**cs 514, lecture 28apr00: knot insertion**

Recall B-spline property (xi). *If $\widehat{\mathbf{t}}$ is obtained from $\mathbf{t}$ by insertion of the point $x$, and, correspondingly,*

$$\widehat{B}_{jk} := B_{j,k,\widehat{\mathbf{t}}}\,,$$

*then*

$$\sum_j a_j B_{jk} = f = \sum_j \widehat{a}_j \widehat{B}_{jk}, \tag{1}$$

*with*

$$\widehat{a}_j = \widehat{\omega}_{jk}(x) a_j + (1 - \widehat{\omega}_{jk}(x)) a_{j-1},$$

*where*

$$\widehat{\omega}_{jk}(x) := \max\{0, \min\{1, \omega_{jk}(x)\}\}.$$

Recall also the *control polygon* for $f = \sum_j a_j B_{j,k,\mathbf{t}}$ as the broken line with vertices the *control point*s

$$P_j = (t^*_{jk}, a_j) \in \mathrm{I\!R}^2,$$

i.e., the coefficients of the spline *curve*

$$x \mapsto (x, f(x)) = \sum_j P_j B_{j,k,\mathbf{t}}(x)$$

that is the graph of $f$. It follows that

$$\sum_j P_j B_{j,k,\mathbf{t}} = \sum_j \widehat{P}_j \widehat{B}_{jk},$$

with

$$\widehat{P}_j = \widehat{\omega}_{jk}(x) P_j + (1 - \widehat{\omega}_{jk}(x)) P_{j-1}.$$

This shows that the refined control polygon has its vertices on the control polygon we started with.

This is worth a picture. I'll make up a (cubic) spline in B-form consisting of just one cubic piece, pretty much like the interpolant to sin, i.e., with B-spline coefficients $(0, 32/9, -32/9, 0)$, except that I'll rescale the interval of interest to be $[0 \,.\,.\, 4]$. Then I'll insert the point 1 three times into the knot sequence, each time plotting the control polygon a little bit thicker. I also circle the final set of control points:

```
k = 4; sp = spmak(augknt([0 4],k),[0 32/9 -32/9 0]);
fnplt(sp), hold on
plot(aveknt(sp.knots,k),sp.coefs,'-r','linew',.5)
sp = fnrfn(sp,1); plot(aveknt(sp.knots,k),sp.coefs,'-r','linew',2)
sp = fnrfn(sp,1); plot(aveknt(sp.knots,k),sp.coefs,'-r','linew',4)
sp = fnrfn(sp,1); plot(aveknt(sp.knots,k),sp.coefs,'-w','linew',2)
plot(aveknt(sp.knots,k),sp.coefs,'ok'), hold off
```

Note that the refined control points lie exactly where they are supposed to lie: Since 1 is the quarter-point on the way from $0 = t_j$ to $4 = t_{j+k-1}$, each new $\widehat{P}_j$ lies at the quarter-point from $P_{j-1}$ to $P_j$.

Also, notice that the initial (and the final) slope of $f$ and its control polygon coincide (why??).

Also, the second time, only the first part of the control polygon changes.

Finally, on the third time, the polygon doesn't change; it only acquires one more vertex, exactly at the point where the control polygon touches the spline. This is not surprising because, by this time, the number 1 is a three-fold knot for this cubic spline, hence there is only one cubic B-spline not zero at that point and its coefficient therefore necessarily equals f(1).

 ©2000 Carl de Boor

In general, *if* $\#t_{j+1} = k-1$, *then* $B_{jk}$ *is the only* $B_{i,k,\mathbf{t}}$ *not zero at* $x := t_{j+1}$, *hence necessarily has the value 1 there and, correspondingly,* $a_j = f(x)$. In other words, we can use repeated knot insertion as a way to evaluate splines.

This fact together with the fact that the refined control polygon has its vertices on the original control polygon is at the basis of B-spline property (xii): *If* $\sum_j a_j B_j$ *and* $\tau_1 < \cdots < \tau_r$ *are such that* $f(\tau_{i-1})f(\tau_i) < 0$ *for all* $i$, *then there exist indices* $j(1) < \cdots < j(r)$ *so that*

$$f(\tau_i)a_{j(i)}B_{j(i)}(\tau_i) > 0, \quad i = 1{:}r.$$

This property is called **variation diminution** since it says, offhand, that the spline cannot oscillate any more times than its control polygon.

But it says much more than that. Since, by B-spline property (v), any linear polynomial is its own control polygon, it says that a spline cannot oscillate across any linear polynomial any more times than its control polygon does.

In particular, if the control polygon is monotone, so is the spline. If the control polygon is convex, so is the spline. This property is also called **shape-preserving**, and the wide use of splines in CAGD is almost entirely due to this property. For it makes it easy to construct a smooth spline curve of a desired shape simply by sketching that shape as a control polygon and then use the corresponding spline.

As an example, take the corners of the unit square as control points, i.e., consider the spline curve

$$x \mapsto \sum_{j=1}^{5} P_j B_{j,k,\mathbf{t}},$$

with

$$P = \begin{bmatrix} 1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 \end{bmatrix},$$

hence $\mathbf{t}$ any knot sequence of length $5+k$. To be definite, take $k=4$, hence $\mathbf{t}$ should have length 9. If we also want the first and last knot have multiplicity $k=4$, that leaves us with just one interior knot. E.g.,

```
t = [0 0 0 0 1 2 2 2 2];
a = [1 -1 -1  1 1; ...
     1  1 -1 -1 1];
sp = spmak(t,a);
fnplt(sp), hold on, plot(a(1,:),a(2,:),'-o'), hold off, axis equal, axis off
```

Because of the end knots of multiplicity $k$, the curve begins at $P_1$ and ends at $P_5 (= P_1)$.

But don't be misled by the terms *control* point and *control* polygon. As my performance at the end of class showed, it is not all that easy to produce a curve of some desired shape by playing with the control points (contrary what some books might want you to believe). The *construction* of a spline with desired properties (i.e., the *reconstruction* of a spline from information about it) is best done by direct interpolation or other fitting of a suitable spline model to specific data, our next (and final) topic.

 ©2000 Carl de Boor