Parametric Curves

These are manifold point sets, just like lines.

Indeed, they are just bent and stretched straight lines. Other types of curve exist, and need to be handled, but they seldom need to be designed. In practice we approximate them by parametric curves.

The bounded form is always assumed. The curve is a map from the interval [0..1] to a point set.

Representations

Whichever basis we use, the form of the equation is the same:

$$P(t) := \sum A_i f_i(t)$$

where the A_i are the **coefficients** used to represent the curve segment and the $f_i(t)$ are the **basis functions** which decide what the map is between the coefficients and the curve itself.

In all useful cases the coefficients are a collection of points and displacements. Some of the forms have just a sequence of points as their representation.

The basis functions are Real to Real functions, and for an equation of the above form to fit the rules for computing points, we have to have some restriction on the basis set. This is:-

> There must be a subset of the i for which the sum of the corresponding basis functions is identically equal to 1. The coefficients of these are points.
>
> The coefficients of any other basis functions are displacements.

There are many possible sets of basis functions fulfilling these conditions.

Theorem : Coordinate System Invariance

If a curve is defined by the above equation, it has coordinate system invariance.

Given: a curve of the form $P(t) = \sum A_i f_i(t) + \sum B_j g_j(t)$, where $\sum f_i(t) = 1$ and so all the A_i are points and the B_j are displacements.

To Prove: that transformation of the coefficients A_i and B_j leads to transformation of every point $P(t)$ by the same transformation.

Construction: Each point is transformed by $A'_i = O + R.A_i$ and every displacement is transformed by $B'_j = R.B_j$.

Proof:

The result of transforming all the coefficients is

$$\begin{aligned} P'(t) &= \sum A'_i f_i(t) + sum B'_j g_j(t) \\ &= \sum [O + R.A_i] f_i(t) + \sum R.B_j g_j(t) \\ &= O \sum f_i(t) + \sum R.A_i f_i(t) + \sum R.B_j g_j(t) \\ &= O + R.[\sum A_i f_i(t) + \sum B_j g_j(t)] \\ &= O + R.P(t) \end{aligned}$$

■

Thus any curve with this kind of equation has the desirable property of coordinate system invariance under linear transformations. Note that this includes affine transformations as well as solid body ones.

For reasons of economy and efficiency we are interested in primarily two families of sets of basis functions:-

Those which span the space of **polynomials**
Those which span the space of **piecewise polynomials**

Polynomial Curves

Polynomials can be of different **degree**s, and for any given degree there are alternative bases all of which span the polynomials of that degree.

Monomials

$$P(t) = Q + tT + t^2U + t^3V + \ldots$$

The coefficients are all displacements except Q. They are related to the derivatives of the curve at the start point.

This can be thought of as a kind of Taylor series. The higher degree coefficients are not particularly easy to choose when designing a curve.

Hermite

This is a definition in terms of both end points and the derivatives there[1]. It is most natural for even degrees. A good example is the cubic

$$P(t) = P(0)f_0(t) + P(1)f_1(t) + P'(0)g_0(t) + P'(1)g_1(t)$$

where

$$\begin{aligned}
f_0(t) &= 1 - 3t^2 + 2t^3 \\
f_1(t) &= 1 - f_0(t) = 3t^2 - 2t^3 \\
g_0(t) &= t(t-1)^2 \\
g_1(t) &= t^2(t-1)
\end{aligned}$$

The major problem when using this form for cubics is that the derivative controls are displacements, whereas the end directions, which they control, are merely directions. It is not obvious what lengths these displacements should have, although rules of thumb (make them as long as the chord between the endpoints) work well, and practitioners soon found tricks that they could play to get specific effects.

Lagrange

These give polynomials interpolating equally spaced[2] data values.

$$f_i(t) = \frac{\Pi_{j \neq i}(t - t_j)}{\Pi_{j \neq i}(t_i - t_j)}$$

Chebyshev

These are not seriously used in CAGD.

Bernstein

Curves defined using these functions are usually referred to in CAGD as **Bézier** curves, after Pierre Bézier who popularised them. Before that we used Hermite cubics under the name of **Ferguson** curves.

1

2

The Bernstein polynomials of degree n are usually expressed in terms of t and $s = 1 - t$ and are given by the terms in the binomial expansion of

$$(s + t)^n$$

Because $(s+t) = 1$, the whole set of basis functions sums to one, and so all the coefficients are points. There is a natural sequence of basis functions, because they have their maxima equally spaced along the interval [0..1], and we usually join the control points in this sequence into a polygon called the **control polygon**.

Thus for cubics we have

$$\begin{aligned} f_0(t) &= s^3 \\ f_1(t) &= 3s^2t \\ f_2(t) &= 3st^2 \\ f_3(t) &= t^3 \end{aligned}$$

The generic form of the i^{th} coefficient is

$$f_i(t) = \frac{n!}{i!(n-i)!} s^{n-i} t^i$$

An important property, the **hodograph** property, is that the first derivative is given by another Bernstein polynomial whose coefficients are just n times the first differences of the original control points.

Immediate Exercise Confirm this derivative property.

This property was theoretically extremely important, because it helped Bernstein to prove the Weierstrass Approximation Theorem, that any $C\infty$ function can be uniformly approximated over an interval arbitrarily closely by a polynomial of high enough degree, together with all its derivatives.

Bernstein used his polynomials, with coefficients taken from the curve to be approximated, and showed that as the degree rose, the polynomial so defined would get ever closer to the required curve. The hodograph property, coupled with the fact that n times the first differences gives a better and better approximation to the derivative, shows the convergence of the derivatives as well.

The hodograph property also has the immediate corollary that the i^{th} derivative at an end is proportional to the i^{th} difference of the control points at that end. This is used to help join adjacent Bezier segments together with continuity. If we require that two segments have i derivatives continuous at their join, we merely have to impose the conditions that the first i differences of their control polygons match across the junction.

Piecewise polynomials.

We are interested in the piecewise polynomials which have maximal continuity at the places where the polynomial pieces meet. These are called **Splines**, and there are again two important variants tabulated below. Because of the maximal continuity constraint the number of control points is essentially one per polynomial piece, plus a few extra at the ends depending on the degree.

Interpolating Splines

These are the equivalent of Lagrange Polynomials, in that the coefficients are a set of points through which the curve passes. They are, however, much less sensitive than the Lagrange Polynomials. This is related to the fact that for small perturbations the cubic spline is the curve of minimum bending energy which interpolates the data points.

B-Splines

These are analogous to Bernstein Polynomials, but have two interesting properties

The basis functions have **finite support**. This means that each function is zero outside a relatively narrow interval.

Apart from at the ends of the curve, the basis functions are all the same shape. They are all translates (shifts in abscissa) of the same function, which is called **the equal interval B-spline**[3] of the given degree.

They also have the property that the derivative is another spline over the B-spline basis one degree lower, whose coefficients are just *degree* times the first differences of the original control points.

The basis functions for the low degree B-splines can be determined by applying the conditions for continuity of Bezier segments. For example, the sequence of Bezier coefficients for the cubic B-spline are

—0 0 0 1 — 1 2 4 4 — 4 4 2 1 — 1 0 0 0 —

all divided by 6.

For low degree interpolating splines, we can determine the basis functions by solving the collocation equations to fit B-splines to cardinal data with a unit value at one point and zero at all the others. The B-splines can then be summed to give the required basis function.

Comparison of polynomial and piecewise curves

We essentially compare Lagrange polynomials with Interpolating splines, and Bernstein polynomials with B-splines.

[3]

Design convenience

For small numbers of control points there is very little difference. For large numbers, the Lagrange polynomial becomes far too sensitive to small movements of the middle control points, while the Bernstein polynomial becomes too insensitive.

For long curves, the ability to make a change to a limited region of the curve, leaving the rest invariant, is a strong argument for the B-splines.

Efficiency of interrogation

Again, for small numbers of control points there is little difference. For large numbers, however, the B-spline shows a clear advantage in that to evaluate the curve at a particular value of t requires only a relatively small number of control points to be accessed.

Although it is possible for a B-spline to be evaluated by determining the polynomial piece that the query point lies at, it is far better, for relatively low degrees, to use a procedure called the **Cox-deBoor** recurrence.

Properties of Basis functions in general

A property important for editing a curve is that each basis function is essentially the derivative of the curve with respect to its coefficient. This is a trivial result of the linear definition. What is means is that basis functions which have clear maxima give good local editing properties. The part of the curve parametrically near that maximum will lie close to the corresponding control point, and they will then be influenced most strongly when the point is moved.

The overall sensitivity of a curve to noisy control points can be assessed by the norms

$\sqrt{\sum_i f_i^2(t)}$ is a function which shows the sensitivity of the curve to uncorrelated noise on the coefficients.

$\sum_i |f_i(t)|$ is a function which shows the worst case effect on the curve at each point by control point perturbations malevolently chosen to give most effect at that point.

The bad sensitivity of the Lagrange interpolant is displayed quite clearly by these measures.