Original Lecture #6: 20 October 1992
Topics: Polynomials
Scribe: Peter Chang*

Today's lecture consists of two parts. First there will be a short discussion, continuing from last class, of the issues that must be considered when modeling curves and surfaces with sets of polynomial equations. We will then proceed to a discussion of the polar forms that we will be using to describe and manipulate splined curves and surfaces.

1 Issues in Modeling a Shape, Cont'd

1.1 Parametric vs Implicit Forms

Suppose we are given some smooth curve or surface S, lying in some larger ambient space Q. We will call Q the *object space*. Let s and q represent the dimensions of S and Q, respectively. We are interested in systems of polynomial equations that mathematically describe S.

There are two standard ways to represent S by such equations, namely parametric models and implicit models. In *parametric* models, the modeling function maps some other space (the parameter space) into Q. In *implicit* models the function maps Q onto some other space (the gauge space)—the shape S is then the collection of points in Q corresponding to the preimage of a gauge point under this map.

More precisely, for *parametric* models, the shape S is defined as the range $F(P)$ of a function $F : P \to Q$, where P is an auxiliary, $s$dimensional space called the *parameter space*. The input to the *parametric* modeling function is a point in parameter space—the coordinates of a "point" in the s-dimensional parameter space—and the output of the modeling function is a point on the shape S. For example, mapping $\mathscr{R}$ into $\mathscr{R}^3$ can be used to specify a space curve.

For *implicit models*, on the other hand, the shape S is defined by the formula $S = F^{-1}(\langle 0,\ldots,0\rangle)$, where $F : Q \to R$ is a function from the object space Q to some other auxiliary space R with dimensions $q-s$. The auxiliary space R is sometimes called the *gauge space*.

To make these ideas more concrete, consider as an example the standard parabola $y = x^2$. This shape is a smooth curve in the plane $Q = \mathscr{R}^2$, so $s = 1$ and $q = 2$.

The function $G : \mathscr{R} \to \mathscr{R}^2$ given by $G(t) = \langle t, t^2\rangle$ is a parametric model for that parabola. The variable t here denotes a parameter value. The parameter space $P = \mathscr{R}$ of a curve is one-dimensional, so it is convenient to think of the parameter as time. The modeling function $G : P \to Q$ maps times to points on the curve S.

The function $H : \mathscr{R}^2 \to \mathscr{R}$ given by $H(z) = H(\langle x, y\rangle) = y - x^2$, where z denotes a point in the object plane, is an implicit model for the same parabola. The corresponding gauge value $H(z)$ is positive, zero, or negative according as the point z lies above, on, or below the curve S.

*Heavily based on the 1991 notes of Eric Kolaczyk and David Karger.

Note that both parametric and implicit models of shapes encode some extra information, over and above the shape S itself. A parametric model, in addition to determining the shape S, also provides a 'roadmap' of S—a way to 'name' each point of S easily. It is natural to use the parametric form to look at select pieces of a shape, because one then only needs to specify the corresponding pieces of the parameter space P and look at the corresponding images under G. On the other hand, an implicit model, in addition to determining S, also associates gauge values with all of the other points in the object space.

While both of these forms have their uses, for the purposes of this class we will be using primarily the *parametric* model.

1.2 Single Equation vs Group of Equations

No matter what model we choose for representing our equations we must consider whether we are going to model a given curve by a single equation (of higher degree) or by a group of (lower degree) equations. While it may be possible to represent the curve we are interested in with a single equation this may not be practical, or desirable, for a variety of reasons.

Using a function of a high degree will mean gaining more flexibility over the available shapes compared to functions lower degrees. But it will also mean an increase in "wiggles" and instabilities. Piecewise smooth curves allow a tradeoff between the number of pieces and the complexity of each piece. A designer may use a large number of simple pieces or a few pieces of relatively high degree. Also, piecewise curves allow the designer *local control* in designing a shape. In other words, the designer is able to locally modify the shape, without affecting parts of it far away. This is very important for the interactive design of shapes. (Curves defined by single polynomial equations can never be edited locally, because any two rational or polynomial curves that agree even on an infinitesimally small arc have to agree everywhere). Whether using polynomial or rational functions, placing a bound on the degree of the polynomials involved will serve as a way of controlling the complexity of the shape being modeled. To this end we will represent our curves with a group of equations, each controlling an arc of the overall curve and joined smoothly to the the next arc at each endpoint. The piecewise polynomial/rational shapes are usually referred to as *splines*.

1.3 Polynomials of What Degree?

Now that we have determined that we are going to represent our curves as groups of equations we need to consider the degree of the polynomials we will use. As mentioned before, using polynomials of higher degree will allow us to better model the curve we have in mind, but it has the drawback of being more difficult to work with. For most applications where we are modeling curves and surfaces, polynomials of fairly low degree seem to be sufficient. In this class we will work mainly with degrees 1, 2, and 3.

Note: When we say that a polynomial $f(t) = a_n t^n + \ldots + a_1 t + a_0$ is n-ic—or for n from 1 to 3, when we say that f is *affine*, *quadratic*, or *cubic*—we do *not* require that $a_n \neq 0$. In other words, an n-ic function is a function of degree *at most* n, not necessarily of degree precisely n.

2 A Closer Look at Parametric Polynomials

Given a *parametric* representation of some cubic curve by the two polynomials

$$\begin{aligned} x(t) &= x_3t^3 + x_2t^2 + x_1t + x_0, \\ y(t) &= y_3t^3 + y_2t^2 + y_0t + y_1, \end{aligned}$$

it is often difficult to visualize what the curve looks like from the values of the coefficients x_i and y_i. We will now examine parametric polynomials, up to cubics, and discuss their geometric shapes. These will form the repertory from which we'll select arcs for building our spline curves.

2.1 Affine and Quadratic Polynomials

Suppose we have polynomials $x = x(t)$ and $y = y(t)$, where our curve is $F(t) = \langle x(t), y(t) \rangle$. For affine (degree at most 1) polynomials the equations $x(t)$ and $y(t)$ have the form

$$\begin{aligned} x(t) &= x_1t + x_0, \\ y(t) &= y_1t + y_0. \end{aligned}$$

In this case we clearly have a parametric description of a straight line, a curve that we are all familiar with.

Now, suppose our polynomials are of degree 2. We have

$$\begin{aligned} x(t) &= x_2t^2 + x_1t + x_0, \\ y(t) &= y_2t^2 + y_1t + y_0. \end{aligned}$$

If we eliminate t among these two equations, we will get a quadratic polynomial equation between x and y. But it is not true that, with polynomials of this form, we can get any conic. In fact, the only conics that arise in this way are the parabolas.

To see this, consider the point at infinity on this curve. As $t \to \infty$, in either the positive or negative direction, the ratio $x(t)/y(t)$ tends to x_2/y_2. Therefore our curve intersects the line at infinity at a single point—the one whose homogeneous coordinates are $[0; x_2, y_2]$. Now ellipses never intersect the line at infinity at all, while hyperbolas intersect the line at infinity at two points (the points where their two asymptotes meet the line at infinity). The only possibility left is therefore a parabola.

We can use a coordinate transformation to make this more clear. Write new parameters u and v as

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} y_2 & -x_2 \\ x_2 & y_2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

This transformation rotates the axes in such a way that the curve goes to infinity in the vertical (v) direction. In fact, this transformation kills the quadratic term of the u component, thus making that coordinate linear in u. This linearization produces a curve of the form $v = au^2 + bu + c$, which is just the standard parabola that we are familiar with from high school math.

The coordinate transformation in the last paragraph makes sense only if at least one of the two quadratic coefficients x_2 and y_2 is nonzero. When $x_2 = y_2 = 0$, then our quadratic curve is actually also affine, and is hence a straight line.

2.2 Classification of Cubic Polynomials

Cubic polynomial curves have the general form

$$F(t) = (x_3t^3 + x_2t^2 + x_1t + x_0, y_3t^3 + y_2t^2 + y_1t + y_0).$$

It will again be easier to understand what happens by rotating the cubic so that it goes to infinity pointing "straight up". As before, as long either x_3 or y_3 is nonzero, this can be achieved by the rotation (and scaling)

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} y_3 & -x_3 \\ x_3 & y_3 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

After this transformation the cubic has the form

$$F(t) = (u(t), v(t)) = (u_2t^2 + u_1t + u_0, v_3t^3 + v_2t^2 + v_1t + v_0).$$

Notice that first coordinate is now quadratic in t rather than cubic.

It turns out that some of the crucial quantities for classifying the shapes of parametric cubic polynomials are the three determinant-like quantities

$$\begin{aligned} \kappa_{12} &= x_1y_2 - x_2y_1, \\ \kappa_{23} &= x_2y_3 - x_3y_2, \\ \kappa_{31} &= x_3y_1 - x_1y_3. \end{aligned}$$

Of these κ_{23} and κ_{31} have already appeared: we have $u_2 = \kappa_{23}$ and $u_1 = -\kappa_{31}$.

We consider first the situation when $\kappa_{23} = u_2 = 0$, but $\kappa_{31} = -u_1 \neq 0$. In this case, the degree of u as a function of t is precisely 1. We can therefore reparameterize to treat u as the time parameter, instead of t. That is, our curve is just the graph of a function of the form $v = p(u)$, where p is some degree three polynomial. (Note that $\kappa_{31} \neq 0$ implies that at least one of x_3 and y_3 must be nonzero, so $v_3 = x_3^2 + y_3^2 > 0$.) These cubics are called *S-shaped*, for obvious reasons. As the figure shows, these curves have a single inflection point, called the *flex* of the cubic, around which the entire curve is symmetric. The figure shows three situations, depending on whether at the flex the v coordinate is increasing (moving in the same direction as the cubic in general), standing still, or decreasing.

We suppose next that both κ_{23} and κ_{31} are zero, but that κ_{12} is nonzero. The algebraic identities

$$\begin{aligned} x_1\kappa_{23} + x_2\kappa_{31} + x_3\kappa_{12} &= 0 \\ y_1\kappa_{23} + y_2\kappa_{31} + y_3\kappa_{12} &= 0 \end{aligned}$$

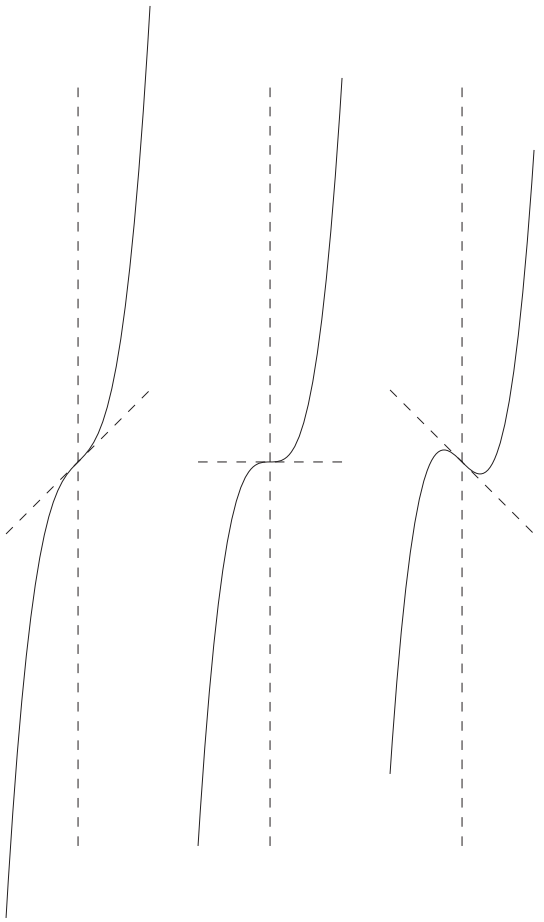

Figure 1: S-shaped Cubics

then allow us to conclude that $x_3 = y_3 = 0$, so our cubic is actually quadratic—in fact, a parabola.

If $\kappa_{12} = \kappa_{31} = \kappa_{23} = 0$, then all of the points of the cubic lie along a straight line; we call such cubics *straight*. But note that x_3 and y_3 may be nonzero in this case; that is, we may have a cubic parameterization of that straight line. Or the parameterization may have degree lower than 3.

Now, if $u_2 = \kappa_{23} \neq 0$, our u coordinate traces a parabola. Assume without loss of generality that $u_2 > 0$. Then there is some minimum u value beyond which our curve does not go. The *summit* is the point on our curve F corresponding to the t attaining this minimum u value. The function $u(t)$ is symmetric about the summit time. It follows that the lines joining points on F corresponding to symmetric times about the summit are all vertical, and their midpoints form a straight line, called the *median* of the cubic.

The shape of these cubic curves can be further described based on the sign of the first derivative of v at the summit. It turns out this sign is the sign of the quantity $3\kappa_{31}^2 - 4\kappa_{12}\kappa_{23}$. If v is increasing at the summit, we get the so-called *humpy* cubics (see figure 2).

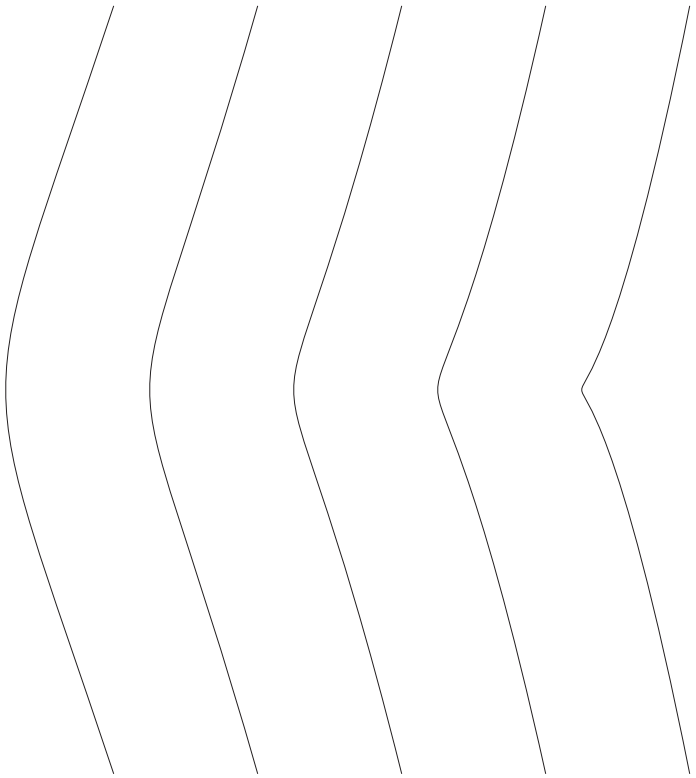

Figure 2: Humpy Cubics

On a humpy cubic, there will always be two points at which the curvature vanishes. These points are again called *inflection points* or *flexes*. The location of the flexes and the summit fully defines a humpy cubic.

If v is decreasing at the vertex—corresponding to a backwards or retrograde motion of the v coordinate, we get the *loopy* cubics (see figure 3). Loopy cubics have one self-intersection. This point is called the *crunode* of the cubic and lies on the median. A loopy cubic has no (real) flexes.

If $v'(t) = 0$ at the vertex (so $3\kappa_{31}^2 - 4\kappa_{12}\kappa_{23} = 0$), our curve is *pointy* (see figure 4). The vertex of the curve is then a *cusp*. Pointy curves can be thought of as the boundary form as we go from humpy curves to loopy curves. The hump gets narrower and more peaked until it turns into a point (the cusp), and then gets even more narrow so that the two sides of the hump continue through each other, forming the loop.

Like pointy cubics, S-shaped cubics are also a boundary case, but it's not the boundary between humpy and loopy, this time: It's the boundary between humpy and humpy. Figure 5 shows a sequence of five cubics. The first two are leftward-pointing humps whose humps have been sheared upwards. The last two are right-pointing humps whose humps have been sheared downwards. The middle cubic in the sequence is S-shaped. So S-shaped cubics occur as the boundary between humpy cubics whose hump is to the left and those whose hump is to the right.

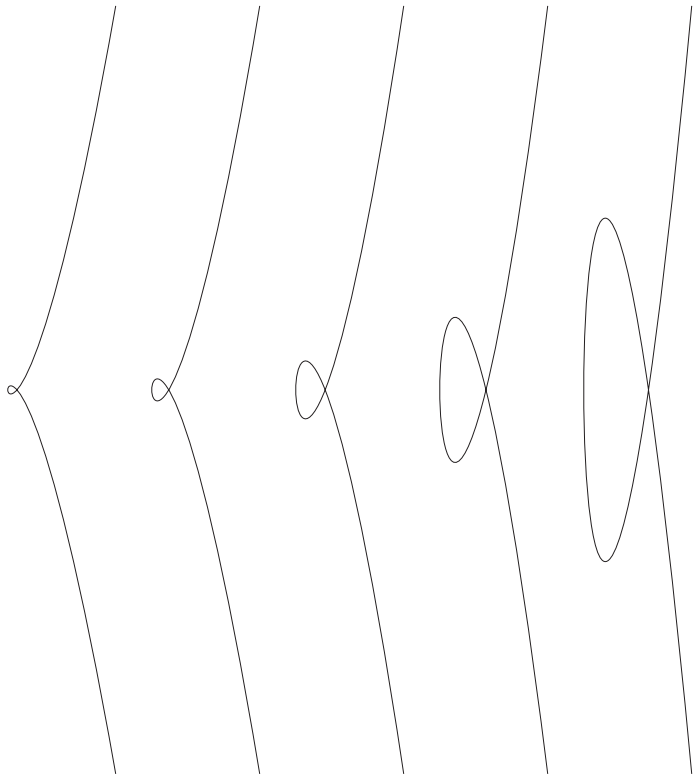

Figure 3: Loopy Cubics

It turns out that any planar parametric polynomial cubic that isn't straight—that is, that doesn't lie entirely along a single line—is equivalent to one of the five shapes we discussed: humpy, loopy, pointy, S-shaped, or parabolic. To be precise, through an affine reparametrization of the t axis and an affine map of the the plane in which the cubic is embedded, each cubic can be converted into one of the following standard forms:

- the *standard humpy cubic*, the curve $H(r) := (r^2, r^3 + r)$;
- the *standard loopy cubic*, the curve $L(r) := (r^2, r^3 - r)$;
- the *standard pointy cubic*, the semi-cubical parabola $P(r) := (r^2, r^3)$;
- the *standard S-shaped cubic*, the cubical parabola $S(r) := (r, r^3)$.
- the *standard parabola*, the quadratic normal curve $Q(r) := (r, r^2)$.

The classification tree in figure 6 shows how the quantities κ_{ij} determine which of these cases our cubic falls in. Incidentally, it should be fairly obvious that all these six cases are affinely distinct, as properties like self-intersection, the number of cusps, flexes, etc. are preserved by any affine map.

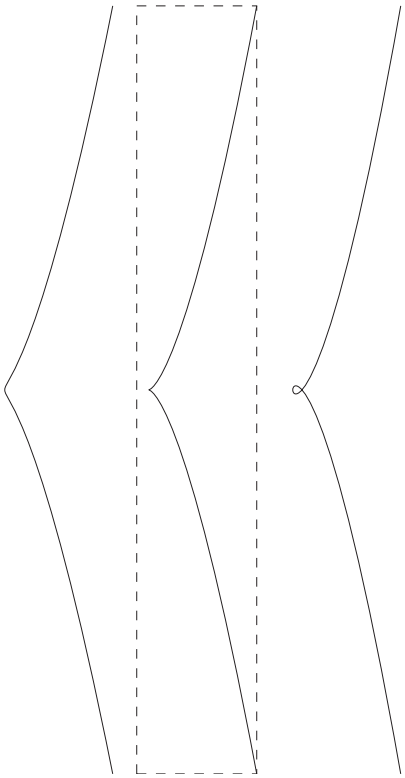

Figure 4: Humpy to loopy through pointy

3 Specifying Cubics

It is obviously not very user friendly to require a user to specify a piece of a cubic curve by giving its parametrization equations and endpoints. Let's look for a better way. Observe that if the user wants to specify a straight line segment, an obvious way to do so is by giving the endpoints of the segment. This gives 4 degrees of freedom, exactly right for specifying a straight segment. Similarly, we might expect to describe the six degrees of freedom of a parabolic arc by specifying three points, and the eight degrees of freedom of a cubic arc by specifying four points. But which points?

The straight line specification allows you to construct the other points on the segment by *interpolation*, using a convex combination of the endpoints. To carry this notion to higher degree curves, we introduce the idea of *polarization*, or *blossoming*.

3.1 Blossoming

We will transform a cubic $F(t)$ into another function $f(t_1,t_2,t_3)$ such that

- $f(t,t,t) = F(t)$.
- f is a symmetric function of t_1,t_2, and t_3.

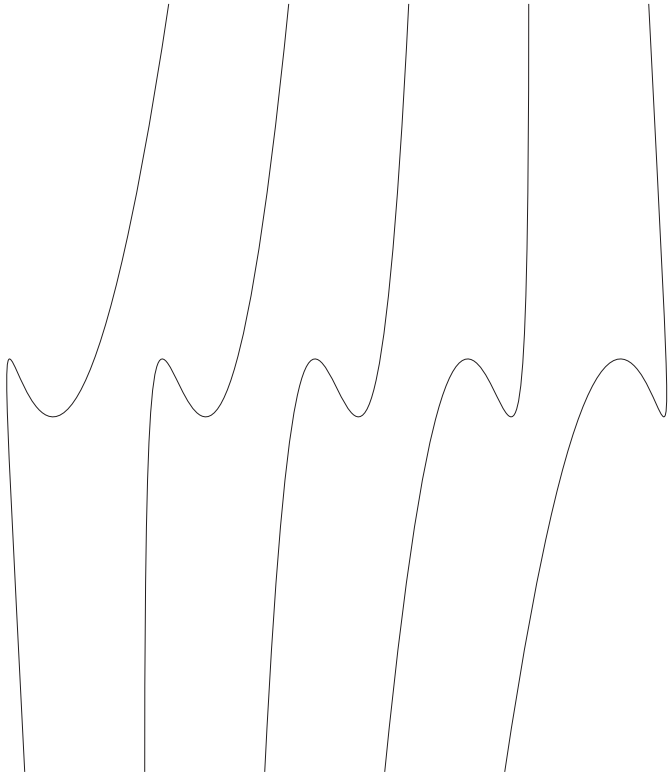

Figure 5: Humpy to humpy through S-shaped

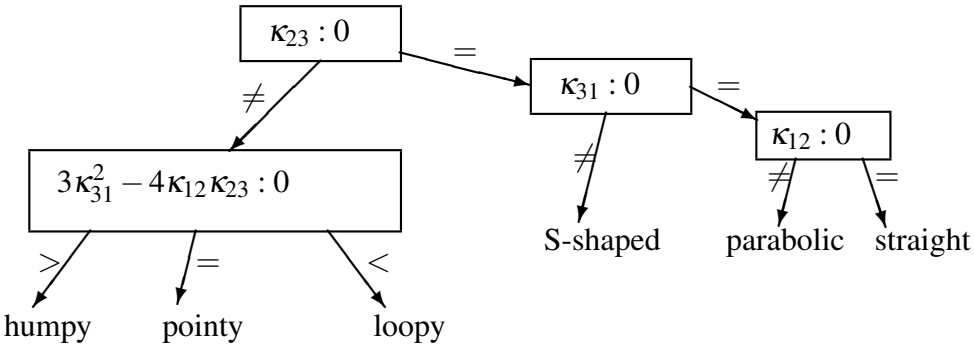

Figure 6: The classification tree for planar cubics

- f is affine in each t_i separately.

This transformation can in fact be applied to polynomials of arbitrary degree d to get a function f of arity d satisfying the appropriate generalizations of the above properties. It turns out the for any degree d polynomial F there always exists a unique function f satisfying these properties.

Let's build up to the cubic case in stages. If F is affine, then our function f is just $f(t) = F(t)$. We can specify f by specifying $f(r)$ and $f(s)$ and interpolating the other points. If F is

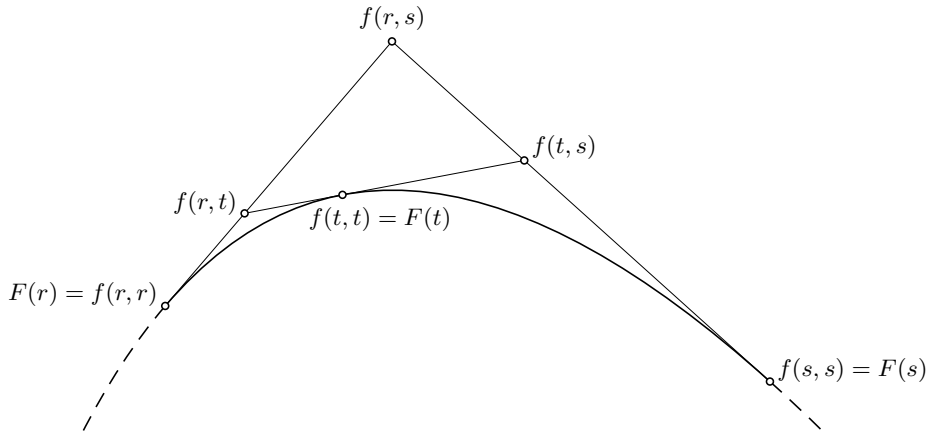

Figure 7: The de Casteljau Algorithm in the case $n = 2$

quadratic, it turns out that the arc $F(r) \ldots F(s)$ can be completely specified by giving the points $f(r,r)$, $f(s,s)$, and some "intermediate" point such as $f(r,s)$. We can find the rest of the curve by "interpolating" these three points appropriately. How can we find f? Well, it can be built by inspection. If

$$F(t) = at^2 + bt + c,$$

a function f satisfying the requirements is just

$$f(t_1,t_2) = at_1t_2 + b(t_1+t_2)/2 + c.$$

To compute a value $F(t) = f(t,t)$ given the *Bézier control points* $f(r,r)$, $f(r,s)$, and $f(s,s)$, we can perform the following computation, using the fact that f is multiaffine (see Figure 7).

- from $f(r,r)$ and $f(r,s)$, interpolate $f(r,t)$.
- from $f(r,s)$ and $f(s,s)$, interpolate $f(t,s) = f(s,t)$.
- from $f(r,t)$ and $f(s,t)$, interpolate $f(t,t) = F(t)$.

Similarly, in the cubic case we can specify a curve by specifying the Bézier control points $f(r,r,r)$, $f(r,r,s)$, $f(r,s,s)$, and $f(s,s,s)$. Note that the other "natural points" such as $f(s,r,s)$ are in fact the same points as the ones we specified, by symmetry. We can then use the same interpolation technique to construct arbitrary $F(t) = f(t,t,t)$ simply by adding one more interpolation stage (see Figure 8, where $r = 0$ and $s = 1$).

The points $f(\{\begin{smallmatrix} r \\ s \end{smallmatrix}\}, \{\begin{smallmatrix} r \\ s \end{smallmatrix}\}, \{\begin{smallmatrix} r \\ s \end{smallmatrix}\})$ (where $\{\}$ denotes choosing r or s) are the *Bezier control points* of the curve F. The define the vertices of a *control polygon*. The interpolation to find $F(t)$ is called the *deCasteljau method* and is vary famous in CAGD. Notice that the intermediate interpolation points contain the Bézier control points of the sub-arcs of the curve. This makes it very easy to manipulate them in the same way, for example by subdividing further. It can be shown that in the limit the control polygon converges the the curve—in fact cubic arcs are

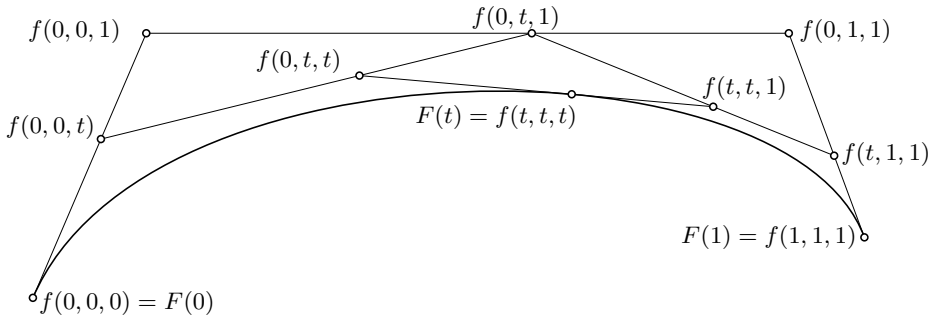

Figure 8: The de Casteljau Algorithm manipulating polar values

often drawn just be repeated subdivision until the corresponding arcs are so small that they can approximated by short line segments.

An interesting point is that since the the spline is defined by linear interpolation, the control points of an affinely transformed spline are just the affine transforms of the original control points.

There is another view of the Bézier control points. Given the four control points $P_0 = F(0)$, $P_1 = f(0,0,1)$, $P_2 = f(0,1,1)$, and $P_3 = F(1)$, it turns out that that the interpolations to produce the value of the curve at time t yield the point

$$F(t) = [(1-t)^3]P_0 + [3(1-t)^2 t]P_1 + [3(1-t)t^2]P_2 + [t^3]P_3.$$

The polynomials in brackets are the *Bernstein basis polynomials*. Observe that they are merely the summands in the expansion of $((1-t)+t)^3$. Thus, the coefficients of the above combination are always non-negative and sum to 1. This shows that the points on the spline are all convex combinations of the points P_i. Therefore, the spline is completely contained in the convex hull of its endpoints. This is a very useful property for many algorithms.