16 Curves

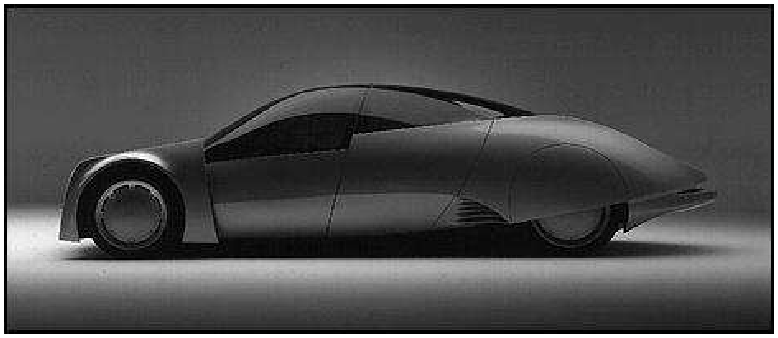

Figure 16.1.
Car design: curves are used to design cars such as the the Ford Synergy 2010 concept car. Source www.ford.com/concept.

Earlier in this book, we mentioned that all letters that you see here were designed by a font designer, and then put into a font library. The font designer's main tool is a cubic curve, also called a cubic Bézier curve. Such curves are handy for font design, but they were initially invented for car design. This happened in France in the early 1960's at Rénault and Citroën in Paris. These techniques are still in use today, as illustrated in Figure 16.1. We will briefly outline this kind of curve, and also apply previous geometric concepts to the study of curves in general. This type of work is called *Geometric Modeling* or *Computer Aided Geometric Design*, see [5] or [12].

16.1 Parametric Curves

You will recall that one way to write a straight line was the *parametric* form:

$$\mathbf{x}(t) = (1-t)\mathbf{a} + t\mathbf{b}.$$

If we interpret t as time, then this says at time $t = 0$, a moving point is at $\mathbf{a}$. It moves towards $\mathbf{b}$, and reaches it at time $t = 1$.

Let us be a bit more ambitious now and study motion along *curves*, i.e., paths that do not have to be straight. The simplest example is that of driving a car along a road. At time $t = 0$, you start, you follow the road, and at time $t = 1$, you have arrived somewhere. It does not really matter what kind of units we use to measure time; the $t = 0$ and $t = 1$ may just be viewed as a normalization of an arbitrary time interval.

Sketch 154.
A Bézier polygon.

We will now attack the problem of modeling curves, and we will choose a particularly simple way of doing this, namely cubic *Bézier curves*. We start with four points in 2D or 3D, called $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2$, and $\mathbf{b}_3$. Connect them with straight lines as shown in Sketch 16.1. The resulting polygon is called a *Bézier polygon.*

Here is how you generate a curve from it. Pick a parameter value t between 0 and 1. Find the corresponding point on each polygon leg by linear interpolation. This gives you three points $\mathbf{b}_0^1, \mathbf{b}_1^1, \mathbf{b}_2^1$. They form a polygon themselves – repeat the linear interpolation process and you get two points $\mathbf{b}_0^2, \mathbf{b}_1^2$. Repeat one more time, and you have a point $\mathbf{b}_0^3$. This is the point on the Bézier curve defined by $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ at the parameter value t.

The recursive process of applying linear interpolation is called the *de Casteljau algorithm*, and it is shown in Sketch 16.1.

Sketch 155.
The de Casteljau algorithm.

EXAMPLE 16.1

A numerical counterpart to Sketch 16.1 follows. Let the polygon be given by

$$\mathbf{b}_0 = \begin{bmatrix} 4 \\ 4 \end{bmatrix}, \quad \mathbf{b}_1 = \begin{bmatrix} 0 \\ 8 \end{bmatrix}, \quad \mathbf{b}_2 = \begin{bmatrix} 8 \\ 8 \end{bmatrix}, \quad \mathbf{b}_3 = \begin{bmatrix} 8 \\ 0 \end{bmatrix}.$$

For simplicity, let $t = 1/2$. Linear interpolation is then nothing but finding midpoints, and we have

$$\mathbf{b}_0^1 = \frac{1}{2}\mathbf{b}_0 + \frac{1}{2}\mathbf{b}_1 = \begin{bmatrix} 2 \\ 6 \end{bmatrix}, \quad \mathbf{b}_1^1 = \frac{1}{2}\mathbf{b}_1 + \frac{1}{2}\mathbf{b}_2 = \begin{bmatrix} 4 \\ 8 \end{bmatrix}, \quad \mathbf{b}_2^1 = \frac{1}{2}\mathbf{b}_2 + \frac{1}{2}\mathbf{b}_3 = \begin{bmatrix} 8 \\ 4 \end{bmatrix}.$$

Next,

$$\mathbf{b}_0^2 = \frac{1}{2}\mathbf{b}_0^1 + \frac{1}{2}\mathbf{b}_1^1 = \begin{bmatrix} 3 \\ 7 \end{bmatrix}, \quad \mathbf{b}_1^2 = \frac{1}{2}\mathbf{b}_1^1 + \frac{1}{2}\mathbf{b}_2^1 = \begin{bmatrix} 6 \\ 6 \end{bmatrix},$$

and finally

$$\mathbf{b}_0^3 = \frac{1}{2}\mathbf{b}_0^2 + \frac{1}{2}\mathbf{b}_1^2 = \begin{bmatrix} \frac{9}{2} \\ \frac{13}{2} \end{bmatrix}.$$

This is the point on the curve corresponding to $t = 1/2$.

In general, we have

$$\begin{aligned} \mathbf{b}_0^3 &= (1-t)\mathbf{b}_0^2 + t\mathbf{b}_1^2 \\ &= (1-t)[(1-t)\mathbf{b}_0^1 + t\mathbf{b}_1^1] + t[(1-t)\mathbf{b}_1^1 + t\mathbf{b}_2^1] \\ &= (1-t)[(1-t)[(1-t)\mathbf{b}_0 + t\mathbf{b}_1] + t[(1-t)\mathbf{b}_1 + t\mathbf{b}_2] + t[(1-t)[(1-t)\mathbf{b}_1 + t\mathbf{b}_2] + t[(1-t)\mathbf{b}_2 + t\mathbf{b}_3 \end{aligned}$$

After some re-assembly, this becomes

$$\mathbf{b}_0^3(t) = (1-t)^3\mathbf{b}_0 + 3(1-t)^2 t\mathbf{b}_1 + 3(1-t)t^2\mathbf{b}_2 + t^3\mathbf{b}_3. \tag{16.1}$$

This is the general form of a cubic Bézier curve. As t traces out values between 0 and 1, the point $\mathbf{b}_0^3(t)$ traces out a curve. See Figure 16.2 for some examples. From now on, we will also use the shorter $\mathbf{b}(t)$ instead of $\mathbf{b}_0^3(t)$.

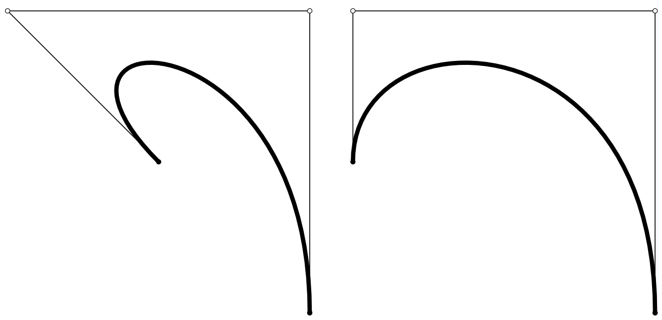

Figure 16.2.

Bézier curves: two examples.

The original curve was given by the control polygon

$$\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3.$$

Inspection of Sketch 16.1 suggests that the two curve segments generated by $\mathbf{b}(t)$ also have control polygons. This is indeed so; the curve segment from $\mathbf{b}_0$ to $\mathbf{b}(t)$ has

$$\mathbf{b}_0, \mathbf{b}_0^1, \mathbf{b}_0^2, \mathbf{b}_0^3$$

as its control polygon. The segment from $\mathbf{b}(t)$ to $\mathbf{b}_3$ has

$$\mathbf{b}_0^3, \mathbf{b}_1^2, \mathbf{b}_2^1, \mathbf{b}_3$$

as its control polygon. This process: generating two Bézier curves from one, is called *subdivision*.

16.2 Properties of Bézier Curves

From inspection of the examples, but also from (16.1), we see that the curve passes through the first and last control points:

$$\mathbf{b}(0) = \mathbf{b}_0, \quad \mathbf{b}(1) = \mathbf{b}_3. \tag{16.2}$$

If we map the control polygon using an affine map, then the curve undergoes the same transformation, as shown in Figure 16.3. This can be seen by inspecting the examples, but also

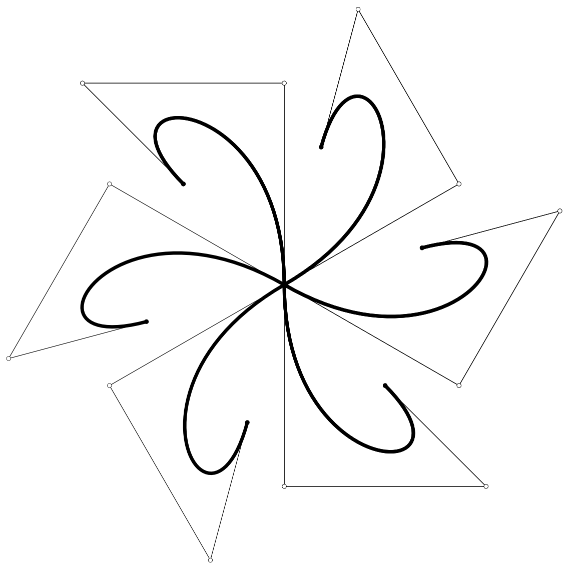

Figure 16.3.

Bézier curves: as the control polygon rotates, so does the curve.

by observing that the cubic coefficients of the control points in (16.1) sum to one. This can be seen as follows:

$$(1-t)^3 + 3(1-t)^2 t + 3(1-t)t^2 + t^3 = [(1-t) + t]^3 = 1.$$

Thus every point on the curve is a *barycentric combination* of the control points. Such relationships are not changed under affine maps, as per Section 6.1.

The curve also lies in the convex hull of the control polygon – a fact called the *convex hull property* . This can be seen by observing that the coefficients of the control points in (16.1) are nonnegative (recall that we only consider values of t between 0 and 1). It follows that every point on the curve is a *convex combination* of the control points, and hence is inside their *convex hull*. For a definition, see Section 8.4; for an illustration, see Figure 16.4.

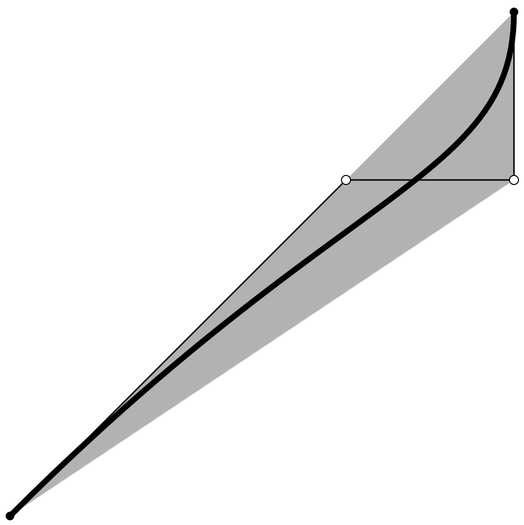

Figure 16.4.
Bézier curves: the curve lies in the convex hull of the control polygon.

Clearly the control polygon is inside its minmax box.[1] Because of the convex hull property, we also know that the curve is inside this box – a property that has numerous applications. See Figure 16.5 for an illustration.

[1]Recall that the minmax box of a polygon is the smallest rectangle with edges parallel to the coordinate axes that contains the polygon.

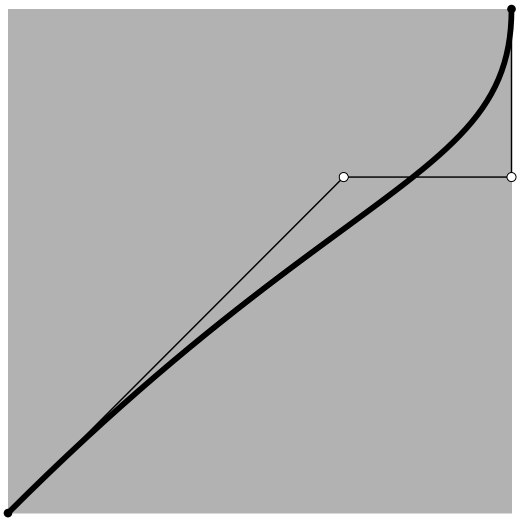

Figure 16.5.
Bézier curves: the curve lies inside the minmax box of the control polygon.

16.3 The Matrix Form

As a preparation for what is to follow, let us rewrite (16.1) using the formalism of dot products. It then looks like this:

$$\mathbf{b}(t) = \begin{bmatrix} \mathbf{b}_0 & \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \end{bmatrix} \begin{bmatrix} (1-t)^3 \\ 3(1-t)^2 t \\ 3(1-t)t^2 \\ t^3 \end{bmatrix}. \qquad (16.3)$$

Most people think of polynomials as combinations of the *monomials*; they are $1, t, t^2, t^3$ for the cubic case. Our expression (16.1) may be rewritten into this form:

$$\mathbf{b}(t) = \mathbf{b}_0 + 3t(\mathbf{b}_1 - \mathbf{b}_0) + 3t^2(\mathbf{b}_2 - 2\mathbf{b}_1 + \mathbf{b}_0) + t^3(\mathbf{b}_3 - 3\mathbf{b}_2 + 3\mathbf{b}_1 - \mathbf{b}_0).$$

This allows a more concise formulation using matrices:

$$\mathbf{b}(t) = \begin{bmatrix} \mathbf{b}_0 & \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \end{bmatrix} \begin{bmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ t \\ t^2 \\ t^3 \end{bmatrix}. \tag{16.4}$$

This is the matrix form of a Bézier curve.

Equation (16.4) shows how to write a Bézier curve in monomial form. A curve in monomial form looks like this:

$$\mathbf{b}(t) = \mathbf{a}_0 + \mathbf{a}_1 t + \mathbf{a}_2 t^2 + \mathbf{a}_3 t^3.$$

Rewritten using the dot product form, this becomes

$$\mathbf{b}(t) = \begin{bmatrix} \mathbf{a}_0 & \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \end{bmatrix} \begin{bmatrix} 1 \\ t \\ t^2 \\ t^3 \end{bmatrix}.$$

Thus the monomial $\mathbf{a}_i$ are defined as

$$\begin{bmatrix} \mathbf{a}_0 & \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_0 & \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \end{bmatrix} \begin{bmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{16.5}$$

How about the inverse process: If we are given a curve in monomial form, how can we write it as a Bézier curve? Simply rearrange (16.5) to solve for the $\mathbf{b}_i$:

$$\begin{bmatrix} \mathbf{b}_0 & \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{a}_0 & \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \end{bmatrix} \begin{bmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1}.$$

A matrix inversion is all that is needed here!

Notice that the square matrix in this equation is nonsingular. Because of its nonsingularity, we can conclude that any cubic curve can be written in either the Bézier or the monomial form.

16.4 Derivatives

Equation (16.1) consists of two (in 2D) or three (in 3D) cubic equations in t. We can take the derivative in each of the components:

$$\frac{\mathrm{d}\mathbf{b}(t)}{\mathrm{d}t} = -3(1-t)^2\mathbf{b}_0 - 6(1-t)t\mathbf{b}_1 + 3(1-t)^2\mathbf{b}_1 - 3t^2\mathbf{b}_2 + 6(1-t)t\mathbf{b}_2 + 3t^2\mathbf{b}_3.$$

Rearranging, and using the abbreviation $\frac{\mathrm{d}\mathbf{b}(t)}{\mathrm{d}t} = \dot{\mathbf{b}}(t)$, we have

$$\dot{\mathbf{b}}(t) = 3(1-t)^2[\mathbf{b}_1 - \mathbf{b}_0] + 6(1-t)t[\mathbf{b}_2 - \mathbf{b}_1] + 3t^2[\mathbf{b}_3 - \mathbf{b}_2]. \tag{16.6}$$

As expected, the derivative of a degree three curve is one of degree two.[2]

For $t = 0$, we obtain

$$\dot{\mathbf{b}}(0) = 3[\mathbf{b}_1 - \mathbf{b}_0],$$

and, similarly, for $t = 1$,

$$\dot{\mathbf{b}}(1) = 3[\mathbf{b}_3 - \mathbf{b}_2].$$

In words, the control polygon is tangent to the curve at the curve's endpoints. This is not a surprising statement if you check the example figures!

EXAMPLE 16.2

Let us compute the derivative of the example curve from Section 16.1 for $t = 1/2$. We obtain

$$\dot{\mathbf{b}}(\frac{1}{2}) = 3 \cdot \frac{1}{4}[\begin{bmatrix}0\\8\end{bmatrix} - \begin{bmatrix}4\\4\end{bmatrix}] + 6 \cdot \frac{1}{4}[\begin{bmatrix}8\\8\end{bmatrix} - \begin{bmatrix}0\\8\end{bmatrix}] + 3 \cdot \frac{1}{4}[\begin{bmatrix}8\\0\end{bmatrix} - \begin{bmatrix}8\\8\end{bmatrix}],$$

which yields

$$\dot{\mathbf{b}}(\frac{1}{2}) = \begin{bmatrix}9\\-3\end{bmatrix}.$$

Sketch 156.
A derivative vector.

See Sketch 16.4 for an illustration.

Note that the derivative of a curve is a *vector*. It is tangent to the curve – apparent from our example, but nothing we want to prove here. A convenient way to think about the derivative is by interpreting it as a *velocity vector*. If you interpret the parameter t as time, and you think of traversing the curve such that at time t you have reached $\mathbf{b}(t)$, then the derivative measures your velocity. The larger the magnitude of the tangent vector, the faster you move.

If we rotate the control polygon, the curve will follow, and so will all of its derivative vectors. In calculus, a "horizontal tangent" has a special meaning; it indicates an extreme value of a function. Not here: the very notion of an extreme value is meaningless for parametric curves since the term "horizontal tangent" depends on the curve's orientation and is not a property of the curve itself.

We may take the derivative of (16.6) with respect to t. We then have the *second derivative*. It is given by

$$\ddot{\mathbf{b}}(t) = -6(1-t)[\mathbf{b}_1-\mathbf{b}_0]-6t[\mathbf{b}_2-\mathbf{b}_1]+6(1-t)[\mathbf{b}_2-\mathbf{b}_1]+6t[\mathbf{b}_3-\mathbf{b}_2]$$

and may be rearranged to

$$\ddot{\mathbf{b}}(t) = 6(1-t)[\mathbf{b}_2 - 2\mathbf{b}_1 + \mathbf{b}_0] + 6t[\mathbf{b}_3 - 2\mathbf{b}_2 + \mathbf{b}_1]. \qquad (16.7)$$

Sketch 157.
A second derivative vector.

The second derivative at $\mathbf{b}_0$ (see Sketch 16.4) is particularly simple – it is given by

$$\ddot{\mathbf{b}}(0) = 6[\mathbf{b}_2 - 2\mathbf{b}_1 + \mathbf{b}_0].$$

Loosely speaking, we may interpret the second derivative $\ddot{\mathbf{b}}(t)$ as acceleration when traversing the curve.

[2]Note that the derivative curve does not have control *points* anymore, but rather *control vectors*!

16.5 Composite Curves

A Bézier curve is a handsome tool, but one such curve would rarely suffice for describing much of any shape! For "real" shapes, we have to be able to line up many cubic Bézier curves. In order to define a smooth overall curve, these pieces must join smoothly.

This is easily achieved. Let $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ and $\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3$ be the control polygons of two Bézier curves with a common point $\mathbf{b}_3 = \mathbf{c}_0$ (see Sketch 16.5). If the two curves are to have the same tangent vector direction at $\mathbf{b}_3 = \mathbf{c}_0$, then all that is required is

$$\mathbf{c}_1 - \mathbf{c}_0 = c[\mathbf{b}_3 - \mathbf{b}_2] \tag{16.8}$$

for some positive real number c, meaning that the three points $\mathbf{b}_2, \mathbf{b}_3 = \mathbf{c}_0, \mathbf{c}_1$ are collinear.

If we use this rule to piece curve segments together, we can design many 2D and 3D shapes; Figure 16.6 gives an example.

Sketch 158.
Smoothly joining Bézier curves.

16.6 The Geometry of Planar Curves

The geometry of planar curves is centered around one concept: their *curvature*. It is easily understood if you imagine driving a car along a road. For simplicity, let's assume you are driving with constant speed. If the road does not curve, i.e., it is straight, you will not have to tilt your steering wheel. When the road does curve, you will have to tilt the steering wheel, and more so if the road curves rapidly. The curviness of the road (our model of a curve) is thus proportional to the tilting of the steering wheel.

Returning to the more abstract concept of a curve, let us sample its tangents at various points (see Sketch 16.6). Where the curve bends sharply, i.e., where its curvature is high, successive tangents differ from each other significantly. In areas where the curve is relatively flat, or where its curvature is low, successive tangents are almost identical. Curvature may thus

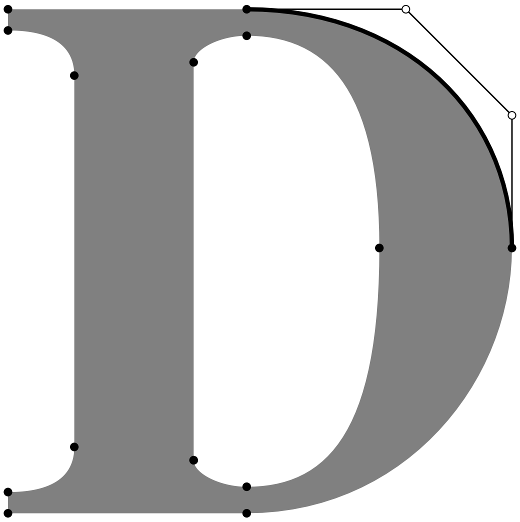

Figure 16.6.

Bézier curves: the letter "D" as a collection of cubic Bézier curves. Only one Bézier curve is shown.

Sketch 159.

Tangents on a curve.

be defined as *rate of change of tangents*. (In terms of our car example, the rate of change of tangents is proportional to the tilt of the steering wheel.)

Since the tangent is determined by the curve's first derivative, its rate of change should be determined by the second derivative. This is indeed so, but the actual formula for curvature is a bit more complex than can be derived in the context of this book. We denote the curvature of the curve at $\mathbf{b}(t)$ by κ; it is given by

$$\kappa(t) = \frac{\|\dot{\mathbf{b}} \wedge \ddot{\mathbf{b}}\|}{\|\dot{\mathbf{b}}\|^3}. \tag{16.9}$$

This formula holds for both 2D and 3D curves. In the 2D case, it may be rewritten as

$$\kappa(t) = \frac{|\dot{\mathbf{b}} \quad \ddot{\mathbf{b}}|}{\|\dot{\mathbf{b}}\|^3} \tag{16.10}$$

with the use of a 2×2 determinant. Since determinants may be positive or negative, curvature in 2D is *signed*. A point where $\kappa = 0$ is called an *inflection point*: the 2D curvature changes sign here. In Figure 16.7, the inflection point is marked. In

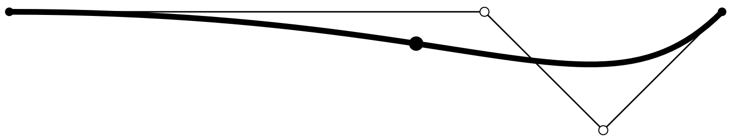

Figure 16.7.
Bézier curves: an inflection point is marked on the curve.

calculus, you learned that a curve has an inflection point if the second derivative vanishes. For parametric curves, the situation is different. An inflection point occurs when the first and second derivative vectors are parallel, or linearly dependent. This can lead to the curious effect of a cubic with *two* inflection points. It is illustrated in Figure 16.8.

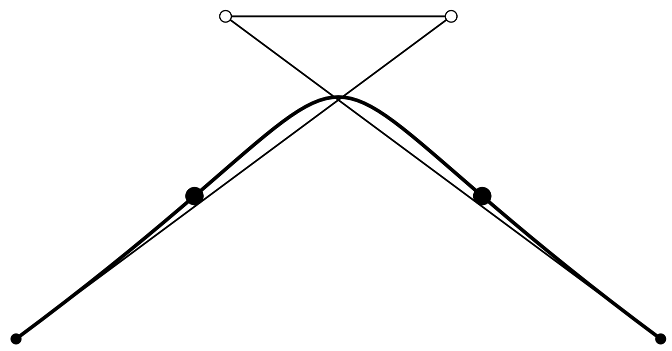

Figure 16.8.
Bézier curves: a cubic with two inflection points.

16.7 Moving along a Curve

Take a look at Figure 16.9. You will see the letter "B" sliding along a curve. If the curve is given in Bézier form, how can that

Figure 16.9.
Curve motions: a letter is moved along a curve.

Sketch 160.
Sliding along a curve.

effect be achieved?

The answer can be seen in Sketch 16.7. If you want to position an object, such as the letter "B," at a point on a curve, all you need to know is the point and the curve's tangent there.

If $\dot{\mathbf{b}}$ is the tangent, then simply define $\mathbf{n}$ to be a vector perpendicular to it.[3] Using the local coordinate system with origin $\mathbf{b}(t)$ and $[\dot{\mathbf{b}}, \mathbf{n}]$-axes, you can position any object as per Section 4.1.

The same story is far trickier in 3D! If you had a point on the curve and its tangent, the exact location of your object would not be fixed; it could still rotate around the tangent. Yet there is a unique way to position objects along a 3D curve. At every point on the curve, we may define a *local coordinate system* as follows.

Let the point on the curve be $\mathbf{b}(t)$; we now want to set up a local coordinate system defined by three vectors $\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3$. Following the 2D example, we set $\mathbf{f}_1$ to be in the tangent direction; $\mathbf{f}_1 = \dot{\mathbf{b}}(t)$. If the curve does not have an inflection point at t, then $\dot{\mathbf{b}}(t)$ and $\ddot{\mathbf{b}}(t)$ will not be collinear. This means that they span a plane, and that plane's normal is given by $\dot{\mathbf{b}}(t) \wedge \ddot{\mathbf{b}}(t)$. See Sketch 16.7 for some visual information. We make the plane's normal one of our local coordinate axes, namely $\mathbf{f}_3$. The plane, by the way, has a name: it is called the *osculating plane* at $\mathbf{x}(t)$. Since we have two coordinate axes, namely $\mathbf{f}_1$ and $\mathbf{f}_3$, it is not hard to come up with the remaining axis, we just set $\mathbf{f}_2 = \mathbf{f}_1 \wedge \mathbf{f}_3$. Thus for every point on the curve (as long as it is not an inflection point), there exists an orthogonal coordinate system. It is customary to use coordinate axes of unit length, and then we have

$$\mathbf{f}_1 = \frac{\dot{\mathbf{b}}(t)}{\|\dot{\mathbf{b}}(t)\|}, \tag{16.11}$$

$$\mathbf{f}_3 = \frac{\dot{\mathbf{b}}(t) \wedge \ddot{\mathbf{b}}(t)}{\|\dot{\mathbf{b}}(t) \wedge \ddot{\mathbf{b}}(t)\|}, \tag{16.12}$$

$$\mathbf{f}_2 = \mathbf{f}_1 \wedge \mathbf{f}_3. \tag{16.13}$$

This system with local origin $\mathbf{b}(t)$ and normalized axes $\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3$ is called the *Frenet frame* of the curve at $\mathbf{b}(t)$. Equipped with

[3] If $\dot{\mathbf{b}} = \begin{bmatrix} \dot{b}_1 \\ \dot{b}_2 \end{bmatrix}$ then $\mathbf{n} = \begin{bmatrix} -\dot{b}_2 \\ \dot{b}_1 \end{bmatrix}$.

For Sketch, see book

Sketch 161.
A Frenet frame.

the tool of Frenet frames, we may now position objects along a 3D curve! See Figure 16.10.

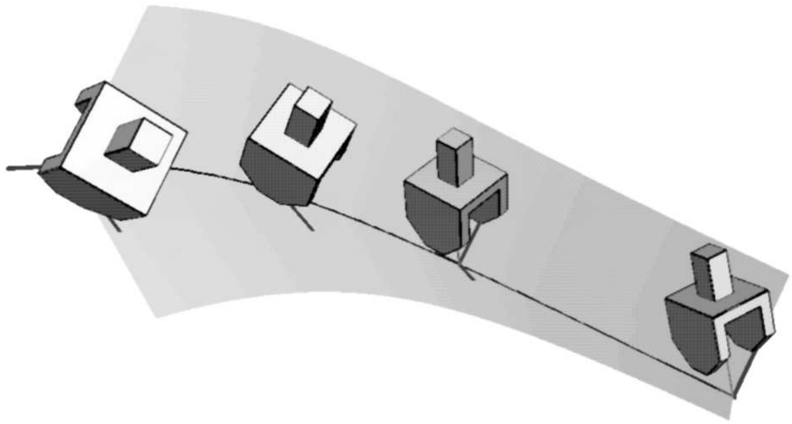

Figure 16.10.
Curve motions: a robot arm is moved along a curve. Courtesy of M. Wagner, Arizona State Universtiy.

Let us now work out exactly how to carry out our object-positioning plan. The object is given in some local coordinate system with axes $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$. Any point of the object has coordinates $\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$. It is mapped to

$$\mathbf{x}(t, \mathbf{u}) = \mathbf{x}(t) + u_1\mathbf{f}_1 + u_2\mathbf{f}_2 + u_3\mathbf{f}_3.$$

A typical application is *robot motion.* Robots are used extensively in automotive assembly lines; one job is to grab a part and move it to its destination inside the car body. This movement happens along well-defined curves. While the car part is being moved, it has to be oriented into its correct position – exactly the process described in this section!

16.8 Exercises

Let a cubic Bézier curve be given by the control polygon

$$\mathbf{b}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{b}_1 = \begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{b}_2 = \begin{bmatrix} 4 \\ 4 \\ 0 \end{bmatrix}, \quad \mathbf{b}_3 = \begin{bmatrix} 4 \\ 4 \\ 4 \end{bmatrix}.$$

1. Sketch this 3D curve manually.
2. Using the de Casteljau algorithm, evaluate it for $t = 1/4$.
3. Evaluate the first and second derivative for $t = 1/4$. Add these vectors to the sketch from exercise 1.
4. What is the control polygon for the curve defined from $t = 0$ to $t = 1/4$ and the curve defined over $t = 1/4$ to $t = 1$?
5. Rewrite it in monomial form.
6. Find its minmax box.
7. Find its curvatures at $t = 0$ and $t = 1/2$.
8. Find its Frenet frame for $t = 1$.