cs 514, lecture 24apr00: an aside: the ppform

Here is a standard method for the evaluation of a *cardinal* cubic spline, i.e., a cubic spline with the uniform knot sequence $\mathbb{Z}$.

Assume that we are interested in the value of $f = \sum_j a_j B_j$ at some $x \in [0 \mathinner{..} 1)$. Then

$$f(x) = \sum_{i=-3}^{0} B_i(x) a_i = \sum_{i=-3}^{0} p_i(x) a_i,$$

with p_i the polynomial that agrees with B_i on the interval $(0 \mathinner{..} 1)$. Let

$$p_i =: \sum_{r=1}^{4} ()^{4-r} C(r, 4+i), \quad i = -3{:}0.$$

Then C is the change-of-basis matrix (or transition matrix) for going from the basis $[p_{-3}, \ldots, p_0]$ for Π_3 to the power basis $[()^3, \ldots, ()^0]$, i.e.,

$$C = [()^3, \ldots, ()^0]^{-1} [p_{-3}, \ldots, p_0],$$

and

$$\begin{aligned} f(x) &= [p_{-3}(x), \ldots, p_0(x)](a_{-3}, \ldots, a_0) \\ &= [x^3, \ldots, x^0]\, C\, (a_{-3}, \ldots, a_0). \end{aligned} \tag{1}$$

In particular, the vector $\mathbf{b} := C\,(a_{-3}, \ldots, a_0)$ provides the coefficients in the **power form** for the cubic polynomial that agrees with f on the interval $(0 \mathinner{..} 1)$. With this vector in hand, we would now compute $f(x)$, not by forming $[x^3, \ldots, x^0]$ and then multiplying $\mathbf{b}$ by it, but by using Nested Multiplication (remember Nested Multiplication??), i.e., by computing

$$f(x) = ((x * b_1 + b_2) * x + b_3) * x + b_4.$$

Once $\mathbf{b}$ is known, then this calculation takes only 3 adds and 3 multiplies, considerably less than working directly with the recurrence relation. (Part of this advantage is thrown away if one follows the CAGD habit of explicitly forming the vector $[x^3, \ldots, x^0]$.) But this gain comes at the cost of computing $\mathbf{b}$ from $\mathbf{a}$, hence pays off only if evaluation at more than just one x is needed.

In `Matlab`, this calculation can be carried out simultaneously for a whole vector or even a matrix x.

If we need $f(x)$ for $x \in [j \mathinner{..} j+1)$ for some $j \neq 0$, then the needed changes are quite minor: the relevant coefficient segment becomes $(a_{j-3}, \ldots, a_j)$ and we would replace x by $x - j$, i.e., we would, in effect, write f on $[j \mathinner{..} j+1)$ in **local power form**:

$$f(x) = \sum_{r=1}^{k} (x-j)^{k-r} b_r.$$

In particular, the matrix C remains unchanged, but this is a special feature of using the *uniform* knot sequence $\mathbb{Z}$.

For an *arbitrary* knot sequence $\mathbf{t}$ and arbitrary order k, the evaluation of $f = \sum_i a_i B_i$ at some $x \in [t_j \mathinner{..} t_{j+1})$ still can be handled this way, i.e., by writing

$$\begin{aligned} f(x) &= [p_{j-k+1}(x), \ldots, p_j(x)](a_{j-k+1}, \ldots, a_j] \\ &= [(x-t_j)^{k-1}, \ldots, (x-t_j)^0]\, C\, (a_{j-k+1}, \ldots, a_j). \end{aligned} \tag{2}$$

But now the change-of-basis matrix C depends on j and $\mathbf{t}$ as well as on k. There are several ways in use (and knot insertion is one of them) to carry out such a conversion, from the **B-form** for f to the corresponding **ppform**. The latter describes f in terms of its **breaks** $(\xi_i : i)$, i.e., the largest strictly increasing subsequence of the knot sequence $\mathbf{t}$, and the coefficient matrix c for its **local power form**, i.e.,

$$f(x) = \sum_{r=1}^{k} (x-\xi_j)^{k-r} c_{j,r}, \quad \xi_j \le x < \xi_{j+1}, \quad \forall j.$$

E.g., the command `ppB = fn2fm(spmak(0:4,1),'pp')` provides the ppform of the cardinal cubic B-spline, and `fnbrk(ppB,'coe')` provides the entries for the matrix C appearing in (1), though differently ordered.