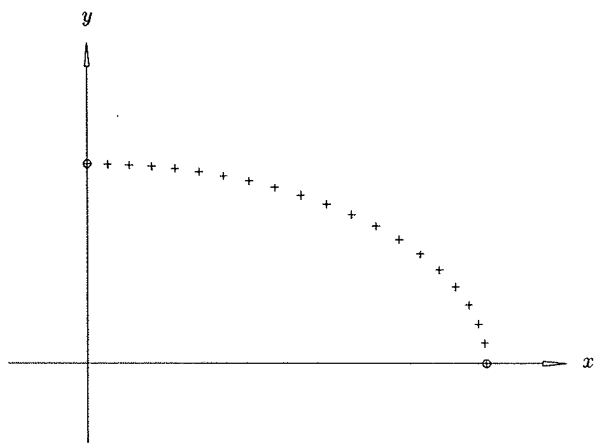


Figure 7.5. Rational parameterization of an elliptic arc (Eq. [7.8]).

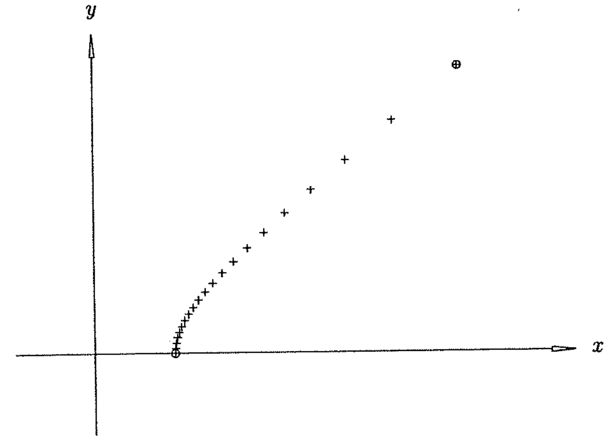


Figure 7.6. Rational parameterization of a hyperbolic arc (Eq. [7.9]).

translation vector, it is easy to see that a general conic in three-dimensional space has the form

$$x(u) = \frac{a_0 + a_1 u + a_2 u^2}{w_0 + w_1 u + w_2 u^2} = \frac{\sum_{i=0}^{2} a_i u^i}{\sum_{i=0}^{2} w_i u^i}.$$

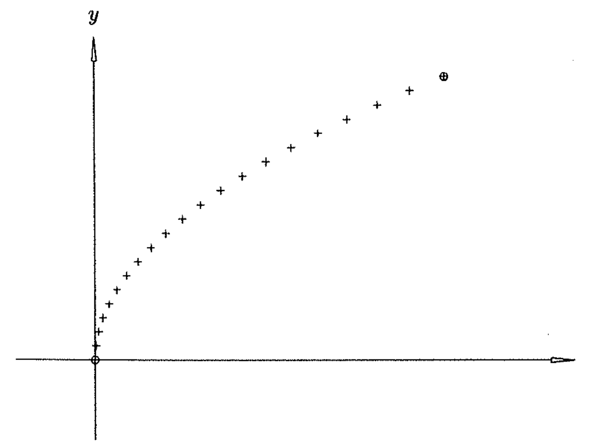


Figure 7.7. Parameterization of a parabolic arc (Eq. [7.10]).

$$y(u) = \frac{b_0 + b_1 u + b_2 u^2}{w_0 + w_1 u + w_2 u^2} = \frac{\sum_{i=0}^{2} b_i u^i}{\sum_{i=0}^{2} w_i u^i}$$

$$z(u) = \frac{c_0 + c_1 u + c_2 u^2}{w_0 + w_1 u + w_2 u^2} = \frac{\sum_{i=0}^{2} c_i u^i}{\sum_{i=0}^{2} w_i u^i} \qquad -\infty < u < \infty \tag{7.11}$$

Setting

$$x_i = \frac{a_i}{w_i}$$

$$y_i = \frac{b_i}{w_i}$$

$$z_i = \frac{c_i}{w_i}$$

and

$$\mathbf{a}_i = (x_i, y_i, z_i)$$

we obtain

$$\mathbf{C}(u) = \frac{w_0 \mathbf{a}_0 + w_1 \mathbf{a}_1 u + w_2 \mathbf{a}_2 u^2}{w_0 + w_1 u + w_2 u^2} = \frac{\sum_{i=0}^{2} w_i \mathbf{a}_i u^i}{\sum_{i=0}^{2} w_i u^i} \tag{7.12}$$

the rational power basis form of the conic. Setting

$$\mathbf{a}_i^w = (w_i x_i, w_i y_i, w_i z_i, w_i) \tag{7.13}$$

yields

$$\mathbf{C}^w(u) = \sum_{i=0}^{2} \mathbf{a}_i^w u^i \tag{7.14}$$

the homogeneous form of the rational power basis conic. Furthermore, any equation in the form of Eq. (7.11) is a conic. This fact will follow from Sections 7.3 and 7.7.

The rational forms can represent rather poor parameterizations of a conic, in the sense that evenly spaced parameter values can map into very unevenly spaced points on the curve. Figures 7.5 to 7.7 show points on sections of the conics given by Eqs. (7.8)–(7.10). The points are images of evenly spaced values of u. From Chapter 6 we know that the parameterization of a rational curve can be changed (and possibly improved) by a reparameterization with a linear rational function.

Now suppose $\mathbf{C}(u) = \big(x(u), y(u)\big)$ is a parametric representation of a conic in standard position. For each type of conic we now give functions $x(u), y(u)$, which yield a good parameterization in the sense that if for arbitrary integer n and parameter bounds a and b we compute n equally spaced parameter values

$$a = u_1, \ldots, u_n = b \qquad u_{i+1} - u_i = \text{constant}, \;\; i = 1, \ldots, n-1$$

then the point sequence $\mathbf{C}(u_1), \ldots, \mathbf{C}(u_n)$ forms the $(n-1)$-sided polygon on $\mathbf{C}(u)$, whose closure has the maximum inscribed area. The ellipse is given by

$$x(u) = a \cos u$$
$$y(u) = b \sin u \qquad 0 \le u \le 2\pi \tag{7.15}$$

(see Figure 7.8). If $a = b$ Eq. (7.15) represents a circle, and the parameterization is uniform. The hyperbola uses the hyperbolic functions

$$\sinh u = \frac{e^u - e^{-u}}{2}$$

$$\cosh u = \frac{e^u + e^{-u}}{2}$$

Its equations are (Figure 7.9)

$$x(u) = a \cosh u$$
$$y(u) = b \sinh u \qquad -\infty < u < \infty \tag{7.16}$$

Equation (7.16) traces out the right branch; the left branch is traced out by

$$x(u) = -a \cosh u$$
$$y(u) = -b \sinh u \qquad -\infty < u < \infty$$

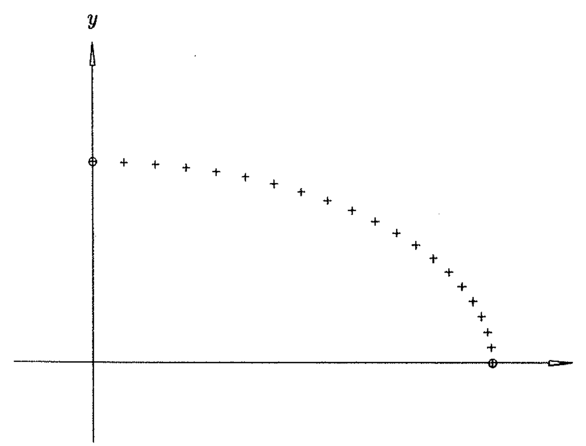


Figure 7.8. Maximum area parameterization of an ellipse (Eq. [7.15]).

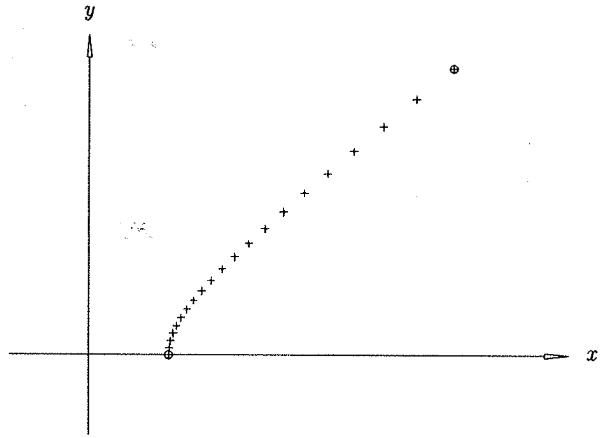


Figure 7.9. Maximum area parameterization of a hyperbola (Eq. [7.16]).

A parametric parabola is given by the equation (Figure 7.7)

$$\begin{aligned} x(u) &= au^2 \\ y(u) &= 2au \qquad -\infty < u < \infty \end{aligned} \tag{7.17}$$

which is the same as Eq. (7.10).

Equations (7.15)–(7.17) can be extended to represent conics in three-dimensional space. Let $\{\mathbf{O}, \mathbf{X}, \mathbf{Y}\}$ be the local coordinate system of the conic in three-dimensional space ($\mathbf{O}$ is a point, and $\mathbf{X}$ and $\mathbf{Y}$ are orthogonal unit length vectors). The conics are defined as (Figure 7.10)

- Ellipse: with $\mathbf{O}$ as the center, $\mathbf{X}$ as the major and $\mathbf{Y}$ as the minor axis, the equation of the ellipse is

$$\mathbf{C}(u) = \mathbf{O} + a\cos u\mathbf{X} + b\sin u\mathbf{Y} \tag{7.18}$$

- Hyperbola: with $\mathbf{O}$ as the center, $\mathbf{X}$ as the transverse axis, and $\mathbf{Y}$ as the imaginary axis, the equation of the left branch of the hyperbola is

$$\mathbf{C}(u) = \mathbf{O} - a\cosh u\mathbf{X} - b\sinh u\mathbf{Y} \tag{7.19}$$

- Parabola: with $\mathbf{O}$ as the vertex and $\mathbf{X}$ as the axis, $\mathbf{Y}$ gives the tangent direction of the parabola at its vertex. The equation of the parabola is

$$\mathbf{C}(u) = \mathbf{O} + au^2\mathbf{X} + 2au\mathbf{Y} \tag{7.20}$$

Equations (7.18)–(7.20) are the conic forms specified in the new Standard for the Exchange of Product Model Data (STEP) [STEP94].

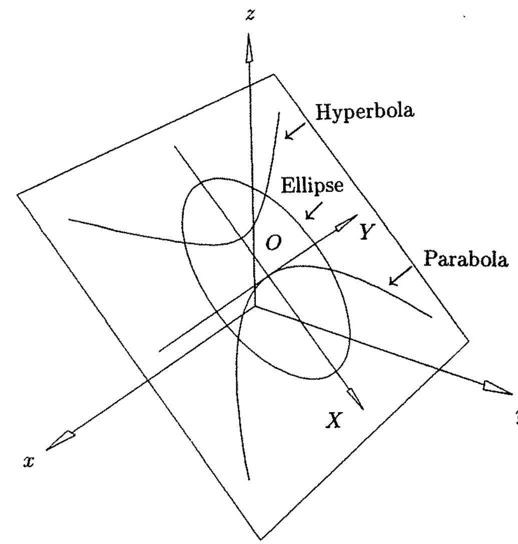


Figure 7.10. Ellipse, parabola, and hyperbola in three-dimensional space.

Almost any book on analytic geometry covers the implicit equation form of the conics in detail, including the classification of conic types, degenerate cases, and the transformations into standard position (Eqs. [7.5]–[7.7]), for example see [Salm79; Coxe80; Ilyi84]. A few modern CADCAM geometry books cover the conics in some detail, particularly the rational and maximum inscribed area parametric forms, e.g., see Rogers [Roge90] and Beach [Beac91]. The maximum inscribed area property of Eqs. (7.15)–(7.17) was first given by [Smit71]. Liming's two books ([Limi44, 79]) contain a wealth of information on conic constructions and the use of conics in engineering design.

7.3 The Quadratic Rational Bézier Arc

The quadratic rational Bézier arc has the form

$$\begin{aligned} \mathbf{C}(u) &= \frac{(1-u)^2 w_0 \mathbf{P}_0 + 2u(1-u)\,w_1\mathbf{P}_1 + u^2 w_2 \mathbf{P}_2}{(1-u)^2 w_0 + 2u(1-u)w_1 + u^2 w_2} \\ &= R_{0,2}(u)\mathbf{P}_0 + R_{1,2}(u)\mathbf{P}_1 + R_{2,2}(u)\mathbf{P}_2 \end{aligned} \tag{7.21}$$

where
$$R_{i,2}(u) = \frac{B_{i,2}(u)w_i}{\sum_{j=0}^{2} B_{j,2}(u)w_j} \qquad u \in [0,1]$$

To show that Eq. (7.21) is a conic, introduce a local, oblique coordinate system

$$\{\mathbf{P}_1, \mathbf{S}, \mathbf{T}\} \tag{7.22}$$

with

$$\mathbf{S} = \mathbf{P}_0 - \mathbf{P}_1 \qquad \mathbf{T} = \mathbf{P}_2 - \mathbf{P}_1$$

(see Figure 7.11). For arbitrary $u \in [0,1]$ the point $\mathbf{C}(u)$ lies in the triangle $\mathbf{P}_0\mathbf{P}_1\mathbf{P}_2$, hence can be written as

$$\mathbf{C}(u) = \mathbf{P}_1 + \alpha(u)\mathbf{S} + \beta(u)\mathbf{T} = \alpha(u)\mathbf{P}_0 + \big(1 - \alpha(u) - \beta(u)\big)\mathbf{P}_1 + \beta(u)\mathbf{P}_2 \tag{7.23}$$

Comparing this with Eq. (7.21) we find that

$$\alpha(u) = R_{0,2}(u) \qquad \beta(u) = R_{2,2}(u) \tag{7.24}$$

Using the identity

$$B_{0,2}(u)\,B_{2,2}(u) = \left(\frac{B_{1,2}(u)}{2}\right)^2$$

and Eq. (7.23), and denoting the denominator function in Eq. (7.21) by $w(u)$, we obtain

$$\begin{aligned}\alpha(u)\beta(u) = R_{0,2}(u)R_{2,2}(u) &= \frac{w_0 w_2\, B_{0,2}(u)B_{2,2}(u)}{\big(w(u)\big)^2} \\ &= w_0 w_2 \frac{\big(B_{1,2}(u)\big)^2}{4\big(w(u)\big)^2} = \frac{w_0 w_2}{w_1^2}\frac{\big(w_1 B_{1,2}(u)\big)^2}{4\big(w(u)\big)^2} \\ &= \frac{w_0 w_2}{w_1^2}\,\frac{1}{4}\big(R_{1,2}(u)\big)^2\end{aligned}$$

Setting

$$k = \frac{w_0 w_2}{w_1^2} \tag{7.25}$$

and using Eq. (7.23) produces

$$\alpha(u)\beta(u) = \frac{1}{4}k\big(1 - \alpha(u) - \beta(u)\big)^2 \tag{7.26}$$

Equation (7.26) is the implicit (second-degree) equation of a conic in the oblique coordinate system $\{\mathbf{P}_1, \mathbf{S}, \mathbf{T}\}$. Note that the constant, k, of Eq. (7.25) is the *conic shape factor* given in Chapter 6, Eq. (6.59). Equation (7.26) says that k determines the conic; if the three weights are changed in such a way that k is not changed, then only the parameterization changes, not the curve.

The type of conic can be determined by looking at the denominator $w(u)$ of Eq. (7.21), which can be written as

$$\begin{aligned}w(u) &= (1-u)^2 w_0 + 2u(1-u)w_1 + u^2 w_2 \\ &= (w_0 - 2w_1 + w_2)u^2 + 2(w_1 - w_0)u + w_0\end{aligned} \tag{7.27}$$

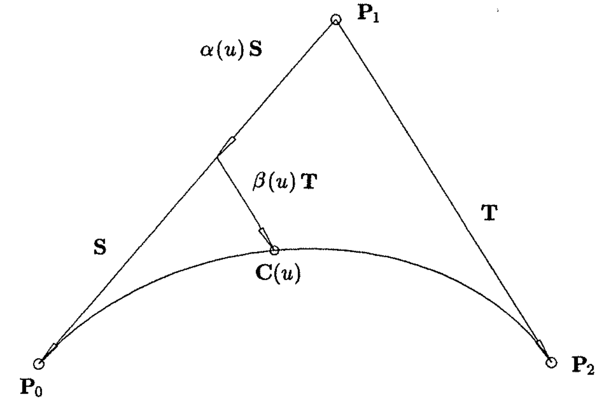


Figure 7.11. Local coordinate system for rational quadratic Bézier curve.

The roots of Eq. (7.27) are

$$u_{1,2} = \frac{w_0 - w_1 \pm w_1\sqrt{1-k}}{w_0 - 2w_1 + w_2} \tag{7.28}$$

where k is the conic shape factor. Equation (7.25) says that any two weights can be chosen arbitrarily; the conic is then determined by the third weight. It is customary to choose $w_0 = w_2 = 1$; this is called the *normal parameterization.* Then if $w_1 = 1$, Eq. (7.21) is a parabola. Assuming $w_1 \neq 1$, Eq. (7.28) implies that

- if $k > 1$, then Eq. (7.27) has no real solutions; there are no points at infinity on the curve, hence it is an ellipse;
- if $k = 1$, Eq. (7.27) has one real solution; there is one point on the curve at infinity, and the curve is a parabola;
- if $k < 1$, Eq. (7.27) has two roots; the curve has two points at infinity, and it is a hyperbola.

Expressing these conditions in terms of w_1, we have:

- ${w_1}^2 < 1$ $(-1 < w_1 < 1) \Longrightarrow$ ellipse;
- ${w_1}^2 = 1$ $(w_1 = 1$ or $-1) \Longrightarrow$ parabola;
- ${w_1}^2 > 1$ $(w_1 > 1$ or $w_1 < -1) \Longrightarrow$ hyperbola.

(see Figure 7.12).

Notice that w_1 can be zero or negative. $w_1 = 0$ yields a straight line segment from $\mathbf{P}_0$ to $\mathbf{P}_2$, and $w_1 < 0$ yields the complementary arc, traversed in the reverse order (see [Lee87] for a simple proof). Notice also that the convex hull property does not hold if $w_1 < 0$.

Varying w_1 yields a family of conic arcs having $\mathbf{P}_0$ and $\mathbf{P}_2$ as endpoints and end tangents parallel to $\mathbf{P}_0\mathbf{P}_1$ and $\mathbf{P}_1\mathbf{P}_2$. However, specifying a weight is not a convenient design tool. A more convenient way to select a conic from the family is to specify a third point on the conic, which is attained at some parameter value, say $u = 1/2$. This point is called the *shoulder point* of the conic, $\mathbf{S} = \mathbf{C}(1/2)$ (see Figure 7.12). Substitution of $u = 1/2$ into Eq. (7.21) yields

$$\mathbf{S} = \frac{1}{1+w_1}\mathbf{M} + \frac{w_1}{1+w_1}\mathbf{P}_1 \tag{7.29}$$

where $\mathbf{M}$ is the midpoint of the chord $\mathbf{P}_0\mathbf{P}_2$. Due to our choice of $w_0 = w_2 = 1$, it follows that the tangent to the conic at $\mathbf{S}$ is parallel to $\mathbf{P}_0\mathbf{P}_2$, i.e., the conic attains its maximum distance from $\mathbf{P}_0\mathbf{P}_2$ at $\mathbf{S} = \mathbf{C}(1/2)$. Let s be a new parameter that gives a linear interpolation between $\mathbf{M}$ and $\mathbf{P}_1$. Then for some value of s we have

$$\mathbf{S} = (1-s)\mathbf{M} + s\mathbf{P}_1 \tag{7.30}$$

From Eqs. (7.29) and (7.30) it follows that

$$s = \frac{w_1}{1+w_1} \qquad w_1 = \frac{s}{1-s} \tag{7.31}$$

The parameter s is a good shape design tool. The designer can move his shoulder point (which determines the fullness of the curve) linearly from $\mathbf{M}$ to $\mathbf{P}_1$. $s = 0$ yields a line segment, $0 < s < 1/2$ yields an ellipse, $s = 1/2$ yields a parabola, and $1/2 < s < 1$ yields a hyperbola.

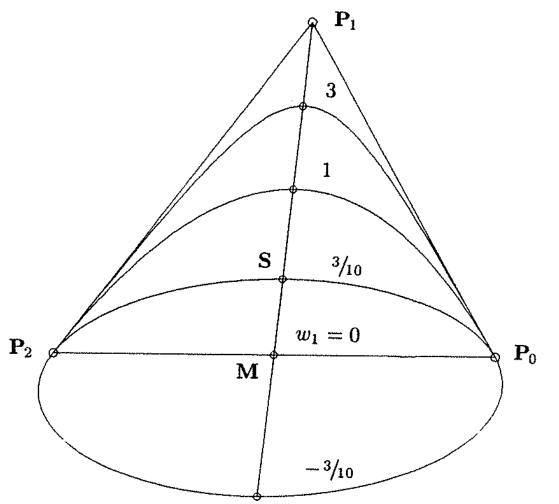


Figure 7.12. Various conic arcs defined by $w_1 = \{3, 1, 3/10, 0, -3/10\}$.

A circular arc of sweep angle less than 180° is also represented by Eq. (7.21). For symmetry $\mathbf{P}_0\mathbf{P}_1\mathbf{P}_2$ must be an isosceles triangle, with $\mathbf{P}_0\mathbf{P}_1 = \mathbf{P}_1\mathbf{P}_2$. Because the circle is a special case of an ellipse, we expect that $0 < w_1 < 1$. Consider Figure 7.13. From Eq. (7.31) it follows that

$$w_1 = \frac{s}{1-s} = \frac{\mathbf{MS}}{\mathbf{SP}_1} \tag{7.32}$$

Let $\theta = \angle \mathbf{P}_1\mathbf{P}_2\mathbf{M}$. From symmetry the arc $\mathbf{P}_2\mathbf{S}$ is the same as $\mathbf{SP}_0$, hence the angle $\angle \mathbf{SP}_2\mathbf{M}$ bisects θ. From the properties of bisectors it follows that

$$w_1 = \frac{\mathbf{MS}}{\mathbf{SP}_1} = \frac{\mathbf{MP}_2}{\mathbf{P}_1\mathbf{P}_2} = \frac{e}{f} = \cos\theta \tag{7.33}$$

Much of the material in this section goes back to the work of Coons and Forrest [Coon67; Ahuj68; Forr68]. It can also be found in [Lee87; Pieg87a].

7.4 Infinite Control Points

In this section we introduce infinite control points, a concept which we use to construct circular and elliptical arcs sweeping 180°. Versprille [Vers75] mentions infinite control points briefly, and Piegl uses them to construct a number of different curves and surfaces [Pieg87c, 88a, 88b].

The notion of a point at infinity is common in projective geometry [Ahuj68; Coxe74; Ries81]. The point $\mathbf{P}^w = (x, y, z, 0)$ in four-dimensional space (x, y, and

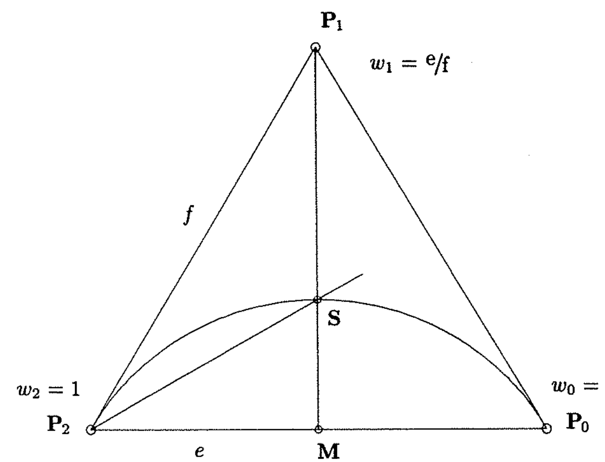


Figure 7.13. Quadratic rational Bézier representation of a circular arc sweeping less than 180°.

z not all zero) is called a *point at infinity*. It is illustrated in three-dimensional space by a direction vector from the origin through the three-dimensional point (x, y, z). Thus, $\mathbf{P} = H\{\mathbf{P}^w\}$ is a direction vector (see Eq. [1.16]). To emphasize this we use the notation $\vec{\mathbf{P}}$.

Assume $\mathbf{P}_j^w = (x_j, y_j, z_j, 0)$ is an infinite control point of a curve $\mathbf{C}(u)$. Then

$$\mathbf{C}^w(u) = \sum_{i=0}^{n} N_{i,p}(u)\mathbf{P}_i^w$$

can be written as

$$\mathbf{C}(u) = \frac{\left(\sum_{j\neq i=0}^{n} N_{i,p}(u)w_i\mathbf{P}_i\right) + N_{j,p}(u)\vec{\mathbf{P}}_j}{\sum_{j\neq i=0}^{n} N_{i,p}(u)w_i}$$

$$= \frac{\sum_{j\neq i} N_{i,p}(u)w_i\mathbf{P}_i}{w(u)} + \frac{N_{j,p}(u)\vec{\mathbf{P}}_j}{w(u)}$$

$$= \bar{\mathbf{C}}(u) + f(u)\vec{\mathbf{P}}_j \tag{7.34}$$

Hence, for fixed u_0, $\mathbf{C}(u_0)$ is $\bar{\mathbf{C}}(u_0)$, which lies in the convex hull of the control points $\mathbf{P}_0, \ldots, \mathbf{P}_{j-1}, \mathbf{P}_{j+1}, \ldots, \mathbf{P}_n$, plus a nonnegative scale factor of the vector $\vec{\mathbf{P}}_j$ (see Figure 7.14). Increasing the magnitude of $\vec{\mathbf{P}}_j$ pulls the curve toward its direction. Note in Figure 7.14 that the curve $\bar{\mathbf{C}}(u)$ touches the control polygon on the segments $\mathbf{P}_1\mathbf{P}_3$ and $\mathbf{P}_3\mathbf{P}_4$. $\mathbf{C}(u)$ is a cubic curve with two distinct internal knots (multiplicity 1). The points at which $\bar{\mathbf{C}}(u)$ touches the control polygon correspond to $\bar{\mathbf{C}}(u)$ evaluated at these internal knot values.

We warn the reader at this point that in the projective geometry sense we are not rigorously correct in stating that $\mathbf{P}^w = (x, y, z, 0)$ is a point at infinity. In projective geometry terminology, the points (x, y, z, w) and $(\alpha x, \alpha y, \alpha z, \alpha w)$, $\alpha \neq 0$, are the same, that is, a point in projective space is what mathematicians call an *equivalence class*. This means that $\vec{\mathbf{P}}_j = (x_j, y_j, z_j, 0)$ and $\vec{\mathbf{P}}_j^* = (\alpha x_j, \alpha y_j, \alpha z_j, 0)$ are two *representations* of the same point in projective space. However, substituting $\vec{\mathbf{P}}_j$ and $\vec{\mathbf{P}}_j^*$ into Eq. (7.34) clearly results in two different curves. In this book we do not delve into projective geometry, thus we choose to sacrifice mathematical rigor for clarity. Strictly speaking, our infinite control point is just a representative of a projective point, whose last coordinate happens to be zero.

Example

Ex7.1 We represent the semicircle of radius r, centered at the origin, with an infinite control point (see Figure 7.15). Let $\mathbf{P}_0^w = (r, 0, 1)$, $\mathbf{P}_1^w = (0, r, 0)$, and $\mathbf{P}_2^w = (-r, 0, 1)$. Then

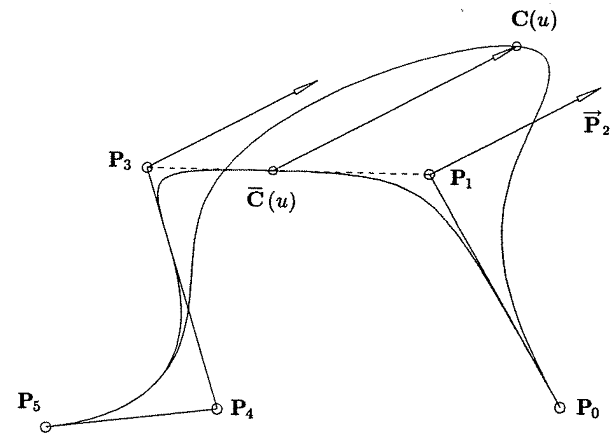


Figure 7.14. NURBS curve with an infinite control point $\vec{P}_2$; notice the stronger convex hull property.

$$\mathbf{C}(u) = \frac{(1-u)^2 w_0\mathbf{P}_0 + u^2 w_2\mathbf{P}_2}{(1-u)^2 w_0 + u^2 w_2} + \frac{2u(1-u)}{(1-u)^2 w_0 + u^2 w_2}\vec{\mathbf{P}}_1$$

$$= \frac{(1-u)^2\mathbf{P}_0 + u^2\mathbf{P}_2}{1 - 2u + 2u^2} + \frac{2u(1-u)}{1 - 2u + 2u^2}\vec{\mathbf{P}}_1$$

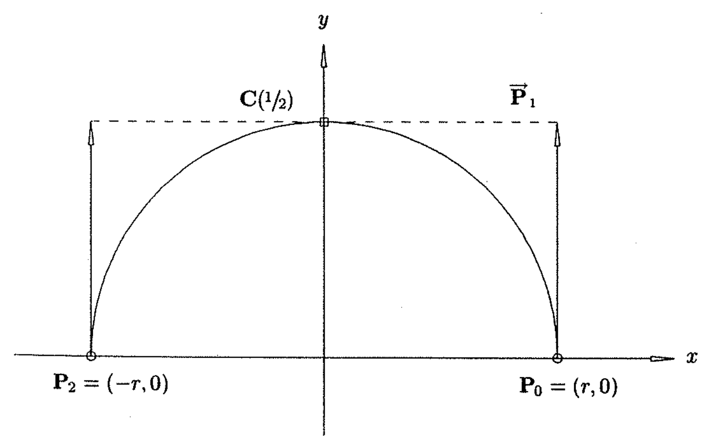


Figure 7.15. A semicircle defined by using an infinite control point.

and it follows that

$$x(u) = \frac{r(1-2u)}{1-2u+2u^2} \qquad y(u) = \frac{2ru(1-u)}{1-2u+2u^2}$$

The reader can verify that $\left(x(u)\right)^2 + \left(y(u)\right)^2 = r^2$ for all u. Notice that

$$\mathbf{C}\left(\frac{1}{2}\right) = (0, r)$$

is the point on the arc farthest from the chord $\mathbf{P}_0\mathbf{P}_2$, and the derivative at this point, $\mathbf{C}'(1/2)$, is parallel to $\mathbf{P}_0\mathbf{P}_2$, i.e., the shoulder point concept of Section 7.3 also holds when $\mathbf{P}_1^w$ is an infinite control point.

Note that there is a difference between infinite control points and zero weights. Figure 7.12 shows that $w_1 = 0$ yields a straight line from $\mathbf{P}_0$ to $\mathbf{P}_2$. Substituting $w_1 = 0$ into Eq. (7.21) and assuming the normal parameterization yields

$$\mathbf{C}(u) = R_{0,2}(u)\mathbf{P}_0 + R_{2,2}(u)\mathbf{P}_2 = \frac{(1-u)^2\mathbf{P}_0 + u^2\mathbf{P}_2}{(1-u)^2 + u^2}$$

that is, $R_{1,2}(u) \equiv 0$, and $\mathbf{C}(u)$ is a straight line from $\mathbf{P}_0$ to $\mathbf{P}_2$; see also Figures 4.2, 4.3c, 4.4, 4.5d, and 4.6. If we think of our NURBS curves in terms of homogeneous control points

$$\mathbf{C}^w(u) = \sum_{i=0}^{n} N_{i,p}(u)\mathbf{P}_i^w \tag{7.35}$$

then we represent an infinite control point by $\mathbf{P}_j^w = (x_j, y_j, z_j, 0)$, and we set w_j to zero by setting $\mathbf{P}_j^w = (0,0,0,0)$. Although one must be careful in dealing with such points individually, they generally cause no problems in B-spline algorithms derived from Eq. (7.35). For example, the point evaluation and knot insertion algorithms of previous chapters (Algorithms `A4.1` and `A5.1`) have no problem with zero weights or infinite control points (assuming, of course, not all weights are zero).

7.5 Construction of Circles

In this section we develop algorithms for constructing circular arcs of arbitrary sweep angle, including full circles. The construction of a general NURBS circular arc is more complicated than first expected, and there are many ways to do it. Much of the material in this section is taken from [Pieg89b], which is a detailed study of the NURBS circle.

We discuss only quadratic representations in this section. Higher degree representations are useful, for example, when an arc of greater than or equal to 180° is desired, without internal knots and without the use of negative weights

or infinite control points. Such representations can be obtained using degree elevation (see Section 7.8).

From Section 7.3 we know how to construct an arc of less than 180°, and clearly we can construct arcs of arbitrary sweep angle by simply piecing together smaller arcs using multiple knots. When deciding how to do this, there are four tradeoffs to consider: continuity, parameterization, convex hull, and number of control points required. The last three are related, in the sense that parameterization and characteristics of the convex hull can be improved by increasing the number of control points. We clarify these trade-offs with some examples. For simplicity, all examples are in the xy plane, centered at the origin, and have radius 1.

Examples

`Ex7.2` A full circle using a nine-point square control polygon: Example `Ex6.7` and Figure 6.12 (and Section 7.3) show the 90° arc in the first quadrant is obtained using $\{\mathbf{P}_i\} = \{(1,0),(1,1),(0,1)\}$ and $\{w_i\} = \{1, \sqrt{2}/2, 1\}$. By piecing four of these arcs together using double knots, we obtain the full circle of Figures 7.16a and 7.16b. The knots, weights, and control points are

$$U = \left\{0,0,0,\frac{1}{4},\frac{1}{4},\frac{1}{2},\frac{1}{2},\frac{3}{4},\frac{3}{4},1,1,1\right\}$$

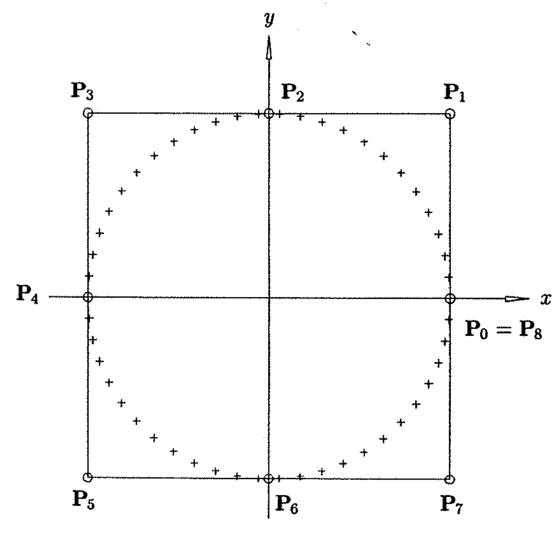
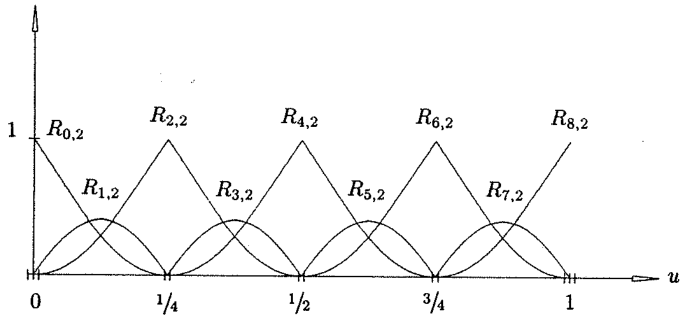


Figure 7.16. A nine-point square-based NURBS circle. (a) Control polygon and parameterization; (b) rational basis functions.



(b)

Figure 7.16. (*Continued.*)

$$\{w_i\} = \left\{1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1\right\}$$

$$\{\mathbf{P}_i\} =$$

$$\{(1,0),(1,1),(0,1),(-1,1),(-1,0),(-1,-1),(0,-1),(1,-1),(1,0)\}$$

To indicate parameterization, points are marked on the circle; they were computed at equally spaced parameter values. The circumscribing square forms a fairly tight convex hull. Now consider continuity at $u = 1/4$ ($u = 1/2$ and $u = 3/4$ are analogous). We can write either

$$\mathbf{C}^w(u) = \sum_{i=0}^{n} N_{i,2}(u)\mathbf{P}_i^w \quad \text{or} \quad \mathbf{C}(u) = \sum_{i=0}^{n} R_{i,2}(u)\mathbf{P}_i$$

Clearly, the basis functions $N_{2,2}(u)$ and $R_{2,2}(u)$ are only C^0 continuous at $u = 1/4$ (see Figure 7.16b). $\mathbf{C}^w(u)$ is also only C^0 continuous, that is, the four parabolic arcs in homogeneous space are linked together around the cone, and where each pair meet there is a cusp. Analytically, this can be seen by computing the first derivative of $\mathbf{C}^w(u)$ at $u = 1/4$ from the left and right. Applying Eq. (3.7) to just the w coordinate yields

$$w'\left(\frac{1}{4}\right)_{\text{left}} = \frac{2}{\frac{1}{4} - 0}(w_2 - w_1) = 8\left(1 - \frac{\sqrt{2}}{2}\right)$$

and
$$w'\left(\frac{1}{4}\right)_{\text{right}} = \frac{2}{\frac{1}{2} - \frac{1}{4}}(w_3 - w_2) = 8\left(\frac{\sqrt{2}}{2} - 1\right)$$

Clearly, the two derivatives are not equal. Surprisingly, $\mathbf{C}(u)$ is C^1 continuous. Using Eqs. (4.9) and (4.10) we obtain

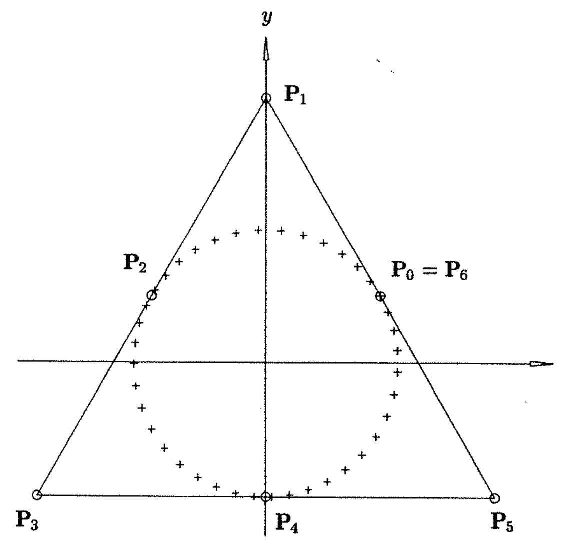
$$\mathbf{C}'\left(\frac{1}{4}\right)_{\text{left}} = \frac{2}{\frac{1}{4} - 0}\,\frac{\sqrt{2}/2}{1}\,(\mathbf{P}_2 - \mathbf{P}_1) = (-4\sqrt{2}, 0)$$

and
$$\mathbf{C}'\left(\frac{1}{4}\right)_{\text{right}} = \frac{2}{\frac{1}{2} - \frac{1}{4}}\,\frac{\sqrt{2}/2}{1}\,(\mathbf{P}_3 - \mathbf{P}_2) = (-4\sqrt{2}, 0)$$

With regard to the first three criteria, this is quite a good representation of the full circle.

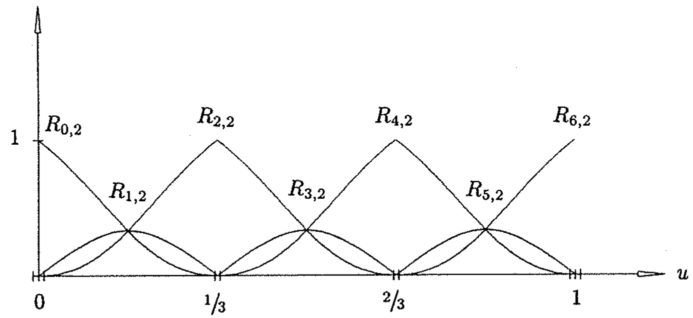
Ex7.3 A full circle using a seven-point triangular control polygon: From elementary geometry it is easy to see that an arc of 120° requires a control triangle whose base angle, $\angle \mathbf{P}_1\mathbf{P}_0\mathbf{P}_2$, is equal to 60°. From Eq. (7.33), $w_1 = \cos 60° = 1/2$. By piecing together three such arcs we obtain the full circle of Figures 7.17a and 7.17b, with

$$U = \left\{0,0,0,\frac{1}{3},\frac{1}{3},\frac{2}{3},\frac{2}{3},1,1,1\right\}$$



(a)

Figure 7.17. A seven-point triangle-based NURBS circle. (a) Control polygon and parameterization; (b) rational basis functions.



(b)

Figure 7.17. (*Continued.*)

$$\{w_i\} = \left\{1, \frac{1}{2}, 1, \frac{1}{2}, 1, \frac{1}{2}, 1\right\}$$

$$\{\mathbf{P}_i\} = \left\{\left(a, \frac{1}{2}\right), (0,2), \left(-a, \frac{1}{2}\right), (-2a,-1), (0,-1), (2a,-1), \left(a, \frac{1}{2}\right)\right\}$$

where $a = \cos 30°$. Points on the circle are marked to show parameterization, which is not quite as good as in Figure 7.16a. The convex hull is also looser. The reader can verify that $\mathbf{C}(u)$ is C^1 continuous at $u = 1/3$ and $u = 2/3$, in spite of the fact that the basis functions are C^0 continuous there (see Figure 7.17b).

Ex7.4 An arc of 240°: Setting w_1 negative yields the complementary arc. Thus, we obtain the 240° arc of Figures 7.18a and 7.18b using

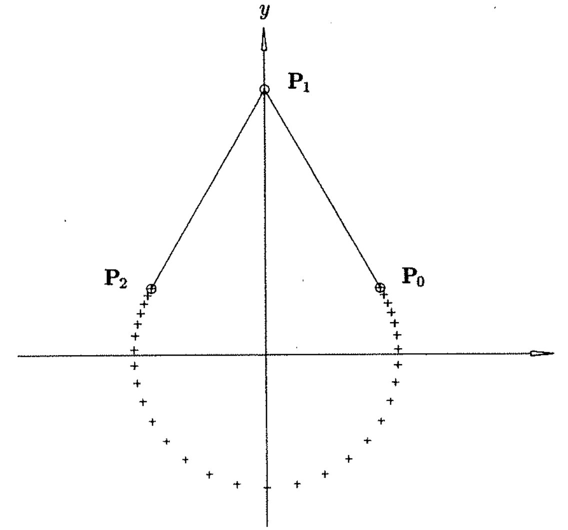
$$U = \{0,0,0,1,1,1\}$$

$$\{w_i\} = \left\{1, -\frac{1}{2}, 1\right\}$$

$$\{\mathbf{P}_i\} = \left\{\left(a, \frac{1}{2}\right), (0,2), \left(-a, \frac{1}{2}\right)\right\}$$

Negative weights are generally undesirable, because we lose the convex hull property, and dividing by zero can occur when computing points on the curve. However, it is important to note that for the circle and ellipse, $w(u) > 0$ for all $u \in [0,1]$. This follows from $w_0 = w_2 = 1$, $| w_1 | < 1$, and $w(u) = (1-u)^2 + 2u(1-u)w_1 + u^2$ (geometrically, the parabolic arc $\mathbf{C}^w(u)$ lies on the cone above the $W = 0$ plane). Thus, no problems arise when using Algorithm A4.1 to evaluate points on this arc. Let us insert a knot at $u = 1/2$. Using Eq. (5.11)

$$\alpha_1 = \frac{\frac{1}{2} - 0}{1 - 0} = \frac{1}{2} \qquad \alpha_2 = \frac{\frac{1}{2} - 0}{1 - 0} = \frac{1}{2}$$



(a)

Figure 7.18. Arc of 240° using a negative weight. (a) Control polygon and parameterization; (b) rational basis functions.

The new control points, $\mathbf{Q}_i^w$, are

$$\mathbf{Q}_0^w = \mathbf{P}_0^w \qquad \mathbf{Q}_3^w = \mathbf{P}_2^w$$

$$\mathbf{Q}_1^w = \frac{1}{2}\mathbf{P}_1^w + \frac{1}{2}\mathbf{P}_0^w = \left(\frac{a}{2}, -\frac{1}{4}, \frac{1}{4}\right)$$

$$\mathbf{Q}_2^w = \frac{1}{2}\mathbf{P}_2^w + \frac{1}{2}\mathbf{P}_1^w = \left(-\frac{a}{2}, -\frac{1}{4}, \frac{1}{4}\right)$$

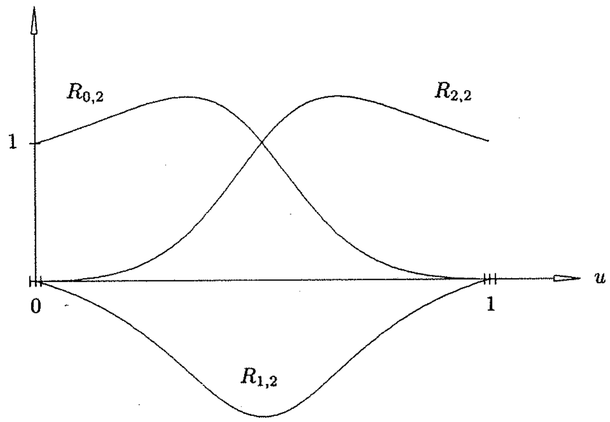
Thus (see Figures 7.19a and 7.19b)

$$U = \left\{0,0,0,\frac{1}{2},1,1,1\right\}$$

$$\{w_i\} = \left\{1, \frac{1}{4}, \frac{1}{4}, 1\right\}$$

$$\{\mathbf{Q}_i\} = \left\{\left(a, \frac{1}{2}\right), (2a,-1), (-2a,-1), \left(-a, \frac{1}{2}\right)\right\}$$

is a representation of the 240° arc without negative weights. The parameterization is not particularly good. $\mathbf{C}^w(u)$ and $\mathbf{C}(u)$ are C^2 continuous on $u \in [0,1]$; inserting the knot did not change this.



(b)

Figure 7.18. (*Continued.*)

Ex7.5 A semicircle using four control points: Consider the semicircle of Example Ex7.1 and Figure 7.15, with $r = 1$. Inserting the knot $u = 1/2$ one time, we obtain (as in Example Ex7.4)

$$\alpha_1 = \alpha_2 = \frac{1}{2}$$

$$\mathbf{Q}_0^w = \mathbf{P}_0^w \qquad \mathbf{Q}_3^w = \mathbf{P}_2^w$$

$$\mathbf{Q}_1^w = \frac{1}{2}\mathbf{P}_1^w + \frac{1}{2}\mathbf{P}_0^w = \left(-\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$$

$$\mathbf{Q}_2^w = \frac{1}{2}\mathbf{P}_2^w + \frac{1}{2}\mathbf{P}_1^w = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$$

Thus (see Figures 7.20a and 7.20b)

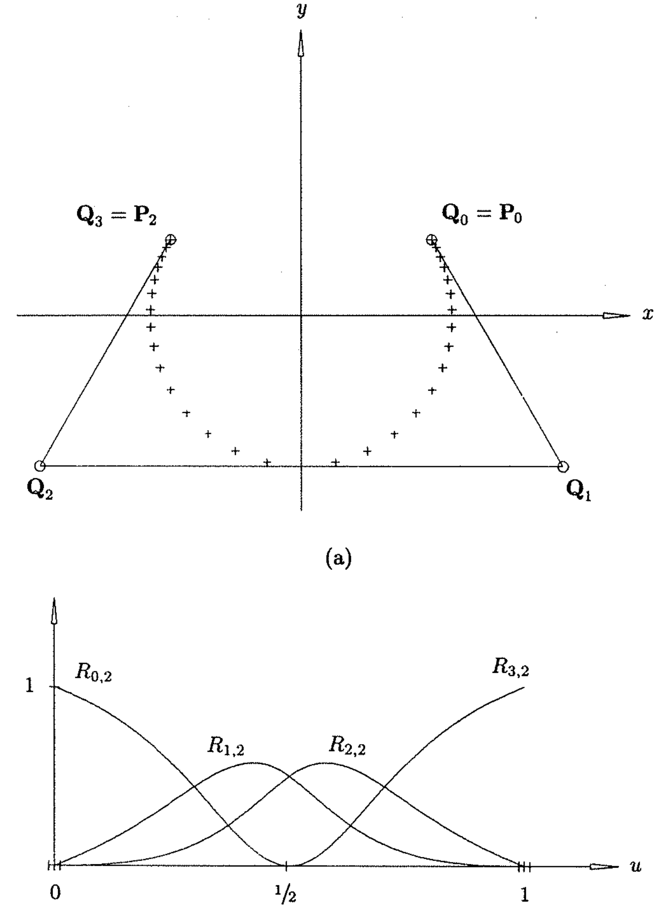
$$U = \left\{0,0,0,\frac{1}{2},1,1,1\right\}$$

$$\{w_i\} = \left\{1,\frac{1}{2},\frac{1}{2},1\right\}$$

$$\{\mathbf{Q}_i\} = \{(1,0),(1,1),(-1,1),(-1,0)\}$$

Ex7.6 A full circle using a seven-point square control polygon: Two semicircles such as in Example Ex7.5 can be pieced together to form a seven-point square circle (see Figures 7.21a and 7.21b)

$$U = \left\{0,0,0,\frac{1}{4},\frac{1}{2},\frac{1}{2},\frac{3}{4},1,1,1\right\}$$



(a)

(b)

Figure 7.19. Arc of Figure 7.18 after inserting $u = 1/2$ one time. (a) Control polygon and parameterization; (b) rational basis functions.

$$\{w_i\} = \left\{1,\frac{1}{2},\frac{1}{2},1,\frac{1}{2},\frac{1}{2},1\right\}$$

$\mathbf{C}(u)$ is C^1 continuous at $u = 1/2$ and C^2 continuous everywhere else. The parameterization is not as good as that of Example Ex7.2 and Figure 7.16a.

Algorithm A7.1 constructs a NURBS circular arc in three-dimensional space of arbitrary sweep angle θ ($0° < \theta \leq 360°$). It pieces together equal arcs of sweep

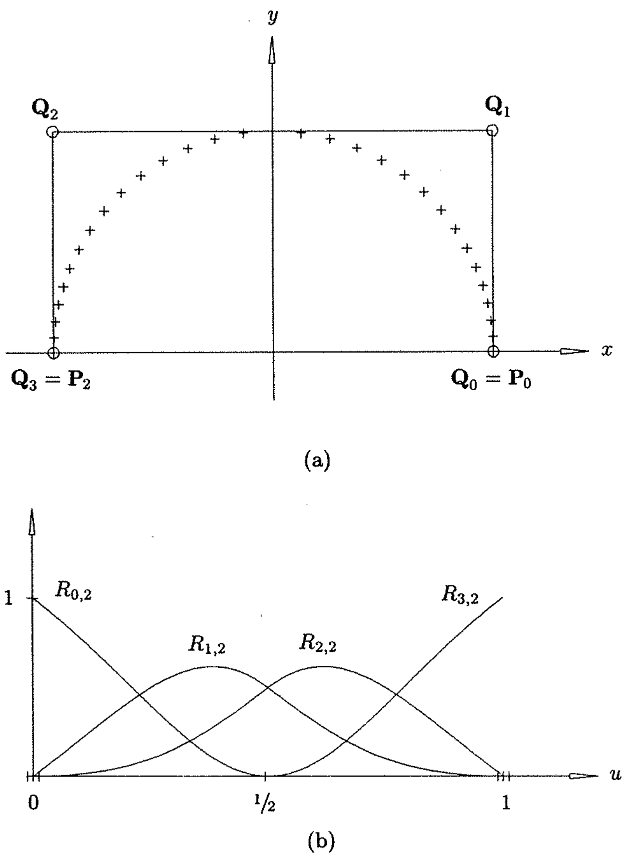


Figure 7.20. Semicircle of Figure 7.15 after inserting $u = 1/2$ one time. (a) Control polygon and parameterization; (b) rational basis functions.

angle $d\theta$, $\min(\theta, 45°) < d\theta \leq 90°$, using double knots, and weights computed by Eq. (7.33). The resulting arc of angle θ is C^1 continuous, has a tight convex hull, and has a good parameterization.

Designers are offered many different interfaces to specify circular arcs, but no matter what the interface is, the following data can be easily generated:

- $\mathbf{O}$: center of circle (origin of local coordinate system);
- $\mathbf{X}$: unit length vector lying in the plane of definition of the circle;
- $\mathbf{Y}$: unit length vector in the plane of definition of the circle, and orthogonal to $\mathbf{X}$;
- r : radius.
- θ_s, θ_e : start and end angles, measured with respect to $\mathbf{X}$.

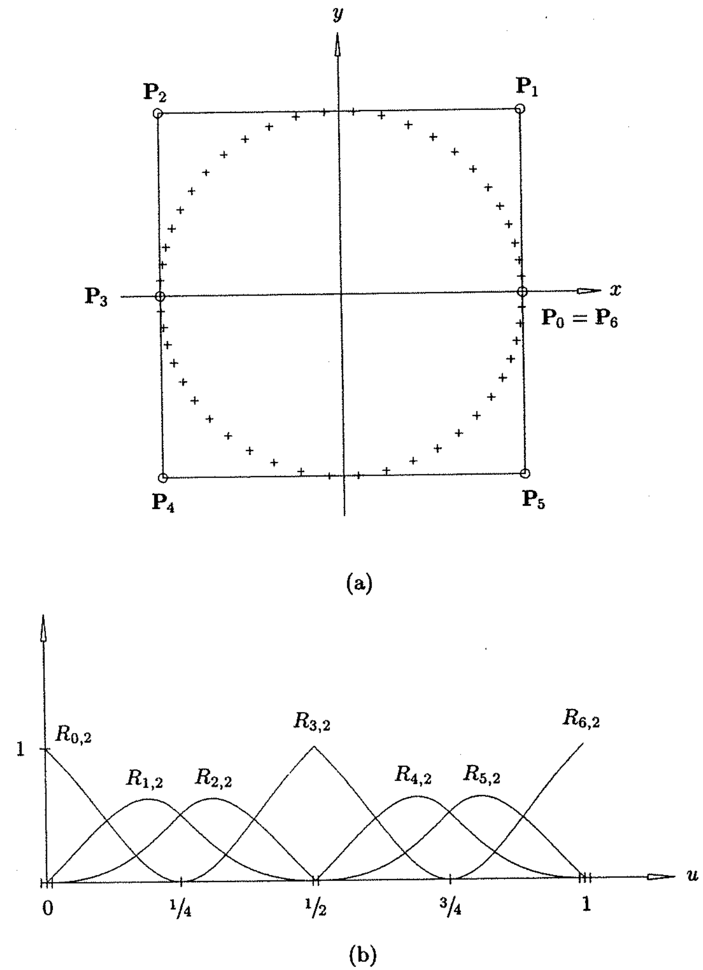


Figure 7.21. A seven-point square-based NURBS circle. (a) Control polygon and parameterization; (b) rational basis functions.

The arc is then represented by

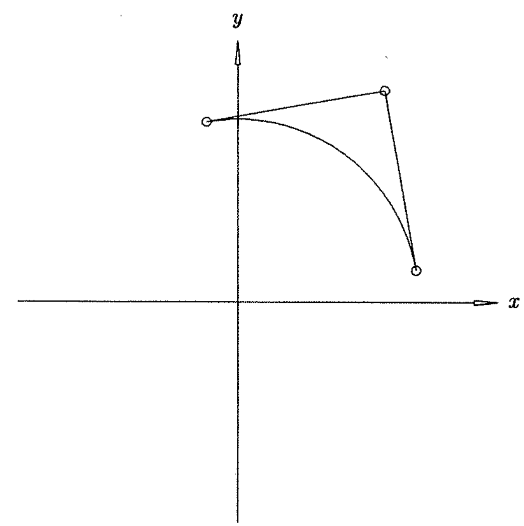
$$\mathbf{C}(u) = \mathbf{O} + r\cos u\,\mathbf{X} + r\sin u\,\mathbf{Y} \qquad \theta_s \leq u \leq \theta_e \tag{7.36}$$

The resulting arc is oriented counter-clockwise in the local coordinate system defined by $\mathbf{O}$, $\mathbf{X}$, and $\mathbf{Y}$. We assume a utility routine `Intersect3DLines(P0,T0,P2,T2,alf0,alf2,P1)`, which computes the intersection ($\mathbf{P}_1$) of two lines in three-dimensional space given by the point/tangent pairs $[\mathbf{P}_0, \mathbf{T}_0]$ and $[\mathbf{P}_2, \mathbf{T}_2]$. It returns the integer value 1 if the lines are parallel, 0 otherwise. α_0 and α_2 are the parametric locations of $\mathbf{P}_1$ along the lines $[\mathbf{P}_0, \mathbf{T}_0]$ and $[\mathbf{P}_2, \mathbf{T}_2]$, respectively. α_0, α_2, and the integer return value are not used in Algorithm `A7.1`, but they are used in Section 7.6. `Pw=w*P` denotes multiplication of a Euclidean point by a weight (`Pw=P` if $w = 1$). The algorithm computes the knots and weighted control points for the NURBS circular arc.

```
ALGORITHM A7.1
  MakeNurbsCircle(O,X,Y,r,ths,the,n,U,Pw)
    { /*  Create arbitrary NURBS circular arc  */
      /*  Input:  O,X,Y,r,ths,the  */
      /*  Output: n,U,Pw  */
    if (the < ths)    the = 360.0 + the;
    theta = the-ths;
    if (theta <= 90.0)   narcs = 1;   /* get number of arcs */
      else
        if (theta <= 180.0)   narcs = 2;
          else
            if (theta <= 270.0)   narcs = 3;
              else
                narcs = 4;
    dtheta = theta/narcs;
    n = 2*narcs;    /* n+1 control points */
    w1 = cos(dtheta/2.0);   /* dtheta/2 is base angle */
    P0 = O + r*cos(ths)*X + r*sin(ths)*Y;
    T0 = -sin(ths)*X + cos(ths)*Y; /* Initialize start values */
    Pw[0] = P0;
    index = 0;   angle = ths;
    for (i=1; i<=narcs; i++)   /* create narcs segments */
      {
      angle = angle + dtheta;
      P2 = O + r*cos(angle)*X + r*sin(angle)*Y;
      Pw[index+2] = P2;
      T2 = -sin(angle)*X + cos(angle)*Y;
      Intersect3DLines(P0,T0,P2,T2,dummy,dummy,P1);
      Pw[index+1] = w1*P1;
      index = index + 2;
      if (i < narcs)   {  P0 = P2;    T0 = T2;  }
      }
    j = 2*narcs+1;   /* load the knot vector */
    for (i=0; i<3; i++)
      {  U[i] = 0.0;   U[i+j] = 1.0;  }
    switch (narcs)
      {
      case 1:  break;
      case 2:  U[3] = U[4] = 0.5;
               break;
      case 3:  U[3] = U[4] = 1.0/3.0;
               U[5] = U[6] = 2.0/3.0;
               break;
      case 4:  U[3] = U[4] = 0.25;
               U[5] = U[6] = 0.5;
               U[7] = U[8] = 0.75;
               break;
      }
    }
```

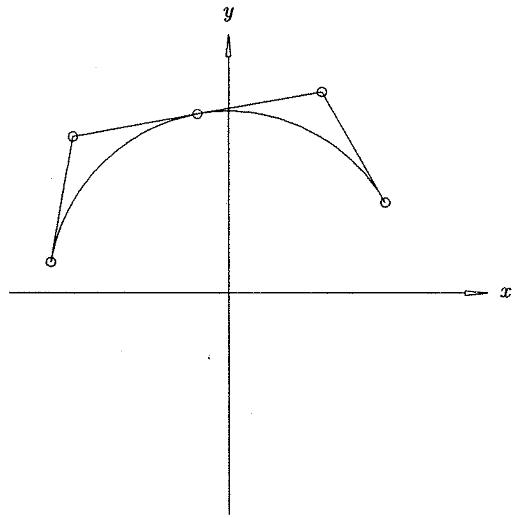

Figures 7.22a–7.22d show circular arc representations using one, two, three, and four segments, respectively.

We close this section with some miscellaneous remarks on NURBS circles. A different algorithm is given by Piegl and Tiller [Pieg89b] for constructing a circular arc of arbitrary sweep angle. It is based on using infinite control points and negative weights, and then using knot insertion to remove them (as in Examples `Ex7.4`–`Ex7.6`). The resulting arcs may require fewer control points and double knots than with Algorithm `A7.1`, but the parameterization is generally not as good. It is also proven that it is impossible to represent a full circle using a quadratic NURBS with positive weights without a double internal knot [Pieg89b].



(a)

Figure 7.22. Circular arcs of different sweep angles. (a) $\theta_s = 10°$, $\theta_e = 100°$; (b) $\theta_s = 30°$, $\theta_e = 170°$; (c) $\theta_s = 20°$, $\theta_e = 250°$; (d) $\theta_s = 40°$, $\theta_e = 330°$.



(b)

Figure 7.22. (*Continued.*)

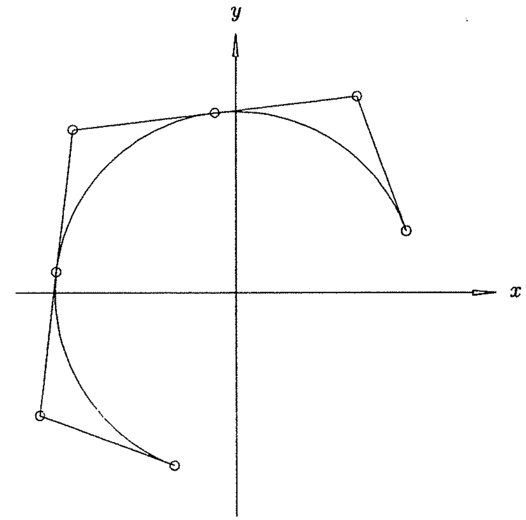
7.6 Construction of Conics

In this section we develop algorithms for constructing conics, discussing only quadratic representations. Parabolic and hyperbolic arcs can always be represented with one rational Bézier curve (no internal knots) and positive weights. As was the case for circles, we may have to piece segments together with internal knots to obtain arbitrary elliptical arcs using only positive weights. In doing this, the issues again are continuity, parameterization, convex hull, and number of control points.

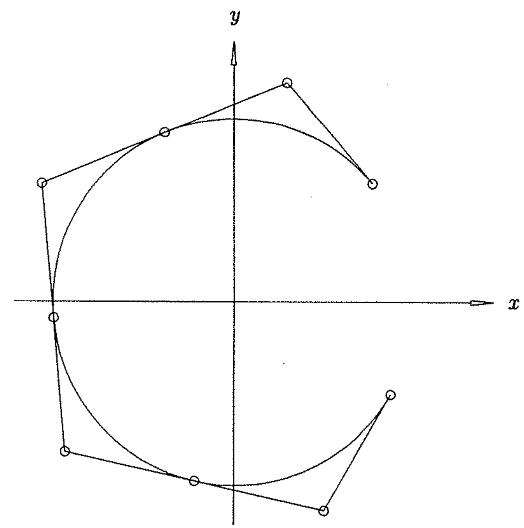
We consider first an arbitrary open conic arc in three-dimensional space. There are many ways to specify a conic arc. In CADCAM applications, two of the most common are:

- the defining geometric parameters such as radii, axes, focal distance, etc., together with specification of start and end points; this leads effectively to representation by Eqs. (7.15)–(7.17);
- specification of start and end points, $\mathbf{P}_0$ and $\mathbf{P}_2$, together with the tangent directions at those two points, $\mathbf{T}_0$ and $\mathbf{T}_2$, plus one additional point on the arc $\mathbf{P}$ (see Figure 7.23).

If the data is available in the form given in item 1, it is easy to derive $\mathbf{P}_0$, $\mathbf{T}_0$, $\mathbf{P}_2$, $\mathbf{T}_2$, and $\mathbf{P}$; thus we assume throughout this section that the data is given as specified in item 2.



(c)



(d)

Figure 7.22. (*Continued.*)

$\mathbf{P}_1$ (Figure 7.23) is easily obtained by intersecting lines $[\mathbf{P}_0, \mathbf{T}_0]$ and $[\mathbf{P}_2, \mathbf{T}_2]$. Setting $w_0 = w_2 = 1$, the only missing item is w_1. Allowing w_1 to be negative, or $\mathbf{P}_1$ to be infinite (temporarily), we can obtain any conic arc with one rational Bézier curve. If necessary, we then split this Bézier arc at suitable locations to obtain positive weights and multiple segments with good parameterization and convex hull characteristics.

The additional point $\mathbf{P}$ determines the conic and hence w_1. Substituting $\mathbf{P} = \mathbf{C}(u)$ into the left side of Eq. (7.21) yields three equations in the two unknowns, u and w_1 (which can be solved). But geometric arguments yield a more efficient algorithm (see Piegl [Pieg87b]). First assume that $\mathbf{P}_1$ is finite. The conic we seek can be considered as a perspective view of the parabola determined by $\mathbf{P}_0$, $\mathbf{P}_1$, and $\mathbf{P}_2$, with $\mathbf{P}_1$ being the center of the perspective. Any pair of conic segments lying in this triangle can be mapped onto one another, including the line $[\mathbf{P}_0, \mathbf{P}_2]$ (degenerate conic) onto the conic we seek. Hence, $\mathbf{P}$ and $\mathbf{Q}$ (Figure 7.23) are corresponding points under this transformation. Now the line $\mathbf{L}(u) = [\mathbf{P}_0, \mathbf{P}_2]$ is obtained by setting $w_1 = 0$ (not using an infinite control point), yielding

$$\mathbf{L}(u) = \frac{(1-u)^2\mathbf{P}_0 + u^2\mathbf{P}_2}{(1-u)^2 + u^2} \tag{7.37}$$

$\mathbf{L}(u)$ is a convex combination of $\mathbf{P}_0$ and $\mathbf{P}_2$, thus the ratio of distances $|\,\mathbf{P}_0\mathbf{Q}\,|$ to $|\,\mathbf{Q}\mathbf{P}_2\,|$ is $u^2 : (1-u)^2$, from which it follows that

$$u = \frac{a}{1+a} \qquad a = \sqrt{\frac{|\,\mathbf{P}_0\mathbf{Q}\,|}{|\,\mathbf{Q}\mathbf{P}_2\,|}} \tag{7.38}$$

(Compare Figure 7.23 and this argument with Figure 4.6 and Eq. [4.4]). Substituting u and $\mathbf{P}$ into Eq. (7.21), we easily obtain w_1, i.e.

$$w_1 = \frac{(1-u)^2(\mathbf{P}-\mathbf{P}_0)\cdot(\mathbf{P}_1-\mathbf{P}) + u^2(\mathbf{P}-\mathbf{P}_2)\cdot(\mathbf{P}_1-\mathbf{P})}{2u(1-u)\,|\,\mathbf{P}_1-\mathbf{P}\,|^2} \tag{7.39}$$

Piegl [Pieg87b] derives w_1 using techniques from projective geometry.

Now suppose $\mathbf{P}_1$ ($\vec{\mathbf{P}}_1$) is infinite ($\mathbf{T}_0$ and $\mathbf{T}_2$ are parallel). In this case, $w_1 = 0$ and $\vec{\mathbf{P}}_1$ is parallel to $\mathbf{T}_0$. Only the magnitude of $\vec{\mathbf{P}}_1$ is unknown. Consider Figure 7.24. The perspective center is now at infinity, and u is obtained exactly as before: $[\mathbf{P}, \mathbf{T}_0]$ (which is $[\mathbf{P}, \vec{\mathbf{P}}_1]$) is intersected with $[\mathbf{P}_0, \mathbf{P}_2]$ to yield $\mathbf{Q}$, and u is then computed from Eq. (7.38). For a rational quadratic Bézier curve with $w_0 = w_2 = 1$, Eq. (7.34) reduces to

$$\mathbf{C}(u) = \frac{(1-u)^2\mathbf{P}_0 + u^2\mathbf{P}_2}{(1-u)^2+u^2} + \frac{2u(1-u)}{(1-u)^2+u^2}\vec{\mathbf{P}}_1 = \bar{\mathbf{C}}(u) + f(u)\vec{\mathbf{P}}_1 \tag{7.40}$$

Let u_0 be the parameter yielding $\mathbf{Q}$. Then

$$\mathbf{P} = \mathbf{Q} + f(u_0)\vec{\mathbf{P}}_1$$

and

$$\vec{\mathbf{P}}_1 = \frac{1}{f(u_0)}(\mathbf{P}-\mathbf{Q}) \qquad f(u_0) = \frac{2u_0(1-u_0)}{(1-u_0)^2+u_0^2} \tag{7.41}$$

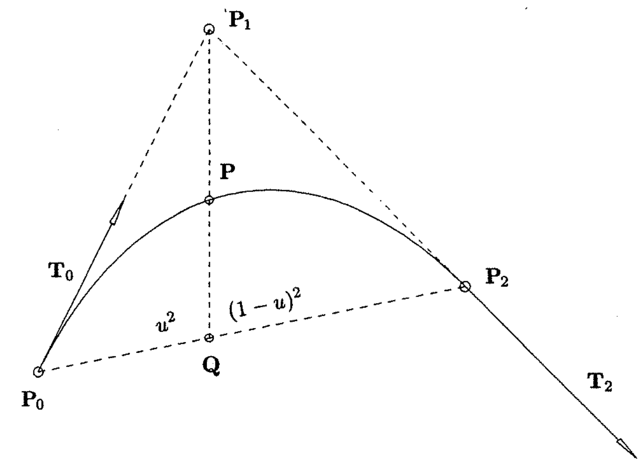


Figure 7.23. A general conic segment defined by endpoints, end tangents, and one additional point lying on the curve.

Based on Eqs. (7.38), (7.39), and (7.41), we now give an algorithm to construct one rational Bézier conic arc. Since w_1 can be negative or $\mathbf{P}_1$ infinite, this algorithm handles any conic arc except a full ellipse. We make use of the

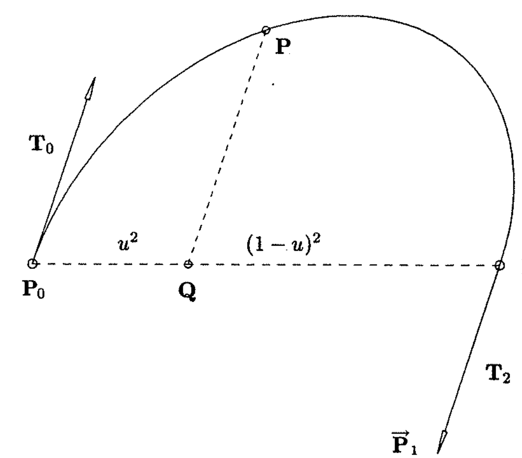


Figure 7.24. A 180° elliptical arc defined by endpoints, end tangents, and one additional point.

α_0,α_2 values computed in `Intersect3DLines`. We further assume a function `Dot(V1,V2)` which returns the dot product of the vectors $\mathbf{V}_1$ and $\mathbf{V}_2$.

```
ALGORITHM A7.2
  MakeOneArc(P0,T0,P2,T2,P,P1,w1)
    { /*  Create one Bézier conic arc  */
      /*  Input:  P0,T0,P2,T2,P  */
      /*  Output: P1,w1  */
    V02 = P2-P0;
    i = Intersect3DLines(P0,T0,P2,T2,dummy,dummy,P1);
    if (i == 0)
      { /* finite control point */
      V1P = P-P1;
      Intersect3DLines(P1,V1P,P0,V02,alf0,alf2,dummy);
      a = sqrt(alf2/(1.0-alf2));
      u = a/(1.0+a);
      num = (1.0-u)*(1.0-u)*Dot(P-P0,P1-P) + u*u*Dot(P-P2,P1-P);
      den = 2.0*u*(1.0-u)*Dot(P1-P,P1-P);
      w1 = num/den;
      return;
      }
      else
      { /* infinite control point, 180 degree arc */
      w1 = 0.0;
      Intersect3DLines(P,T0,P0,V02,alf0,alf2,dummy);
      a = sqrt(alf2/(1.0-alf2));
      u = a/(1.0+a);
      b = 2.0*u*(1.0-u);
      b = -alf0*(1.0-b)/b;
      P1 = b*T0;
      return;
      }
    }
```

Algorithm `A7.2` is adequate for parabolic and hyperbolic arcs, and for elliptical arcs for which $w_1 > 0$ and whose sweep angle is not too large. Splitting an ellipse into segments is not as easy as was the case for circles. We do not know the major and minor axes and radii, and an equation of the form of Eq. (7.36) is not available with our input. A convenient point at which to split is the shoulder point, $\mathbf{S}$. To split the arc $\mathbf{P}_0\mathbf{P}_1\mathbf{P}_2$ we make use of the rational deCasteljau algorithm, Eq. (1.19). There are two steps in splitting the ellipse (see Figures 7.25a–7.25c):

1. Split at $u = 1/2$. Using the deCasteljau Algorithm to obtain

$$\mathbf{Q}_1^w = \frac{1}{2}\mathbf{P}_0^w + \frac{1}{2}\mathbf{P}_1^w \qquad \mathbf{R}_1^w = \frac{1}{2}\mathbf{P}_1^w + \frac{1}{2}\mathbf{P}_2^w$$

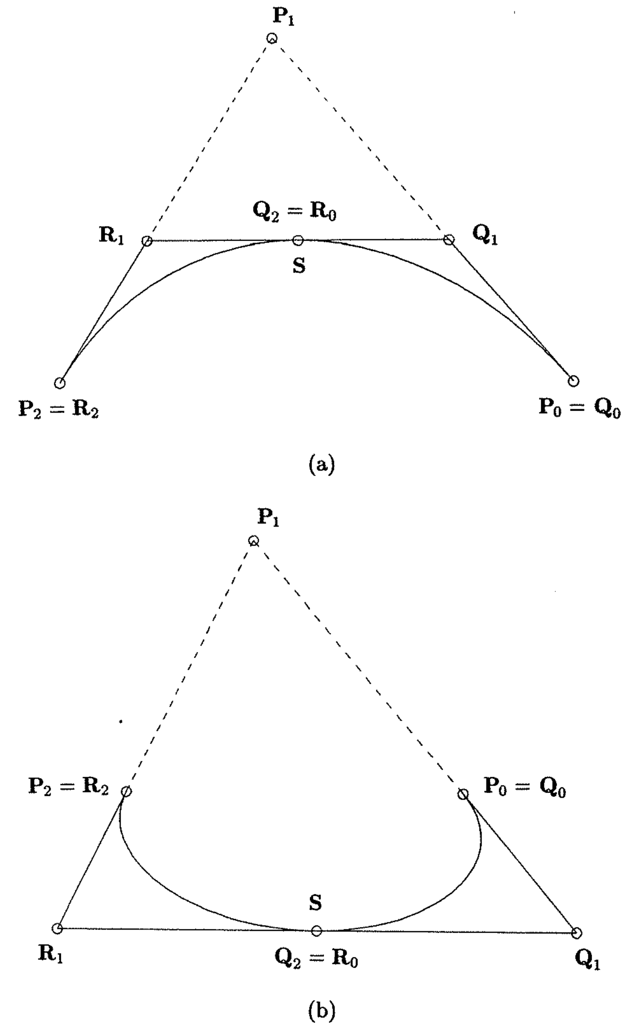


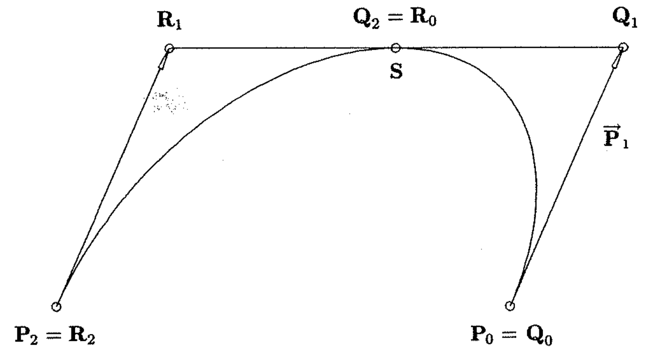
Figure 7.25. Splitting a conic curve. (a) Arc sweeping less than 180°; (b) arc sweeping more than 180°; (c) arc sweeping 180°.

and recalling that $w_0 = w_2 = 1$, it follows that

$$\mathbf{Q}_1 = \frac{\mathbf{P}_0 + w_1\mathbf{P}_1}{1+w_1} \qquad \mathbf{R}_1 = \frac{w_1\mathbf{P}_1 + \mathbf{P}_2}{1+w_1} \tag{7.42}$$

and

$$w_q = w_r = \frac{1}{2}(1+w_1) \tag{7.43}$$



(c)

Figure 7.25. (*Continued.*)

where w_q and w_r are the weights at $\mathbf{Q}_1$ and $\mathbf{R}_1$, respectively. A second application of the deCasteljau Algorithm yields

$$\mathbf{R}_0 = \mathbf{Q}_2 = \mathbf{S} = \frac{1}{2}(\mathbf{Q}_1 + \mathbf{R}_1) \tag{7.44}$$

and

$$w_s = \frac{1}{2}(1 + w_1) \tag{7.45}$$

2. Reparameterize so that the end weights are 1 for both of the two new segments. After splitting, the weights for the first segment are

$$w_0 = 1 \quad w_q = \frac{1}{2}(1 + w_1) \quad w_s = \frac{1}{2}(1 + w_1)$$

We want the weights to be

$$w_0 = 1 \quad w_{q_1} \quad w_{q_2} = 1$$

where w_{q_1} is to be determined. Using the conic shape factor (Eq. [6.59])

$$\frac{w_0 w_s}{w_q^2} = \frac{w_0 w_{q_2}}{w_{q_1}^2}$$

which implies that

$$w_{q_1} = \sqrt{\frac{1 + w_1}{2}} \tag{7.46}$$

Due to symmetry, we also have

$$w_{r_1} = \sqrt{\frac{1 + w_1}{2}} \tag{7.47}$$

Figures 7.25a and 7.25b illustrate ellipse splitting with positive and negative weights. The preceding process also works for infinite control points (see Figure 7.25c); however, the resulting formulas

$$\mathbf{Q}_1 = \mathbf{P}_0 + \vec{\mathbf{P}}_1 \quad \mathbf{R}_1 = \mathbf{P}_2 + \vec{\mathbf{P}}_1 \tag{7.48}$$

$$\mathbf{R}_0 = \mathbf{Q}_2 = \mathbf{S} = \frac{1}{2}(\mathbf{Q}_1 + \mathbf{R}_1) \tag{7.49}$$

$$w_{q_1} = w_{r_1} = \frac{\sqrt{2}}{2} \tag{7.50}$$

are even simpler. Equation (7.50) should not be surprising, as the semiellipse of Figure 7.25c is obtained by applying an affine transformation (specifically, a nonuniform scaling) to a semicircle. Such transformations do not change the weights. We leave it as an exercise for the reader to write a routine `SplitArc(P0,P1,w1,P2,Q1,S,R1,wqr)` which implements Eqs. (7.42)–(7.50).

Making use of Algorithm `A7.2` and `SplitArc()`, we now present an algorithm which constructs an arbitrary open conic arc in three-dimensional space. The resulting NURBS curve consists of either one, two, or four segments connected with C^1 continuity. The output is the knots (`U`), the number of control points less 1 (`n`), and the control points in homogeneous form (`Pw`). We assume a utility, `Angle(P,Q,R)`, which returns the angle $\angle\mathbf{PQR}$.

```
ALGORITHM A7.3
  MakeOpenConic(P0,T0,P2,T2,P,n,U,Pw)
    { /*  Construct open conic arc in 3D  */
      /*  Input:  P0,T0,P2,T2,P  */
      /*  Output: n,U,Pw  */
    MakeOneArc(P0,T0,P2,T2,P,P1,w1);
    if (w1 <= -1.0)      /* parabola or hyperbola */
      return(error);     /* outside convex hull */
    if (w1 >= 1.0)    /* classify type & number of segments */
      nsegs = 1;      /* hyperbola or parabola, one segment */
      else
      {   /* ellipse, determine number of segments */
      if (w1 > 0.0 && Angle(P0,P1,P2) > 60.0)   nsegs = 1;
      else
      if (w1 < 0.0 && angle(P0,P1,P2) > 90.0)   nsegs = 4;
        else        nsegs = 2;
      }
    n = 2*nsegs;
    j = 2*nsegs+1;
    for (i=0; i<3; i++)    /* load end knots */
      {  U[i] = 0.0;    U[i+j] = 1.0;  }
    Pw[0] = P0;   Pw[n] = P2;   /* load end ctrl pts */
    if (nsegs == 1)
```

```
    {
    Pw[1] = w1*P1;
    return;
    }
  SplitArc(P0,P1,w1,P2,Q1,S,R1,wqr);
  if (nsegs == 2)
    {
    Pw[2] = S;
    Pw[1] = wqr*Q1;     Pw[3] = wqr*R1;
    U[3] = U[4] = 0.5;
    return;
    }
    /* nsegs == 4 */
  Pw[4] = S;
  w1 = wqr;
  SplitArc(P0,Q1,w1,S,HQ1,HS,HR1,wqr);
  Pw[2] = HS;
  Pw[1] = wqr*HQ1;     Pw[3] = wqr*HR1;
  SplitArc(S,R1,w1,P2,HQ1,HS,HR1,wqr);
  Pw[6] = HS;
  Pw[5] = wqr*HQ1;     Pw[7] = wqr*HR1;
  for (i=0; i<2; i++)    /* load the remaining knots */
    {
    U[i+3] = 0.25;    U[i+5] = 0.5;    U[i+7] = 0.75;
    }
  return;
  }
```

Figures 7.26a and 7.26b show examples using two and four segments, respectively.

Algorithm A7.3 produces ellipses which have good parameterization. For applications where parameterization is not considered important, a simpler algorithm which produces C^2 continuous curves with equally good convex hulls (see [Pieg90]) is:

1. call MakeOneArc (Algorithm A7.2);
2. insert knots at appropriate locations to cut the corners and to eliminate negative weights or an infinite control point (e.g., at $u = 1/2$, or $u = 1/3$ and $u = 2/3$).

For comparison, Figure 7.26c shows an elliptic arc which was constructed using

- Algorithm A7.3 (points marked by □);
- MakeOneArc, with subsequent insertion of $u = 1/2$ one time (points marked by +).

There remains the problem of full ellipses. If the major and minor axes and radii are known, a rectangular control polygon is appropriate. The weights and

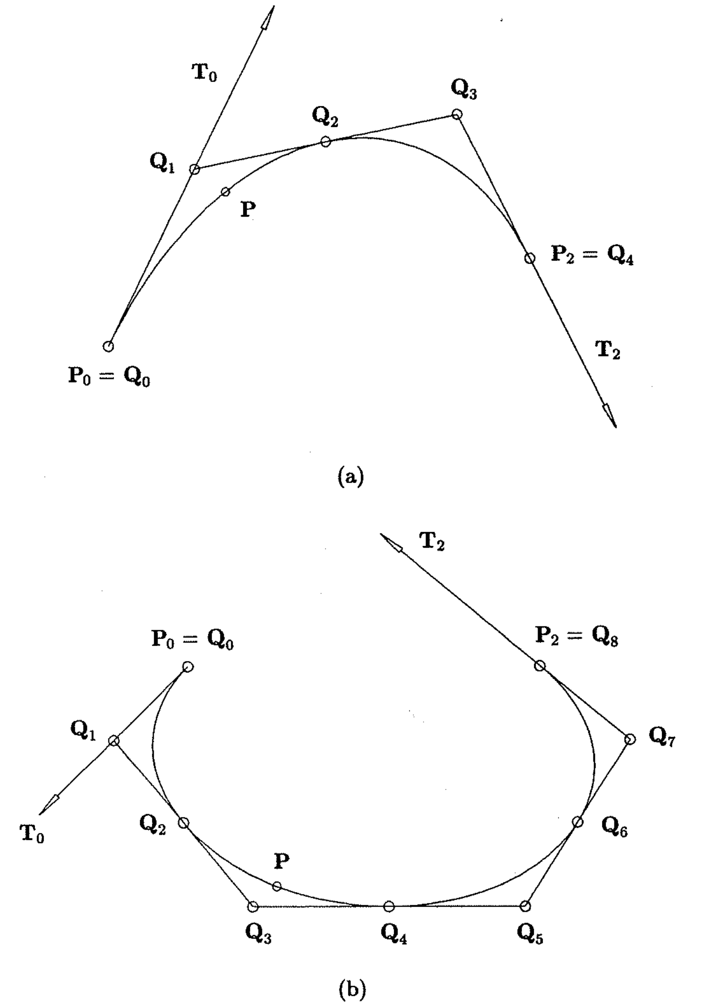


Figure 7.26. NURBS representation of conic segments. (a) Arc sweeping less than 180°; (b) arc sweeping more than 180°; (c) comparison between representations obtained by curve splitting (□) and knot insertion (+).

knots are the same as those of the circle in Example Ex7.2 (Figure 7.16a), or a configuration analogous to Example Ex7.6 (Figure 7.21a) can be used.

If input data is $\mathbf{P}_0, \mathbf{T}_0, \mathbf{P}_2, \mathbf{T}_2, \mathbf{P}$, together with the knowledge that the conic is a full ellipse (starting and ending at $\mathbf{P}_0$), then an appropriate construction is:

1. call MakeOneArc to get the rational Bézier representation of one segment of the ellipse (with positive weight w_1);

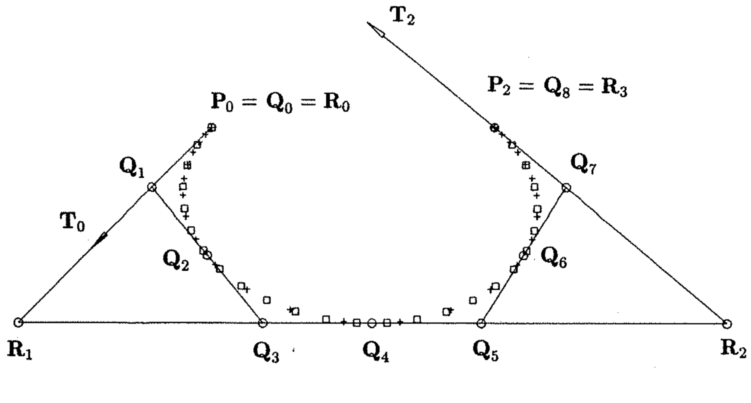


Figure 7.26. (*Continued.*)

2. compute the center of the ellipse $\mathbf{C}$;
3. compute the major and minor axes, (unit) vectors $(\mathbf{U}, \mathbf{V})$, along with the major and minor radii (r_1, r_2);
4. compute the control points

$$
\begin{aligned}
\mathbf{Q}_0 &= \mathbf{C} + r_1\mathbf{U} \quad (= \mathbf{Q}_8)\\
\mathbf{Q}_1 &= \mathbf{Q}_0 + r_2\mathbf{V}\\
\mathbf{Q}_2 &= \mathbf{C} + r_2\mathbf{V}\\
\mathbf{Q}_3 &= \mathbf{Q}_2 - r_1\mathbf{U}\\
\mathbf{Q}_4 &= \mathbf{C} - r_1\mathbf{U}\\
\mathbf{Q}_5 &= \mathbf{Q}_4 - r_2\mathbf{V}\\
\mathbf{Q}_6 &= \mathbf{C} - r_2\mathbf{V}\\
\mathbf{Q}_7 &= \mathbf{Q}_6 + r_1\mathbf{U}
\end{aligned}
$$

The details of Steps 2 and 3 are given in the next section, and the results are illustrated in Figure 7.28.

7.7 Conic Type Classification and Form Conversion

In previous sections we presented five forms for representing conics:

- implicit equation (Eqs. [7.3] and [7.5]–[7.7]);
- maximum inscribed area forms (Eqs. [7.15]–[7.17]);

- quadratic rational power basis (Eqs. [7.8]–[7.14]);
- quadratic rational Bézier (Eq. [7.21]);
- quadratic NURBS form.

In this section we discuss conic-type classification and form conversion, that is, given a curve in one of these forms, determine what type of conic it is – ellipse, parabola, or hyperbola – and convert it to another form. When we refer to the implicit form we assume that the conic is in the xy plane; otherwise the conic can be in three-dimensional space.

Consider the first two forms. If a conic is in standard position and can be described by one of Eqs. (7.5)–(7.7), its type is obvious. If it is not in standard position (Eq. [7.3]), then its type is determined by the values of α and D (determinant) as described in Section 7.2, and it can easily be transformed into standard position. For the maximum inscribed area forms, the conic type is inherent in the definition. Note that the maximum area form is quite similar to the standard position implicit form, in that their defining equations are given in terms of the conics' so-called *geometric characteristics*, i.e., axes, vertices, center, radii, etc. Hence, the conversion between these two forms is trivial.

Conversion between the quadratic rational power basis (Eq. [7.14]) and Bézier forms is given by the methods of Chapter 6, in particular Eqs. (6.89) and (6.93). Their rational counterparts are

$$[\mathbf{a}_i^w] = R_2 M_2 [\mathbf{P}_i^w] \tag{7.51}$$

and

$$[\mathbf{P}_i^w] = M_2^{-1} R_2^{-1} [\mathbf{a}_i^w] \tag{7.52}$$

where M_2 and R_2 are the quadratic (3×3) Bézier and reparameterization matrices, respectively. The type of a Bézier conic is easily gleaned from the conic shape factor, Eq. (7.25). The type of a power basis conic is computed from Eqs. (7.52) and (7.25). For simplicity, assume our power basis segment is defined on $0 \le u \le 1$, with R_2 the identity matrix. Denote by w_i^a the power basis weights (Eq. [7.14]), and by w_i^P the Bézier weights. Then from Eqs. (6.94) and (7.52) we find that

$$[w_i^P] = M_2^{-1}[w_i^a] = \begin{bmatrix} 1 & 0 & 0 \\ 1 & \frac{1}{2} & 0 \\ 1 & 1 & 1 \end{bmatrix} [w_i^a]$$

from which it follows that

$$
\begin{aligned}
w_0^P &= w_0^a\\
w_1^P &= w_0^a + \frac{1}{2}w_1^a\\
w_2^P &= w_0^a + w_1^a + w_2^a
\end{aligned} \tag{7.53}
$$

When substituted into Eq. (7.25), these w_i^P yield the conic type.

A quadratic NURBS curve with internal knots may or may not represent a unique conic (it is a conic, line, or point on each span). The classification of a quadratic NURBS curve is a three-step process:

1. using Eq. (6.62), compute the conic shape factor of each nondegenerate segment;
2. if all shape factors indicate a common type (all $c_i < 1, = 1,$ or > 1), decompose the curve into Bézier segments;
3. compute the geometric characteristics of each segment and compare them.

Notice that all shape factors can indicate a common type, and in fact may even be equal, but the curve may still not be a unique conic. As an example, recall that the weights of a circular arc depend only on the sweep angle (Eq. [7.33]); hence one could piece together circular arcs of different radii, but with equal sweep angles and shape factors.

Based on this discussion, for classification and conversion purposes the five forms can be put into two groups:

1. implicit and maximum inscribed area forms;
2. quadratic rational power basis, Bézier, and NURBS forms.

Conic representations are easily obtained within a group if certain data is available, that is

- geometric characteristics (Group 1);
- start points and end points, together with their tangent directions and an additional point, $\mathbf{P}_0, \mathbf{T}_0, \mathbf{P}_2, \mathbf{T}_2, \mathbf{P}$ (Group 2).

Using Algorithms `A7.1–A7.3`, conversion from a form in Group 1 to one in Group 2 is easy because it involves only point and tangent computation from Eqs. (7.3) and (7.5)–(7.7) or (7.15)–(7.17). Conversion from Group 2 to Group 1 requires more work. Beach [Beac91] derives the formulas for converting from the rational power basis to implicit form (Eq. [7.3]). We show here how to convert from rational Bézier form to Eq. (7.3), and how to compute the geometric characteristics of a conic from its rational Bézier form.

Let $\mathbf{C}(u)$ be the rational Bézier conic given by Eq. (7.21). Recalling Eqs. (7.22) through (7.26) and Figure 7.11

$$\alpha(u)\beta(u) = \frac{1}{4}k\big(1-\alpha(u)-\beta(u)\big)^2 \tag{7.54}$$

is the implicit equation of the conic in the oblique coordinate system given by $\{\mathbf{P}_1, \mathbf{S}, \mathbf{T}\}$, where

$$k = \frac{w_0 w_2}{{w_1}^2} \qquad \alpha(u) = R_{0,2}(u) \qquad \beta(u) = R_{2,2}(u)$$

In order to obtain Eq. (7.3) from Eq. (7.54) we need to express α and β in terms of Cartesian coordinates x and y. Equation (7.20) says that $\alpha(u)$ and $\beta(u)$ are the barycentric coordinates of the point $\mathbf{C}(u)$ in the triangle $\{\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2\}$. Fixing an arbitrary u_0, let $\alpha = \alpha(u_0)$, $\beta = \beta(u_0)$, and $\mathbf{P} = \mathbf{C}(u_0) = (x, y)$. Then

$$\alpha = \frac{\text{area}\,(\mathbf{P}, \mathbf{P}_1, \mathbf{P}_2)}{\text{area}\,(\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2)} \qquad \beta = \frac{\text{area}\,(\mathbf{P}, \mathbf{P}_0, \mathbf{P}_1)}{\text{area}\,(\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2)}$$

from which it follows that

$$\begin{aligned} \alpha &= \frac{(y-y_1)(x_2-x_1)-(x-x_1)(y_2-y_1)}{(y_0-y_1)(x_2-x_1)-(x_0-x_1)(y_2-y_1)} \\ \beta &= \frac{(y-y_1)(x_0-x_1)-(x-x_1)(y_0-y_1)}{(y_2-y_1)(x_0-x_1)-(x_2-x_1)(y_0-y_1)} \end{aligned} \tag{7.55}$$

Now let

$$h_0 = x_0 - x_1 \qquad h_1 = x_2 - x_1$$
$$h_2 = y_0 - y_1 \qquad h_3 = y_2 - y_1$$

and

$$g_1 = h_2 h_1 - h_0 h_3$$

Then $$\alpha = \frac{h_1(y-y_1)-h_3(x-x_1)}{g_1} \qquad \beta = \frac{h_0(y-y_1)-h_2(x-x_1)}{-g_1} \tag{7.56}$$

Finally, setting $s_1 = h_0 - h_1$ and $s_2 = h_3 - h_2$, and substituting Eq. (7.56) into Eq. (7.54), yields the six coefficients of Eq. (7.3), i.e.

$$\begin{aligned} a &= s_2^2 + \frac{4}{k}h_2h_3 \\ b &= s_1^2 + \frac{4}{k}h_0h_1 \\ h &= s_1s_2 - \frac{2}{k}(h_0h_3 + h_1h_2) \\ f &= -hy_1 - ax_1 + g_1s_2 \\ g &= -hx_1 - by_1 + g_1s_1 \\ c &= by_1^2 + 2hx_1y_1 + ax_1^2 - 2g_1(y_1s_1 + x_1s_2) + g_1^2 \end{aligned} \tag{7.57}$$

We turn now to the computation of the geometric characteristics from the rational Bézier form. We present the formulas here without proof: although not difficult, the derivations are involved. For the proofs we refer the reader to Lee's elegant article [Lee87], which is the source of this material.

We are given a quadratic rational Bézier conic, defined by Eq. (7.21), with all $w_i > 0$. We assume the conic is not degenerate. We define the symbols

$$\mathbf{S} = \mathbf{P}_0 - \mathbf{P}_1 \qquad \mathbf{T} = \mathbf{P}_2 - \mathbf{P}_1$$

$$k = \frac{w_0 w_2}{w_1^2} \qquad \epsilon = \frac{k}{2(k-1)}$$

$$\alpha = |\mathbf{S}|^2 \qquad \beta = \mathbf{S}\cdot\mathbf{T} \qquad \gamma = |\mathbf{T}|^2$$

$$\delta = \alpha\gamma - \beta^2 = |\mathbf{S}\times\mathbf{T}|^2$$

$$\zeta = \alpha + \gamma + 2\beta = |\mathbf{S}+\mathbf{T}|^2$$

$$\eta = \alpha + \gamma - 2\beta = |\mathbf{S}-\mathbf{T}|^2$$

For the parabola we have the formulas

$$\text{Axis} \;:\; \frac{1}{\sqrt{\zeta}}(\mathbf{S}+\mathbf{T}) \tag{7.58}$$

$$\text{Focus} \;:\; \mathbf{P}_1 + \frac{\gamma\mathbf{S}+\alpha\mathbf{T}}{\zeta} \tag{7.59}$$

$$\text{Vertex} \;:\; \mathbf{P}_1 + \left(\frac{\gamma+\beta}{\zeta}\right)^2\mathbf{S} + \left(\frac{\alpha+\beta}{\zeta}\right)^2\mathbf{T} \tag{7.60}$$

The center of an ellipse or hyperbola is given by

$$\text{Center} \;:\; \mathbf{P}_1 + \epsilon(\mathbf{S}+\mathbf{T}) \tag{7.61}$$

Let $\lambda_1 \le \lambda_2$ be the solutions to the quadratic equation

$$2\delta\lambda^2 - (k\eta + 4\beta)\lambda + 2(k-1) = 0$$

The roots of this equation are real. For an ellipse, $\lambda_2 \ge \lambda_1 > 0$ and $\epsilon > 0$. For a hyperbola, $\lambda_1 < 0 < \lambda_2$ and $\epsilon < 0$. The major and minor radii are

$$\text{Ellipse} \;:\; r_1 = \sqrt{\frac{\epsilon}{\lambda_1}} \qquad r_2 = \sqrt{\frac{\epsilon}{\lambda_2}} \tag{7.62}$$

$$\text{Hyperbola} \;:\; r_1 = \sqrt{\frac{\epsilon}{\lambda_1}} \qquad r_2 = \sqrt{\frac{-\epsilon}{\lambda_2}} \tag{7.63}$$

Define the following

$$\text{if} \quad \left|\frac{k}{2} - \gamma\lambda_1\right| > \left|\frac{k}{2} - \alpha\lambda_1\right|$$

$$\bar{x} = \frac{k}{2} - \gamma\lambda_1 \qquad \bar{y} = \beta\lambda_1 - \frac{k}{2} + 1$$

$$\text{else} \quad \bar{x} = \beta\lambda_1 - \frac{k}{2} + 1 \qquad \bar{y} = \frac{k}{2} - \alpha\lambda_1$$

and

$$\rho = \alpha\bar{x}^2 + 2\beta\bar{x}\bar{y} + \gamma\bar{y}^2$$

$$x_0 = \frac{\bar{x}}{\rho} \qquad y_0 = \frac{\bar{y}}{\rho}$$

Then the points $\mathbf{Q}_1$ and $\mathbf{Q}_2$

$$\mathbf{Q}_1 = \mathbf{P}_1 + (\epsilon + r_1x_0)\mathbf{S} + (\epsilon + r_1y_0)\mathbf{T}$$

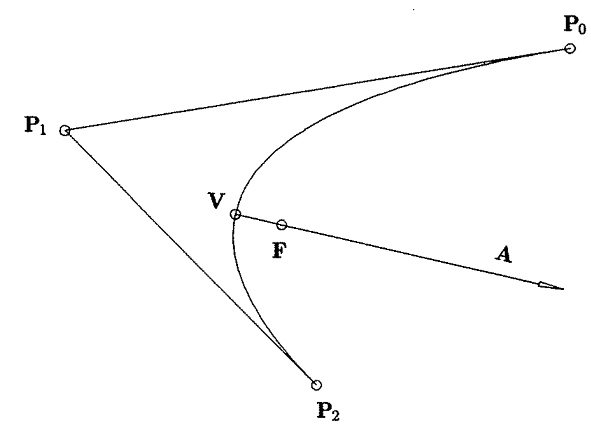
$$\mathbf{Q}_2 = \mathbf{P}_1 + (\epsilon - r_1x_0)\mathbf{S} + (\epsilon - r_1y_0)\mathbf{T} \tag{7.64}$$

lie on the major axis of the ellipse or on the transverse axis of the hyperbola. The other orthogonal axis is easily obtained, and the vertices of the ellipse or the hyperbola are computed from the center, the radii, and the axis directions.

Note that for the circle, which is a special case of the ellipse, Eq. (7.62) yields $r_1 = r_2$. An algorithm should check for this case, because it makes no sense to apply Eq. (7.64) to a circle.

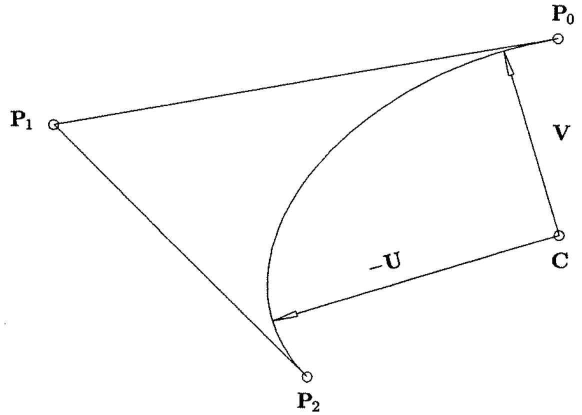
Figures 7.27a–7.27c illustrate these formulas. Figure 7.27a shows a parabolic arc along with the vertex (**V**), focus (**F**), and axis vector (**A**). In Figure 7.27b an elliptical arc is depicted, with center (**C**), negative major axis ($-\mathbf{U}$), and minor axis (**V**). Figure 7.27c shows the hyperbolic case. Again, center (**C**), major (**U**) and minor (**V**) axes are computed. This figure also shows how to compute the asymptotes of the arc, $\mathcal{A}_1$ and $\mathcal{A}_2$, i.e., the lines that are tangential to the curve at two points at infinity.

Finally, Figure 7.28 shows how to compute the NURBS representation of a full ellipse, given an arc in Bézier form (see the four steps given at the end of the previous section).

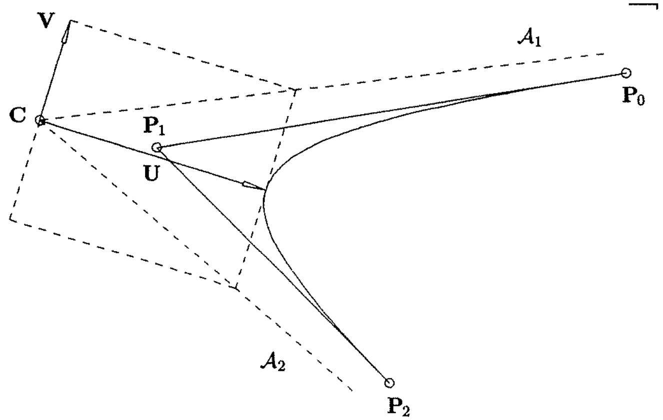


(a)

Figure 7.27. Geometric data computed from the Bézier representation. (a) A parabola with vertex (**V**), focus (**F**), and axis (**A**); (b) an ellipse with center (**C**), negative major axis ($-\mathbf{U}$), and minor axis (**V**); (c) a hyperbola with center (**C**), major axis (**U**), minor axis (**V**), and the two asymptotes ($\mathcal{A}_1$ and $\mathcal{A}_2$).



(b)



(c)

Figure 7.27. (*Continued.*)

7.8 Higher Order Circles

In some applications it is useful to represent full circles or circular arcs of sweep angle greater than 180° with one rational Bézier segment (no internal knots), no infinite control points, and no negative weights. In previous sections we eliminated infinite control points and negative weights by using knot insertion. Degree elevation also does the trick. A few examples are:

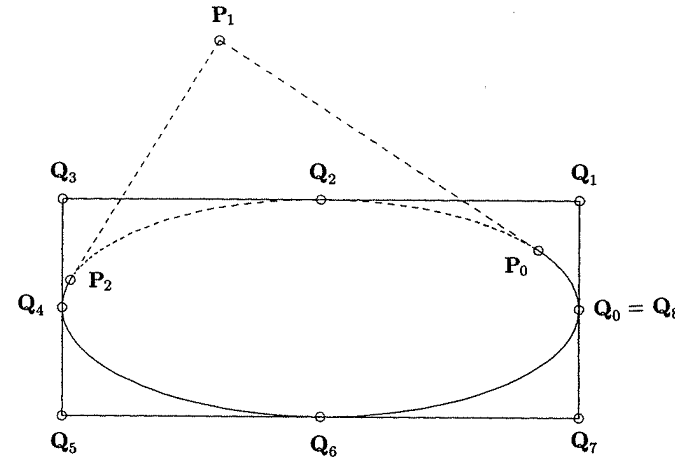


Figure 7.28. NURBS representation of a full ellipse given in Bézier form (dashed line).

Examples

`Ex7.7` The semicircle of Example `Ex7.1` can be degree elevated to yield

$$\mathbf{Q}_0^w = \mathbf{P}_0^w \qquad \mathbf{Q}_3^w = \mathbf{P}_2^w$$

$$\mathbf{Q}_1^w = \frac{1}{3}\mathbf{P}_0^w + \frac{2}{3}\mathbf{P}_1^w = \left(\frac{1}{3}, \frac{2}{3}, \frac{1}{3}\right)$$

$$\mathbf{Q}_2^w = \frac{2}{3}\mathbf{P}_1^w + \frac{1}{3}\mathbf{P}_2^w = \left(-\frac{1}{3}, \frac{2}{3}, \frac{1}{3}\right)$$

Thus (see Figure 7.29)

$$U = \{0,0,0,0,1,1,1,1\}$$

$$\{w_i\} = \left\{1, \frac{1}{3}, \frac{1}{3}, 1\right\}$$

$$\{\mathbf{Q}_i\} = \{(1,0),(1,2),(-1,2),(-1,0)\}$$

`Ex7.8` The arc of 240° given in Example `Ex7.4` is degree elevated to yield (see Figure 7.30):

$$U = \{0,0,0,0,1,1,1,1\}$$

$$\{w_i\} = \left\{1, \frac{1}{6}, \frac{1}{6}, 1\right\}$$

$$\{\mathbf{Q}_i\} = \left\{\left(a, \frac{1}{2}\right), (2a,-3), (-2a,-3), \left(-a, \frac{1}{2}\right)\right\}$$

with $a = \sqrt{3}/2$.

The following fourth- and fifth-degree full circle constructions are due to Chou [Chou95] and Strotman [Stro91].

`Ex7.9` We write the semicircle (with radius 1) in the complex plane as

$$z = e^{i\pi u} = f(u) = g(u) + h(u)i \qquad 0 \le u \le 1 \tag{7.65}$$

Squaring Eq. (7.65), we obtain the full circle

$$e^{i2\pi u} = (g^2 - h^2) + (2gh)i \qquad 0 \le u \le 1 \tag{7.66}$$

From Example `Ex7.1` we know that

$$g(u) = \frac{1-2u}{1-2u+2u^2} \qquad h(u) = \frac{2u(1-u)}{1-2u+2u^2} \tag{7.67}$$

Substituting Eq. (7.67) into Eq. (7.66) and returning to the real xy plane, we obtain

$$x(u) = \frac{1-4u+8u^3-4u^4}{1-4u+8u^2-8u^3+4u^4}$$

$$y(u) = \frac{4u(1-3u+2u^2)}{1-4u+8u^2-8u^3+4u^4} \tag{7.68}$$

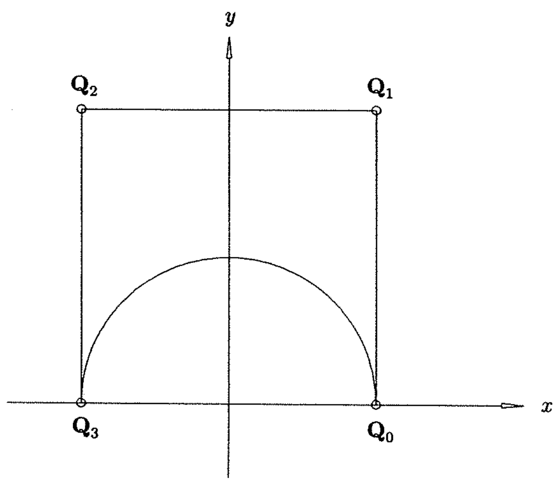


Figure 7.29. Rational cubic Bézier representation of the semicircle.

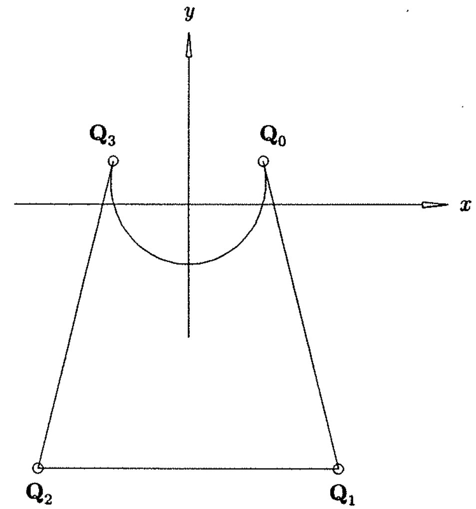


Figure 7.30. Rational cubic Bézier representation of an arc sweeping 240°.

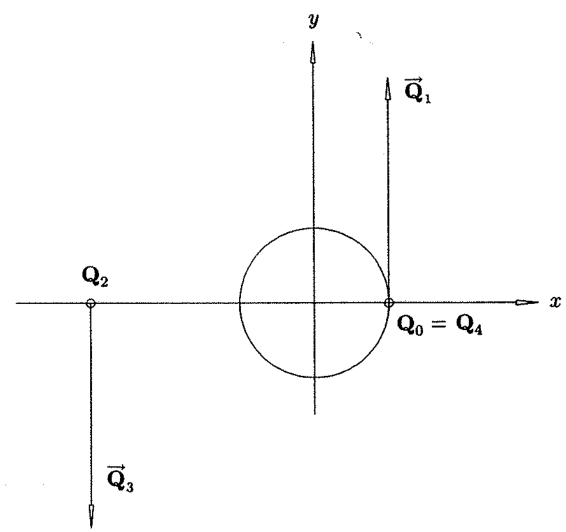


Figure 7.31. Rational quartic representation of the full circle using two infinite control points.

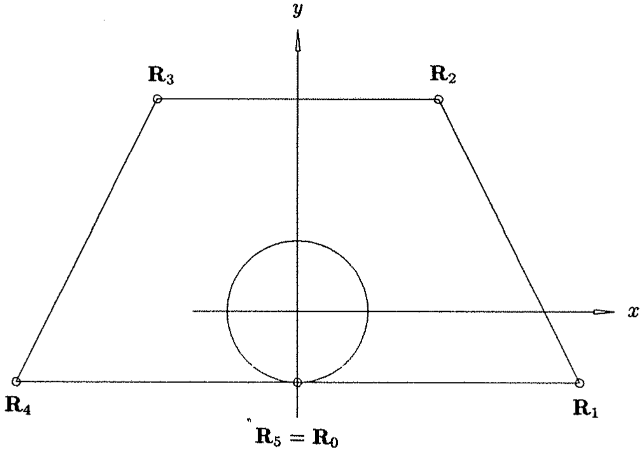


Figure 7.32. Rational quintic representation of the full circle using only finite control points.

Converting to Bézier form yields

$$\{\mathbf{Q}_i^w\} = \{(3,0,3),(0,3,0),(-3,0,1),(0,-3,0),(3,0,3)\}$$

This is a fourth-degree full circle with two infinite control points (see Figure 7.31).

Ex7.10 Simple degree elevation of the fourth-degree circle of Example Ex7.9 yields

$$\mathbf{R}_0^w = (0,-5,5) \quad \mathbf{R}_1^w = (4,-1,1) \quad \mathbf{R}_2^w = (2,3,1)$$
$$\mathbf{R}_3^w = (-2,3,1) \quad \mathbf{R}_4^w = (-4,-1,1) \quad \mathbf{R}_5^w = (0,-5,5)$$

This is a fifth-degree Bézier full circle using only finite control points and positive weights (see Figure 7.32).

Chou [Chou95] proves that it is impossible to obtain a fourth-degree Bézier full circle using only positive weights and finite control points.

Finally, we remind the reader that circles are simply perspective projections of curves which lie on a right circular cone. Thus any such curves, including nonplanar ones, will do. The possibilities are endless.

Exercises

7.1. Prove the statement made in Section 7.3, "Due to our choice of $w_0 = w_2 = 1$, it follows that the tangent to the conic at $\mathbf{S} = \mathbf{C}(1/2)$ is parallel to $\mathbf{P}_0\mathbf{P}_2$." Hint: Use Eqs. (7.25) and (7.26) and the properties of similar triangles.

7.2. Consider Figure 7.13 and Eq. (7.33). They imply that w_1 depends only on the sweep angle (not radius). Consider what happens to w_1 in the two limiting cases:

- hold $\mathbf{P}_0$ and $\mathbf{P}_2$ fixed and let $\mathbf{P}_1$ go to infinity;
- fix a circle, and let $\mathbf{P}_0$ and $\mathbf{P}_2$ come together.

7.3. The two concepts of "setting a weight to zero" and "infinite coefficients" also exist for the rational power basis representation. Is one of these concepts being used to obtain Eqs. (7.8) and (7.9)? If so, which one?

7.4. Another representation of the full circle is obtained by using the nine-point square control polygon and the knots of Example Ex7.2 but with weights $\{w_i\} = \{1,1,2,1,1,1,2,1,1\}$. Investigate the continuity at $u = 1/2$. Use formulas from Chapter 5 to remove the knot $u = 1/2$ as many times as possible. Compare this with Example Ex7.2 and explain the results. Visualize the two circles lying on the cone in homogeneous space.

7.5. Use Eqs. (7.53) and (7.25) to verify that Eqs. (7.8) and (7.9) represent an ellipse and a hyperbola, respectively.

7.6. A NURBS curve is defined by

$$U = \left\{0,0,0,\frac{1}{2},1,1,1\right\}$$
$$\mathbf{P}_i = \{(1,0),(1,1),(-2,1),(-2,-1)\}$$
$$w_i = \left\{1,\frac{1}{2},\frac{1}{2},1\right\}$$

Is this curve a unique conic?