cs 514, lecture 8apr02: information about f from its B-spline coefficients

Let

$$f := \sum_j a_j B_{jk}.$$

If $k > 1$, then, from the recurrence relations,

$$\begin{aligned} f &= \sum_j a_j (\omega_{jk} B_{j,k-1} + (1 - \omega_{j+1,k}) B_{j+1,k-1}) \\ &= \sum_j \left(a_j \omega_{jk} + a_{j-1}(1 - \omega_{j,k})\right) B_{j,k-1}. \end{aligned}$$

Hence, with

$$a_j^{[i+1]} := \begin{cases} a_j, & i = 0; \\ a_j^{[i]} \omega_{j,k-i+1} + a_{j-1}^{[i]} (1 - \omega_{j,k-i+1}) = \dfrac{(\cdot - t_j) a_j^{[i]} + (t_{j+k-i} - \cdot) a_{j-1}^{[i]}}{t_{j+k-i} - t_j}, & i > 0, \end{cases}$$

we get

$$f = \sum_j a_j^{[i]} B_{j,k-i+1}, \qquad i = 1{:}k.$$

In particular,

$$f = \sum_j a_j^{[k]} B_{j,1},$$

with each $a_j^{[k]}$ a polynomial of degree $< k$, i.e., a polynomial of **order** k,

$$a_j^{[k]} \in \Pi_{<k}.$$

Consider some specific sequences $a = (a_j : j)$.

1. $a = \delta_j$, i.e., $f = B_{jk}$. Then, by induction, $a_j^{[i]}, \ldots, a_{j+i-1}^{[i]}$ are the only nonzero entries in $a^{[i]}$. In particular, B_{jk} has its support in the interval $[t_j \mathinner{..} t_{j+k})$.

2. $a_j = 1$, all j. Then also $a_j^{[i]} = 1$ for all j, therefore

$$\sum_j B_{jk} = \sum_j B_{j1} = 1,$$

showing that $(B_{jk} : j)$ forms a (positive and local) partition of unity.

3. This example (actually not done in class) is the prettiest:

$$a_j = (t_{j+1} - \tau) \cdots (t_{j+k-1} - \tau) =: \psi_{jk}(\tau), \quad \forall j,$$

with τ arbitrary. Then

$$\begin{aligned} a_j^{[2]} &= \psi_{jk}(\tau)\omega_{jk} + \psi_{j-1,k}(\tau)(1 - \omega_{jk}) \\ &= \psi_{j,k-1}(\tau)\left((t_{j+k-1} - \tau)\omega_{jk} + (t_j - \tau)(1 - \omega_{jk})\right) \\ &= \psi_{j,k-1}(\tau)(\cdot - \tau). \end{aligned}$$

In other words,

$$\sum_j \psi_{jk}(\tau) B_{jk} = (\cdot - \tau) \sum_j \psi_{j,k-1}(\tau) B_{j,k-1} = \cdots = (\cdot - \tau)^{k-1} \sum_j \psi_{j,1}(\tau) B_{j,1} = (\cdot - \tau)^{k-1}.$$

This is **Marsden's identity**:

$$(\cdot - \tau)^{k-1} = \sum_j \psi_{jk}(\tau) B_{jk}.$$

From this, one gets a formula for writing any $p \in \Pi_{<k}$ as a weighted sum of the B_{jk}, as follows:

$$p = \sum_j \lambda_{jk} p\, B_{jk}, \qquad \forall p \in \Pi_{<k}, \tag{1}$$

with

$$\lambda_{jk} : f \mapsto \sum_{\nu=1}^{k} \frac{(-D)^{\nu-1}\psi_{jk}(\tau)}{(k-1)!} D^{k-\nu} f(\tau). \tag{2}$$

Take in the fact that this holds for an arbitrary τ. (In fact, it is easy to verify that, *for any* $f \in \Pi_{<k}$, $\lambda_{jk} f$ *is independent of* τ).

As a quick check, take for p a constant polynomial. Then all derivatives of p are zero everywhere, hence

$$\lambda_{jk} p = \frac{(-D)^{k-1}\psi_{jk}(\tau)}{(k-1)!} p(\tau),$$

and this equals $p(\tau)$ since $\psi_{jk}(\tau) = (-\tau)^{k-1} + \text{l.o.t.}$, hence $(-D)^{k-1}\psi_{jk} = (k-1)!$. We conclude that

$$\lambda_{jk} p = p(\tau), \qquad p \in \Pi_0.$$

A more interesting case occurs when p is a linear polynomial, say $p = \ell \in \Pi_1$. Now $D^i p(\tau) = 0$ for any $i > 1$. Therefore,

$$\lambda_{jk} \ell = \ell(\tau) + \frac{(-D)^{k-2}\psi_{jk}(\tau)}{(k-1)!} D\ell(\tau).$$

Now, since ψ_{jk} is a polynomial of exact degree $k-1$, its $(k-2)$nd derivative is a polynomial of exact degree 1, hence has exactly one zero. This zero turns out to be the point

$$t^*_{jk} := (t_{j+1} + \cdots + t_{j+k-1})/(k-1).$$

In other words

$$\lambda_{jk} \ell = \ell(t^*_{jk}), \qquad \forall \ell \in \Pi_1,$$

hence, by (1),

$$\ell = \sum_j \ell(t^*_{jk}) B_{jk}, \qquad \forall \ell \in \Pi_1. \tag{3}$$

It turns out that the formula (1) holds not just for $p \in \Pi_{<k}$, but for every $p = \sum_j a_j B_{jk}$ with arbitrary coefficient sequence $(a_j : j)$, provided only that the τ appearing in the definition (2) of the linear functional λ_{jk} be, more precisely, some point τ_j in the support of B_{jk}, i.e., from the interval $(t_j \mathinner{..} t_{j+k})$. With that choice, we have

$$\lambda_{ik} B_{jk} = \delta_{ij}, \quad \forall i, j.$$

For this reason, the λ_{ik} are called the **dual functionals** (for the corresponding B-spline sequence).

In particular, assuming that $B_j k \neq 0$, i.e., $t_j < t_{j+k}$, for all j, $(B_{jk} : j)$ is linearly independent, hence a basis for its span,

$$\$_{k,\mathbf{t}} := \operatorname{span}(B_{j,k,\mathbf{t}} : j).$$

It is for this reason that their creator, I. J. Schoenberg, gave them the letter 'B', as an acronym for 'Basis' or 'basic'.

$\$_{k,\mathbf{t}}$ comprises the **splines of order k with knot sequence $\mathbf{t}$**. Its elements are **piecewise polynomial, of order k with breaks at the t_i**, meaning that, on each interval $(t_i \,..\, t_{i+1})$, each $f \in \$_{k,\mathbf{t}}$ is (or, agrees with) some polynomial of degree $< k$. In addition, each such f satisfies at least $k - \#t_i$ smoothness conditions across the breakpoint t_i, with

$$\#t_i := \#\{j : t_j = t_i\}$$

the *multiplicity* of t_i in the knot sequence $\mathbf{t}$. These two properties *characterize* the space $\$_{k,\mathbf{t}}$.

For example, earlier in the course, you considered $B_3 := B_1 * B_1 * B_1 = B_1 * B_2$, and know that B_3 is piecewise polynomial of order 3 with breaks at $0, 1, 2, 3$ and in C^1. Hence, it is an element of $\$_{k,\mathbb{Z}}$, therefore writeable as

$$B_3 = \sum_j (\lambda_{j3} B_3)\, B(\cdot | j, j+1, j+2, j+3).$$

Now, for $j \neq 0$, we can choose $\tau_j \in (j \,..\, j+3)$ to lie outside the interval $[0 \,..\, 3]$, hence get $\lambda_{j3} B_3 = 0$ for $j \neq 0$. What about $j = 0$? Well, we know that $B_2(x) = x$ on $[0 \,..\, 1]$, hence

$$B_3(x) = \int B_1(x-y) B_2(y)\, \mathrm{d}y = \int_{x-1}^{x} B_2(y)\, \mathrm{d}y = \int_0^x y\, \mathrm{d}y = x^2/2$$

for $0 \le x \le 1$. But, with $\tau_0 = 0^+$, we compute

$$\lambda_{03} B_3 = \psi_{03}(0)/2! = (1-0)(2-0)/2 = 1.$$

Hence,

$$B_3 = B_{0,3,\mathbb{Z}} = B(\cdot | 0, 1, 2, 3).$$

Next: What information about $f = \sum_j a_j B_{jk}$ is 'easily' obtained from its B-spline coefficients $(a_j : j)$?

1. evaluation: . To compute $f(x)$, (i) determine j such that $t_j \le x < t_{j+1}$, then use the recurrence to compute $a_j^{[k]}(x)$ from $a_{j-k+1}, \ldots, a_j$ via $a_{j-k+i}^{[i]}(x), \ldots, a_j^{[i]}(x)$, $i = 2{:}k-1$. Explicitly, it means some like this: Initialize $b := (a_{j+1-k}, \ldots, a_j)$; then

```
 for i=2:k
   for r=k:-1:i
      b(r) = ((x-t(j-k+r))*b(r) + (t(j+r-i+1)-x)*b(r-1))/...
             ((x-t(j-k+r))       + (t(j+r-i+1)-x)        );
   end
end
```

After this, $b(k)$ contains the value of f at x. Note that the index for the inner loop runs down rather than up (why?). To be sure, a preferable implementation would compute the quantities `x-t(i)` and `t(k+i)-x`, `i=1:k`, needed here outside the double loop, in which case computation of the denominator is no more costly than in its simpler form `-t(j-k+r) + t(j+r-i+1)`. The present form is preferable for rounding-error control.

2. Differentiation The derivative of a spline $f = \sum_j a_j B_{jk}$ is a spline of one order lower, and its coefficients are difference quotients of the coefficients of the spline itself:

$$D\big(\sum_j a_j B_{jk}\big) = \sum_j \frac{a_j - a_{j-1}}{(t_{j+k-1} - t_j)/(k-1)} B_{j,k-1}.$$

To be sure, if, e.g., $t_{j+k-1} = t_j$, then that quotient multiplying $B_{j,k-1}$ is not defined. However, in that case, $B_{j,k-1}$ is the zero function, and we don't care.

Note that, in this case, $\#t_j \ge k$, i.e., f itself may have a jump discontinuity across t_j, and is not even differentiable at t_j. In effect, we ignore that, by taking the derivative here piecewise-polynomial style, i.e., for each polynomial piece separately.

As a consequence, $\int_x^y (Df)(s)\,\mathrm{d}s$ will equal $f(y) - f(x)$ in general only if the spline f is continuous on the interval $[x \mathinner{..} y]$, for example if $\#t_i < k$ for all $t_i \in (x \mathinner{..} y)$.

3. Good condition aka stable basis We already saw that, for $t_j \le x < t_{j+1}$, the value $f(x) = \sum_{i=j-k+1}^{j} a_i B_{ik}(x)$ is a *convex* combination of the k coefficients a_{j-k+i}, $i = 1{:}k$. In particular, the value $f(x)$ must lie between the smallest and the largest of these k coefficients. On the other hand, at least for modest k, none of these k coefficients can be too far from the value $f(x)$. Precisely,

$$|a_i| \le D_{k,\infty} \| \sum_j a_j B_{jk} \|_{[t_{i+1} .. t_{i+k-1}]},$$

with $D_{k,\infty} \approx 2^{k-3/2}$.

This makes (B_{jk}) a **stable** basis (or Riesz basis) in the uniform norm in the sense that

$$(1/D_{k,\infty}) \|a\|_\infty \le \| \sum_j a_j B_{jk} \|_\infty \le \|a\|_\infty.$$

But the B-spline basis has this property even *locally*.

4. Control polygon and refinability (aka subdivision) The close connection between the value $f(x)$ of $f = \sum_j a_j B_{jk}$ and the 'nearby' coefficients $(a_i : B_{ik}(x) \ne 0)$ is made visible in CAGD by considering the *curve*

$$x \mapsto (x, f(x)) = (\sum_j t^*_{jk} B_{jk}, \sum_j a_j B_{jk}) =: \sum_j P_j B_{jk}$$

(note the use of (3) here), with

$$P_j = P_{j,k,\mathbf{t}} f := (t^*_{jk}, a_j) \in \mathbb{R}^2$$

called the **control point**s, and the broken line connecting these control points, and denoted here by

$$C_{k,\mathbf{t}} f,$$

called the **control polygon**.

This nomenclature arose in CAGD (:= Computer-Aided Geometric Design), where one considers, more generally, **spline curves**, i.e., curves of the form $x \mapsto \sum_{jk} P_j B_{jk}(x)$ with P_j arbitrary vectors in the plane (or even in 3-space or higher dimensions) and, correspondingly, its control polygon, i.e., the piecewise linear curve $x \mapsto \sum_j P_j B_{jk}(x)$.

The control polygon provides a rough outline or caricature of the spline itself. At the same time, by the stability of the B-spline basis, for modest order k, this control polygon cannot be too far from the curve itself. Sticking with a spline *function*, i.e., our scalar-valued spline $f = \sum_j a_j B_{jk}$, one infers directly from the dual functionals that

$$a_j = f(t^*_{jk}) + O((t_{j+k-1} - t_{j+1})^2 \|D^2 f\|_{[t_{j+1} .. t_{j+k-1}]}).$$

This implies that the control polygon is close to the spline itself when the **mesh spacing**

$$|\mathbf{t}| := \sup_i (t_{i+1} - t_i)$$

is sufficiently small.

E.g., try out this simple example, in which a cubic spline is generated by interpolation, then plotted, along with its control polygon:

```
x = sort(rand(1,21))*4*pi; k = 4; sp = spapi(k,x,sin(x)./(.3+x));
fnplt(sp)
hold on, plot(aveknt(fnbrk(sp,'knots'),k), fnbrk(sp,'coef') ,'k'), hold off
```


What if the mesh spacing is not small? Well, we can make it smaller by *refining* the knot sequence. After all, if $\mathbf{t}$ is a subsequence of $\widehat{\mathbf{t}}$, then $\$_{k,\mathbf{t}}$ is a subset of $\$_{k,\widehat{\mathbf{t}}}$, i.e.,

$$\mathbf{t} \subset \widehat{\mathbf{t}} \quad \Longrightarrow \quad \$_{k,\mathbf{t}} \subset \$_{k,\widehat{\mathbf{t}}},$$

hence, in that case, each $f \in \$_{k,\mathbf{t}}$ is also uniquely writeable as a weighted sum of the $\widehat{B}_{jk} := B_{j,k,\widehat{\mathbf{t}}}$:

$$\sum_j a_j B_{jk} = f = \sum_j \widehat{a}_j \widehat{B}_{jk}. \tag{4}$$

E.g., continue the example:

```
sp = fnrfn(sp,aveknt(x,3));
hold on, plot(aveknt(fnbrk(sp,'knots'),k), fnbrk(sp,'coef') ,'r'), hold off
```

The formula for the $\widehat{a}_j$ can be quite involved. However, we can obtain any **refinement** $\widehat{\mathbf{t}}$ of $\mathbf{t}$ in a sequence of steps, each of which consists of adding just *one* knot. Hence, it is sufficient to know the formula for $\widehat{a}_j$ in the special case that $\widehat{\mathbf{t}}$ is obtained from $\mathbf{t}$ by the insertion of just one additional knot.

knot insertion. *If $\widehat{\mathbf{t}}$ is obtained from $\mathbf{t}$ by insertion of the point s, then (4) holds with*

$$\widehat{a}_j = \widehat{\omega}_{jk}(s) a_j + (1 - \widehat{\omega}_{jk}(s)) a_{j-1}, \tag{5}$$

where

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

This knot-insertion, or refinement, process has the following very striking geometric interpretation. Applying it to the B-spline coefficients $(t^*_{jk} : j)$ of $x \mapsto \sum_j t^*_{jk} B_{jk}(x)$, we find that also

$$\widehat{t}^*_{jk} = \widehat{\omega}_{jk}(s) t^*_{jk} + (1 - \widehat{\omega}_{jk}(s)) t^*_{j-1,k},$$

hence

$$\widehat{P}_j = \widehat{\omega}_{jk}(s) P_j + (1 - \widehat{\omega}_{jk}(s)) P_{j-1}. \tag{6}$$

This says that *the control polygon $C_{k,\widehat{\mathbf{t}}} f$ has all its vertices (i.e., control points) $\widehat{P}_j$ on the control polygon $C_{k,\mathbf{t}} f$*, either because it is one of the P_j, or else it lies on the straight line between P_{j-1} and P_j. In other words, the finer control polygon interpolates the rougher one. This means that we can also visualize the insertion process as *corner cutting* (draw the picture in a simple case, e.g., for $k = 3$; why doesn't the picture for $k = 2$ give us any insight?), and corner-cutting only smoothes things out.

By repeatedly inserting knots, we can obtain in this way a control polygon for f arbitrarily close to f itself. But such repeated interpolation by broken lines can only decrease the number of crossings

- of the x-axis, hence f *can have a zero only near a zero of* $C_{k,\mathbf{t}} f$;
- in a particular direction of any line parallel to the x-axis, hence f *must be mononote near where* $C_{k,\mathbf{t}} f$ *is mononote*;
- in any particular direction of any straight line, hence f *must be convex (concave) near where* $C_{k,\mathbf{t}} f$ *is convex (concave).*

See the pictures on the next page.

This *shape preservation* explains the popularity of **Schoenberg's variation-diminishing spline operator**:

$$Vg := \sum_j g(t^*_{jk})\, B_{jk}.$$

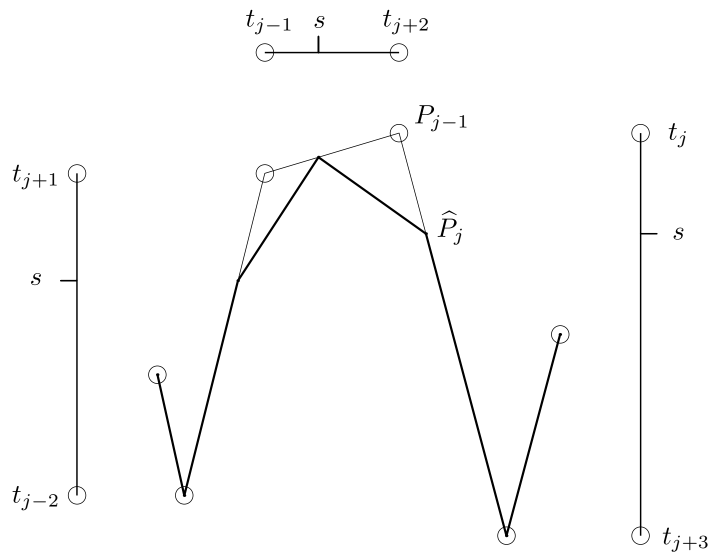

Figure 7 Insertion of $s = 2$ into the knot sequence $\mathbf{t} = (0,0,0,0,\mathbf{1},\mathbf{3},5,5,5,5)$, with $k = 4$.

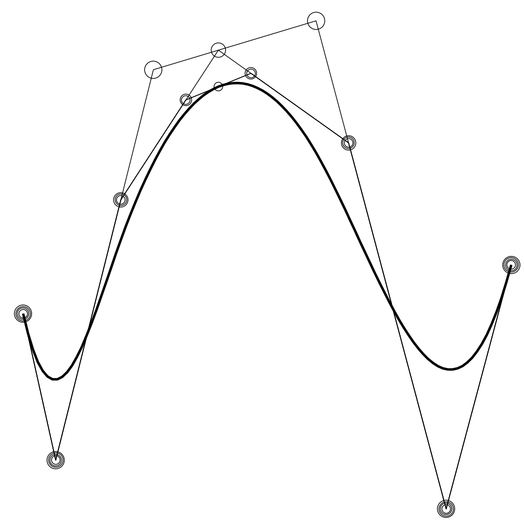

Figure 8 Three-fold insertion of the same knot provides a point on the graph of a cubic spline.

