MAT4170

Exercises for Spline Methods

August Femtehjell

Spring 2025

Contents

Abstract

This document contains my solutions to the exercises for the course MAT4170–Spline Methods, taught at the University of Oslo in the spring of 2025. The code for everything, as well as this document, can be found at my GitHub repository: `https://github.com/augustfe/MAT4170`.

1 Bernstein-Bézier polynomials

Exercise 1.1. It is sometimes necessary to convert a polynomial in BB form to monomial form. Consider a quadratic BB polynomial,

$$p(x) = c_0(1-x)^2 + 2c_1x(1-x) + c_2x^2.$$

Express p in the monomial form

$$p(x) = a_0 + a_1x + a_2x^2.$$

Solution 1.1. Rather than using the explicit formula for conversion, we can just expand the coefficients and collect terms.

$$\begin{aligned} p(x) &= c_0(1-x)^2 + 2c_1x(1-x) + c_2x^2 \\ &= c_0(1-2x+x^2) + 2c_1(x-x^2) + c_2x^2 \\ &= c_0 - 2c_0x + c_0x^2 + 2c_1x - 2c_1x^2 + c_2x^2 \\ &= c_0 + (-2c_0+2c_1)x + (c_0-2c_1+c_2)x^2. \end{aligned}$$

Exercise 1.2. Consider a polynomial $p(x)$ of degree $\leq d$, for arbitrary d. Show that if

$$p(x) = \sum_{j=0}^{d} a_jx^j = \sum_{i=0}^{d} c_iB_i^d(x),$$

then

$$a_j = \binom{d}{j}\Delta^j c_0.$$

Hint: Use a Taylor approximation to p to show that $a_j = p^{(j)}(0)/j!$.

Solution 1.2. We have that

$$p(x) = \sum_{j=0}^{d} a_jx^j = \sum_{i=0}^{d} c_iB_i^d(x).$$

By the Taylor approximation, we have that

$$p(x) = p(x+0) = \sum_{j=0}^{d} \frac{p^{(j)}(0)}{j!}x^j.$$

We thus have that

$$a_j = \frac{p^{(j)}(0)}{j!}.$$

By properties of the Bézier curves, we have that

$$p^{(j)}(x) = \frac{d!}{(d-j)!} \sum_{i=0}^{d-j} \Delta^j c_i B_i^{d-j}(x),$$

and specifically for $x = 0$,

$$p^{(j)}(0) = \frac{d!}{(d-j)!} \Delta^j c_0.$$

Combining these results, we have that

$$a_j = \frac{p^{(j)}(0)}{j!} = \frac{d!}{(d-j)!j!} \Delta^j c_0 = \binom{d}{j} \Delta^j c_0,$$

as we wanted to show.

Exercise 1.3. We might also want to convert a polynomial from monomial form to BB form. Using Lemma 1.2, show that in the notation of the previous question,

$$c_i = \frac{i!}{d!} \sum_{j=0}^{i} \frac{(d-j)!}{(i-j)!} a_j.$$

Solution 1.3. Lemma 1.2 states that for $j = 0, 1, \dots, d$,

$$x^j = \frac{(d-j)!}{d!} \sum_{i=j}^{d} \frac{i!}{(i-j)!} B_i^d(x).$$

We have that

$$\sum_{j=0}^{d} a_j x^j = \sum_{i=0}^{d} c_i B_i^d(x)$$

$$\sum_{j=0}^{d} a_j \left[\frac{(d-j)!}{d!} \sum_{i=j}^{d} \frac{i!}{(i-j)!} B_i^d(x) \right] = \sum_{i=0}^{d} c_i B_i^d(x)$$

As we have $i \geq j$, we can reorder the summation to the form $j \leq i$, by using

$$\sum_{j=0}^{d} \sum_{i=j}^{d} (\dots) = \sum_{i=0}^{d} \sum_{j=0}^{i} (\dots).$$

This gives us

$$\sum_{i=0}^{d}\left[\sum_{j=0}^{i} a_j \frac{(d-j)!}{d!}\frac{i!}{(i-j)!}\right] B_i^d(x) = \sum_{i=0}^{d} c_i B_i^d(x).$$

Which by isolating the coefficients, gives us

$$c_i = \frac{i!}{d!}\sum_{j=0}^{i}\frac{(d-j)!}{(i-j)!}a_j,$$

as we wanted to show.

Exercise 1.4. Implement the de Casteljau algorithm for cubic Bézier curves in Matlab or Python (or some other programming language), taking repeated convex combinations. Choose a sequence of four control points and plot both the control polygon and the Bézier curve, like those in Figure 1.3.

Solution 1.4. The de Casteljau algorithm uses recursion to compute the value of a point along a Bézier curve by the following formula:

1. Initialize by setting $c_i^0 = c_i$ for $i = 0, 1, \ldots, d$.
2. Then, for each $r = 1, 2, \ldots, d$, let
$$c_i^r = (1-x)c_i^{r-1} + xc_{i+1}^{r-1}, \quad i = 0, 1, \ldots, d-r.$$
3. The last value c_0^d is the value of the Bézier curve at x.

This is implemented using Jax in Python in `de_casteljau.py`, and the result is shown in Figure 1, bearing a striking resemblance to the figure in the book.

Exercise 1.5. Show that the graph, $g(x) = (x, p(x))$ of the BB polynomial p in (1.6) is a Bézier curve in $\mathbb{R}^2$, with control points (ξ_i, c_i), $i = 0, 1, \ldots, d$, where $\xi_i = i/d$. *Hint:* Express x as a linear combination of $B_0^d(x), \ldots, B_d^d(x)$.

Solution 1.5. We can again utilize Lemma 1.2 to express x as a linear combination of the Bernstein polynomials. We have that, writing $x = x^1$ for clarity,

$$x^1 = \frac{(d-1)!}{d!}\sum_{i=1}^{d}\frac{i!}{(i-1)!}B_i^d(x) = \sum_{i=1}^{d}\frac{i}{d}B_i^d(x) = \sum_{i=0}^{d}\frac{i}{d}B_i^d(x) = \sum_{i=0}^{d}\xi_i B_i^d(x).$$

We can now express the graph of p as a Bézier curve in $\mathbb{R}^2$ by

$$g(x) = (x, p(x)) = \left(\sum_{i=0}^{d}\xi_i B_i^d(x), \sum_{i=0}^{d} c_i B_i^d(x)\right) = \sum_{i=0}^{d}(\xi_i, c_i)B_i^d(x) = \sum_{i=0}^{d}\boldsymbol{c}_i B_i^d(x),$$

where $\boldsymbol{c}_i = (\xi_i, c_i)$ are the control points of the Bézier curve.

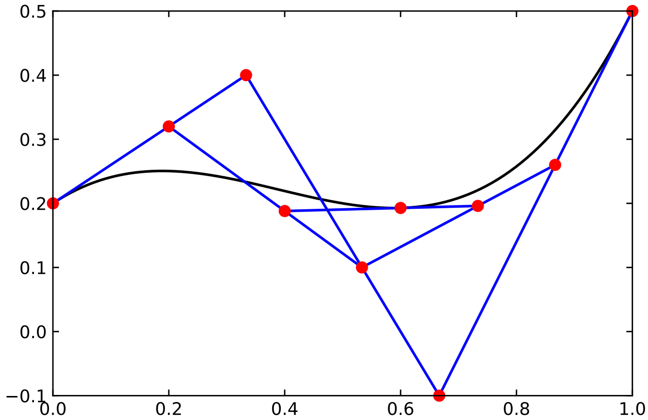

Figure 1: The de Casteljau algorithm applied to a cubic Bézier curve, with control points $(0.2, 0.4, -0.1, 0.5)$, illustrated at the point $x = 0.6$.

Exercise 1.6. Show that the tangent vector $\boldsymbol{p}'(x)$ of the Bézier curve in (1.6) lies in the convex cone of the vectors $\Delta\boldsymbol{c}_i$, i.e., in

$$\operatorname{cone}(\Delta\boldsymbol{c}_0, \ldots, \Delta\boldsymbol{c}_{d-1}) = \left\{ \sum_{i=0}^{d-1} \mu_i \Delta\boldsymbol{c}_i : \mu_1, \ldots, \mu_{d-1} \geq 0 \right\}.$$

Solution 1.6. The derivative (or perhaps *gradient* is the correct term) of the Bézier curve $\boldsymbol{p}(x)$ is given by

$$\boldsymbol{p}'(x) = d \sum_{i=0}^{d-1} (\boldsymbol{c}_{i+1} - \boldsymbol{c}_i) B_i^{d-1}(x) = d \sum_{i=0}^{d-1} \Delta\boldsymbol{c}_i B_i^{d-1}(x).$$

As $B_i^{d-1}(x) \geq 0$ for $x \in [0,1]$, we can set $\mu_i = dB_i^{d-1}(x)$, and we have that

$$\boldsymbol{p}'(x) = \sum_{i=0}^{d-1} \mu_i \Delta\boldsymbol{c}_i \in \operatorname{cone}(\Delta\boldsymbol{c}_0, \ldots, \Delta\boldsymbol{c}_{d-1}),$$

as we wanted to show.

Exercise 1.7. Show that the first derivative of p in (1.6) can be expressed (and computed) as

$$p'(x) = d(c_1^{d-1} - c_0^{d-1}),$$

where c_1^{d-1}, c_0^{d-1} are the points of order $d-1$ in de Casteljau's algorithm (1.10).

Solution 1.7. We have that

$$p(x) = c_0^d = (1-x)c_0^{d-1} + xc_1^{d-1},$$

and thus by differentiating with respect to x, we have that

$$p'(x) = c_1^{d-1} - c_0^{d-1}.$$

This tells us that we cannot be as naive as this, as c_0^d is actually a function of x, and not simply a constant.

What we might instead need to note is that

$$c_i^r = \sum_{j=0}^{r} c_{i+j} B_j^r(x),$$

and combining this with the fact that

$$\left(B_i^d\right)'(x) = d\left(B_{i-1}^{d-1} - B_i^{d-1}\right)(x),$$

we have that

$$\begin{aligned} p'(x) &= d\sum_{i=0}^{d-1}\left(c_{i+1} - c_i\right)B_i^{d-1}(x) = d\left[\sum_{i=0}^{d-1} c_{i+1}B_i^{d-1}(x) - \sum_{i=0}^{d-1} c_i B_i^{d-1}(x)\right] \\ &= d(c_1^{d-1} - c_0^{d-1}), \end{aligned}$$

as we wanted to show.

Exercise 1.8. Show that the Bernstein basis polynomial $B_i^d(x)$ has only one maximum in $[0, 1]$, namely at $x = i/d$.

Solution 1.8. We do this by firstly computing the derivative of $B_i^d(x)$, which is given by

$$\left(B_i^d\right)'(x) = d\left(B_{i-1}^{d-1}(x) - B_i^{d-1}(x)\right).$$

A maximum or minimum of a function occurs where the derivative is zero, so we set

$$\begin{aligned} B_{i-1}^{d-1}(x) &= B_i^{d-1}(x) \\ \frac{(d-1)!}{(i-1)!(d-i)!}x^{i-1}(1-x)^{d-i} &= \frac{(d-1)!}{i!(d-1-i)!}x^i(1-x)^{d-i-1} \\ \frac{\cancel{(d-1)!}i!(d-1-i)!}{(i-1)!(d-i)!\cancel{(d-1)!}}\cancel{x^{i-1}}(1-x)^{\cancel{d-i}} &= x^{\cancel{i}}\cancel{(1-x)^{d-i-1}} \\ \frac{i}{d-i}(1-x) &= x \\ i - ix &= dx - ix \\ x &= \frac{i}{d}. \end{aligned}$$

We have thus shown that the Bernstein basis polynomials only have one extremal point.

We can use the second derivative to test if this is a maximum or a minimum, however we can instead note that $B_i^d(x)$ is a non-negative polynomial, which is only zero at either $x = 0$ or $x = 1$, and thus $x = i/d$ must be a maximum.

Exercise 1.9. Give a proof of the forward difference formula, (1.15).

Solution 1.9. The forward difference formula (1.15) is given by

$$\Delta^r c_0 = \sum_{i=0}^{r} \binom{r}{i} (-1)^{r-i} c_i.$$

The forward difference operator is defined by the recursion

$$\Delta^r c_i = \Delta^{r-1} c_{i+1} - \Delta^{r-1} c_i,$$

where $\Delta^0 c_i = c_i$.

We prove this by induction on r. For the base case $r = 1$, we have that

$$\Delta c_0 = c_1 - c_0 = \binom{1}{0} (-1)^{1-0} c_0 + \binom{1}{1} (-1)^{1-1} c_1.$$

For the induction step, we assume that the formula holds for $r = k$, and show that it holds for $r = k + 1$. We have that

$$\begin{aligned}
&\Delta^{k+1} c_0 \\
&= \Delta^k c_1 - \Delta^k c_0 \\
&= \sum_{i=0}^{k} \binom{k}{i} (-1)^{k-i} c_{i+1} - \sum_{i=0}^{k} \binom{k}{i} (-1)^{k-i} c_i \\
&= \sum_{i=1}^{k+1} \binom{k}{i-1} (-1)^{k-i+1} c_i - \sum_{i=0}^{k} \binom{k}{i} (-1)^{k-i} c_i \\
&= \binom{k}{k} c_{k+1} + \sum_{i=1}^{k} \left(\binom{k}{i-1} (-1)^{k-i+1} - \binom{k}{i} (-1)^{k-i} \right) c_i - \binom{k}{0} (-1)^k c_0 \\
&= \binom{k+1}{k+1} c_{k+1} + \sum_{i=1}^{k} (-1)^{(k+1)-i} \left(\binom{k}{i-1} + \binom{k}{i} \right) c_i + \binom{k+1}{0} (-1)^{k+1} c_0 \\
&= \sum_{i=0}^{k+1} \binom{k+1}{i} (-1)^{k+1-i} c_i,
\end{aligned}$$

as we wanted to show.

Exercise 1.10. The Bernstein approximation to a function $f : [0, 1] \to \mathbb{R}$ of order d is the polynomial $g : [0, 1] \to \mathbb{R}$ defined by

$$g(x) = \sum_{i=0}^{d} f\left(\frac{i}{d}\right) B_i^d(x).$$

Show that if f is a polynomial of degree $m \leq d$, then g has degree m.

Solution 1.10. Let q be the polynomial defined by

$$q(x) = f(x) - g(x).$$

We have that f is a polynomial of degree $m \leq d$. As

$$q\left(\frac{i}{d}\right) = f\left(\frac{i}{d}\right) - g\left(\frac{i}{d}\right) = 0,$$

we have that q has $d+1$ roots, and thus q is either a polynomial of degree $d+1$, or $q = 0$. However, as q is the sum of two polynomials of degree m and d, respectively, we have that q is a polynomial of degree $\max(m, d)$. As $m \leq d$, we have that q is at most a polynomial of degree d, and thus $q = 0$. q being the zero polynomial implies that $g = f$, and thus g has degree m.

Exercise 1.11. Show that the length of the Bézier curve p in (1.9) is bounded by the length of its control polygon,

$$\operatorname{length}(p) \leq \sum_{i=0}^{d-1} \|\Delta \boldsymbol{c}_i\|.$$

Solution 1.11. The length of a curve is given by the integral of the norm of the derivative of the curve, i.e.,

$$\operatorname{length}(p) = \int_0^1 \|\boldsymbol{p}'(x)\| \, dx.$$

We have that

$$\|\boldsymbol{p}'(x)\| = \|d \sum_{i=0}^{d-1} \Delta \boldsymbol{c}_i B_i^{d-1}(x)\| = d\|\sum_{i=0}^{d-1} \Delta \boldsymbol{c}_i B_i^{d-1}(x)\| \leq d \sum_{i=0}^{d-1} \|\Delta \boldsymbol{c}_i\| B_i^{d-1}(x),$$

where we in the last inequality use the fact that $B_i^{d-1}(x)$ is a non-negative scalar for $x \in [0,1]$. We then have

$$\begin{aligned}
\operatorname{length}(p) &= \int_0^1 \|\boldsymbol{p}'(x)\|\,dx \\
&\leq \int_0^1 d\sum_{i=0}^{d-1} \|\Delta \boldsymbol{c}_i\| B_i^{d-1}(x)\,dx \\
&= \sum_{i=0}^{d-1} \|\Delta \boldsymbol{c}_i\| \int_0^1 dB_i^{d-1}(x)\,dx \\
&= \sum_{i=0}^{d-1} \|\Delta \boldsymbol{c}_i\|,
\end{aligned}$$

using the property $\int_0^1 B_i^{d-1}(x)\,dx = 1/d$ in the last step, showing the identity.

2 Splines in Bernstein-Bézier Form

Exercise 2.1. Prove equation (2.8).

Solution 2.1. Equation (2.8) states that the condition for C^2 continuity of s can be expressed as the two equations in (2.7), namely

$$e_0 = c_d \quad \text{and} \quad e_1 = (1-\alpha)c_d + \alpha c_{d-1}$$

where

$$\alpha = \frac{c-a}{b-a},$$

plus the equation

$$e_2 = (1-\alpha)^2 c_d + 2(1-\alpha)\alpha c_{d-1} + \alpha^2 c_{d-2}.$$

In the original theorem, we have that a spline s has C^r continuity, for $r = 0, 1, \ldots, d$, if and only if

$$\frac{1}{(b-a)^k}\Delta^k c_{d-k} = \frac{1}{(c-b)^k}\Delta^k e_0, \qquad k = 0, 1, \ldots, r.$$

For $r = 2$, we must then solve for $k = 0, 1, 2$. In the first case we simply have

$$\begin{aligned}
\frac{1}{(b-a)^0}\Delta^0 c_d &= \frac{1}{(c-b)^0}\Delta^0 e_0, \\
c_d &= e_0.
\end{aligned}$$

For $k = 1$ we have then have

$$\begin{aligned}
\frac{1}{(b-a)^1}\Delta^1 c_{d-1} &= \frac{1}{(c-b)^1}\Delta^1 e_0, \\
\frac{c_d - c_{d-1}}{b-a} &= \frac{e_1 - e_0}{c-b} = \frac{e_1 - c_d}{c-b}, \\
e_1 - c_d &= \frac{c-b}{b-a}(c_{d-1} - c_d), \\
e_1 &= (1-\alpha)c_d + \alpha c_{d-1}.
\end{aligned}$$

Finally, for $k = 2$ we have

$$\begin{aligned}
\frac{1}{(c-b)^2}\Delta^2 e_0, &= \frac{1}{(b-a)^2}\Delta^2 c_{d-2} \\
\frac{e_2 - 2e_1 + e_0}{(c-b)^2} &= \frac{c_d - 2c_{d-1} + c_{d-2}}{(b-a)^2}, \\
e_2 - 2\left((1-\alpha)c_d + \alpha c_{d-1}\right) + c_d &= \alpha^2\left(c_{d-2} - 2c_{d-1} + c_d\right) \\
e_2 &= \left(\alpha^2 - 2\alpha + 1\right)c_d + 2\left(\alpha - \alpha^2\right)c_{d-1} + \alpha^2 c_{d-2} \\
e_2 &= (1-\alpha)^2 c_d + 2(1-\alpha)\alpha c_{d-1} + \alpha^2 c_{d-2},
\end{aligned}$$

which is the desired result.

Exercise 2.2. Implement the de Casteljau algorithm for a planar Bézier curve of arbitrary degree d over a general interval $[a, b]$. Use the routine to make a program to plot the quadratic spline curve $\mathbf{s} : [0, 2] \to \mathbb{R}$, with pieces

$$\mathbf{p}(t) = \sum_{i=0}^{2} \mathbf{c}_i B_i^2(t), \qquad 0 \le t \le 1,$$
$$\mathbf{q}(t) = \sum_{i=0}^{2} \mathbf{d}_i B_i^2(t-1), \qquad 1 < t \le 2,$$

where $\mathbf{c}_0 = (-1, 1)$, $\mathbf{c}_1 = (-1, 0)$, $\mathbf{c}_2 = (0, 0)$, and $\mathbf{d}_0 = (0, 0)$, $\mathbf{d}_1 = (1, 0)$, $\mathbf{d}_2 = (2, 1)$.

Solution 2.2. The de Casteljau algorithm is implemented in `spline_cj.py`, and is practically identical to the one implemented in the previous section, due to the vectorization possibilities in `JAX`. The resulting figure is shown in Figure 2.

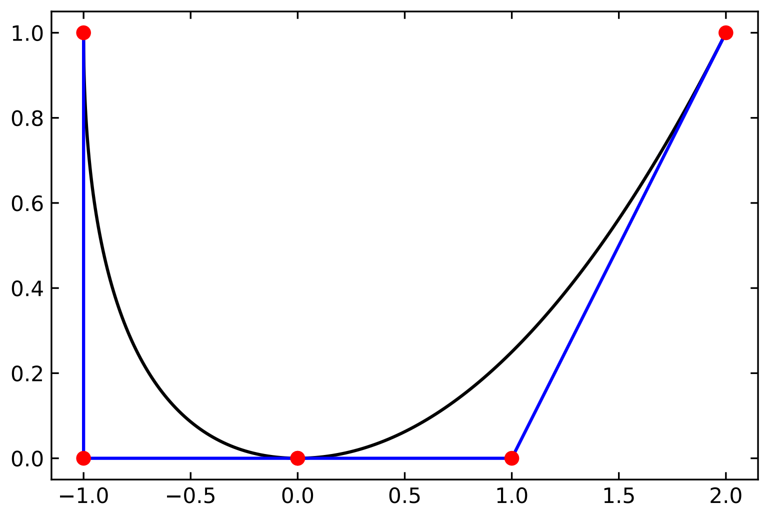

Figure 2: The quadratic spline curve $\mathbf{s} : [0, 2] \to \mathbb{R}$, with pieces $\mathbf{p}(t)$ and $\mathbf{q}(t)$.

Exercise 2.3. What is the order of continuity of $\mathbf{s}$ in Exercise 2.2 at the breakpoint $t = 1$?

Solution 2.3. We clearly have C^0 continuity at the breakpoint $t = 1$, as

$$\mathbf{c}_2 = (0, 0) = \mathbf{d}_0.$$

For C^1 continuity, we must have

$$\begin{gathered}\frac{\mathbf{c}_2 - \mathbf{c}_1}{1-0} = \frac{\mathbf{d}_1 - \mathbf{d}_0}{2-1} \\ (0,0) - (-1,0) = (1,0) - (0,0),\end{gathered}$$

which also holds. Finally for C^2 continuity, we must have

$$\begin{gathered}\frac{\mathbf{c}_2 - 2\mathbf{c}_1 + \mathbf{c}_0}{1^2} \stackrel{?}{=} \frac{\mathbf{d}_2 - 2\mathbf{d}_1 + \mathbf{d}_0}{1^2} \\ (0,0) - 2(-1,0) + (-1,1) = (1,1) \neq (0,1) = (2,1) - 2(1,0) + (0,0),\end{gathered}$$

which it seems like we do not have. Thus, the order of continuity of $\mathbf{s}$ at the breakpoint $t = 1$ is C^1.

Exercise 2.4. The curvature of a parametric curve $\mathbf{r}(t)$ in $\mathbb{R}^2$ can be expressed as

$$\kappa(t) = \frac{\mathbf{r}'(t) \times \mathbf{r}''(t)}{\|\mathbf{r}'(t)\|^3},$$

where $(a_1, a_2) \times (b_1, b_2) := a_1 b_2 - a_2 b_1$. What are the curvatures of $\mathbf{p}$ and $\mathbf{q}$ in Exercise 2.2 at the breakpoint $t = 1$? What can you say about the smoothness of $\mathbf{s}$?

Solution 2.4. We firstly begin by tabulating the derivatives of $\mathbf{p}$ and $\mathbf{q}$, based on the general formula

$$p^{(r)}(x) = \frac{d!}{(d-r)!} \frac{1}{h^r} \sum_{i=0}^{d-r} \Delta^r c_i B_i^{d-r}(\lambda).$$

In both cases here, we have $h = 1$, $d = 2$, and $\lambda = t$ for $\mathbf{p}$ and $\lambda = t - 1$ for $\mathbf{q}$. The derivatives for $\mathbf{p}$ are then

$$\begin{aligned}\mathbf{p}'(t) &= 2\left((\mathbf{c}_1 - \mathbf{c}_0)\, B_0^1(\lambda) + (\mathbf{c}_2 - \mathbf{c}_1)\, B_1^1(\lambda)\right) & \mathbf{p}''(t) &= 2\,(\mathbf{c}_2 - 2\mathbf{c}_1 + \mathbf{c}_0)\, B_0^0(\lambda) \\ &= 2\left((\mathbf{c}_1 - \mathbf{c}_0)(1-t) + (\mathbf{c}_2 - \mathbf{c}_1)\, t\right) & &= 2\,(1,1) \\ &= 2\left((\mathbf{c}_2 - 2\mathbf{c}_1 + \mathbf{c}_0)t + (\mathbf{c}_1 - \mathbf{c}_0)\right) & &= (2,2), \\ &= 2\left((1,1)t + (0,-1)\right) & & \\ &= (2t, 2t-2), & &\end{aligned}$$

while we for $\mathbf{q}$ have

$$\begin{aligned}\mathbf{q}'(t) &= 2\left((\mathbf{d}_1 - \mathbf{d}_0)\, B_0^1(\lambda) + (\mathbf{d}_2 - \mathbf{d}_1)\, B_1^1(\lambda)\right) & \mathbf{q}''(t) &= 2\,(\mathbf{d}_2 - 2\mathbf{d}_1 + \mathbf{d}_0)\, B_0^0(\lambda) \\ &= 2\left((\mathbf{d}_1 - \mathbf{d}_0)\, t + (\mathbf{d}_2 - \mathbf{d}_1)(1-t)\right) & &= 2\,(0,1) \\ &= 2\left((-\mathbf{d}_2 + 2\mathbf{d}_1 - \mathbf{d}_0)t + (\mathbf{d}_2 - \mathbf{d}_1)\right) & &= (0,2), \\ &= 2\left((0,-1)t + (1,1)\right) & & \\ &= (2, -2t+2), & &\end{aligned}$$

where we have used that $B_0^1(1-t) = t$ and $B_1^1(1-t) = 1-t$. The curvatures are then

$$\kappa_{\mathbf{p}}(t) = \frac{(2t, 2t-2) \times (2,2)}{\sqrt{(2t)^2 + (2t-2)^2}^3} = \frac{4}{(8t^2 - 8t + 4)^{3/2}},$$
$$\kappa_{\mathbf{q}}(t) = \frac{(2, -2t+2) \times (0,2)}{\sqrt{2^2 + (-2t+2)^2}^3} = \frac{4}{(4t^2 - 8t + 8)^{3/2}}.$$

These give, at the breakpoint $t = 1$, the curvatures

$$\kappa_{\mathbf{p}}(1) = \frac{4}{2^3} = \frac{1}{2} \quad \text{and} \quad \kappa_{\mathbf{q}}(1) = \frac{4}{2^3} = \frac{1}{2}.$$

As we see, the curvatures are equal at the breakpoint $t = 1$. I'm not sure what this means. . .

Exercise 2.5. Show that the minimization property of Theorem 2.9 also holds for the natural spline of Section 2.7.

Solution 2.5. The natural spline of Section 2.7 is defined as the spline g defined for $x_1 < x_2 < \cdots < x_m$ by the piecewise cubic polynomials g_i such that $g(x) = g_i(x)$ for $x \in [x_i, x_{i+1}]$. In addition, we have the conditions

$$\begin{aligned} g_i(x_i) &= y_i & \qquad & i = 1, 2, \ldots, m, \\ g''_{i-1}(x_i) &= g''_i(x_i) & & i = 2, \ldots, m-1. \end{aligned}$$

combined with the natural end conditions

$$g''_1(x_1) = 0 \quad \text{and} \quad g''_m(x_m) = 0.$$

To prove the minimization property of Theorem 2.9, let h be any C^2 function satisfying the interpolation conditions. Let $e = g - h$. Then $e \in C^2[x_1, x_m]$ and

$$e(x_i) = g(x_i) - h(x_i) = 0 \quad \text{and} \quad e''(x_1) = e''(x_m) = 0$$

for $i = 1, 2, \ldots, m$. Continuing as in the proof of Theorem 2.9, we have $h = g + e$ so

$$\begin{aligned} \int (h'')^2 &= \int (g'' + e'')^2 \\ &= \int (g'')^2 + 2\int g''e'' + \int (e'')^2 \\ &\geq \int (g'')^2 + 2\int g''e''. \end{aligned}$$

Let $\phi = g''$, which then is piecewise linear. We then rewrite the equation as

$$\int (h'')^2 - \int (g'')^2 \geq 2 \int \phi e'',$$

where the goal is now to show that the right-hand side is nonnegative.

We have

$$\int \phi e'' = \sum_{i=1}^{m-1} \int_{x_i}^{x_{i+1}} \phi e'' = \sum_{i=1}^{m-1} [\phi e']_{x_i}^{x_{i+1}} - \sum_{i=1}^{m-1} \int_{x_i}^{x_{i+1}} \phi' e'$$

where we have used integration by parts. By the natural end conditions, the first term vanishes, as

$$\sum_{i=1}^{m-1} [\phi e']_{x_i}^{x_{i+1}} = \phi(x_m) e'(x_m) - \phi(x_1) e'(x_1) = 0.$$

The second term also vanishes, because as ϕ is piecewise linear, ϕ' is piecewise constant. This allows us to write

$$\int_{x_i}^{x_{i+1}} \phi' e' = \phi' \Big|_{[x_i, x_{i+1}]} \int_{x_i}^{x_{i+1}} e' = \phi' \Big|_{[x_i, x_{i+1}]} \Big(e(x_{i+1}) - e(x_i) \Big) = 0,$$

which vanishes as $e(x_i) = 0$ for all i. Thus, we have shown that the right-hand side is nonnegative, and the minimization property of Theorem 2.9 also holds for the natural spline of Section 2.7.

3 B-splines

Exercise 3.1. Suppose that x_0, x_1, x_2 are distinct, and let $f_i = f(x_i)$, $i = 0, 1, 2$, for some function f. Show by direct calculation that the recursive formula

$$[x_0, x_1, x_2]f = \frac{\frac{f_2-f_1}{x_2-x_1} - \frac{f_1-f_0}{x_1-x_0}}{x_2 - x_0}$$

can be expressed as

$$[x_0, x_1, x_2]f = \sum_{i=0}^{2} \frac{f_i}{\prod_{j\neq i}(x_i - x_j)}.$$

Solution 3.1. We have

$$[x_0, x_1, x_2]f = \frac{\frac{f_2-f_1}{x_2-x_1} - \frac{f_1-f_0}{x_1-x_0}}{x_2 - x_0}$$

where we begin by expanding the top fractions.

$$\begin{aligned}
\frac{f_2 - f_1}{x_2 - x_1} - \frac{f_1 - f_0}{x_1 - x_0} &= \frac{f_2}{x_2 - x_1} - f_1\left(\frac{1}{x_2 - x_1} + \frac{1}{x_1 - x_0}\right) + \frac{f_0}{x_1 - x_0} \\
&= \frac{f_2}{x_2 - x_1} - f_1\frac{x_1 - x_0 + x_2 - x_1}{(x_2 - x_1)(x_1 - x_0)} + \frac{f_0}{x_1 - x_0} \\
&= \frac{f_2}{x_2 - x_1} - f_1\frac{x_2 - x_0}{(x_2 - x_1)(x_1 - x_0)} + \frac{f_0}{x_1 - x_0} \\
&= \frac{f_2}{x_2 - x_1} + \frac{f_1(x_2 - x_0)}{(x_1 - x_2)(x_1 - x_0)} + \frac{f_0}{x_1 - x_0} \qquad (3.1)
\end{aligned}$$

Dividing (3.1) by $x_2 - x_0$ gives

$$\begin{aligned}
[x_0, x_1, x_2]f &= \frac{\frac{f_2-f_1}{x_2-x_1} - \frac{f_1-f_0}{x_1-x_0}}{x_2 - x_0} \\
&= \frac{f_2}{(x_2 - x_1)(x_2 - x_0)} + \frac{f_1}{(x_1 - x_2)(x_1 - x_0)} + \frac{f_0}{(x_1 - x_0)(x_2 - x_0)} \\
&= \frac{f_2}{(x_2 - x_1)(x_2 - x_0)} + \frac{f_1}{(x_1 - x_2)(x_1 - x_0)} + \frac{f_0}{(x_0 - x_1)(x_0 - x_2)} \\
&= \sum_{i=0}^{2} \frac{f_i}{\prod_{j\neq i}(x_i - x_j)},
\end{aligned}$$

as desired.

Exercise 3.2. Show that if $f(x) = 1/x$ and that $x_0, x_1, \ldots, x_k \neq 0$ then

$$[x_0, \ldots, x_k]f = (-1)^k \frac{1}{x_0 x_1 \cdots x_k}.$$

Solution 3.2. In the base case we have simply

$$[x_0]f = \frac{1}{x_0} = (-1)^0 \frac{1}{x_0}.$$

In the case of $x_0 = x_1 = \cdots = x_k$ we have

$$[\underbrace{x_0, x_0, \ldots, x_0}_{k+1}]f = \frac{f^{(k)}(x_0)}{k!} = (-1)^k \frac{1}{x_0^{k+1}} \frac{k!}{k!} = (-1)^k \frac{1}{x_0 x_1 \cdots x_k},$$

so mulitplicities are handled correctly. For two distinct points x_0, x_1 we have

$$[x_0, x_1]f = \frac{f_1 - f_0}{x_1 - x_0} = \frac{1/x_1 - 1/x_0}{x_1 - x_0} = \frac{x_0 - x_1}{x_0 x_1 (x_1 - x_0)} = (-1)^1 \frac{1}{x_0 x_1},$$

so the formula holds for $k = 1$. Assume that the formula holds for $k = n$, and consider $k = n + 1$. We have

$$\begin{aligned}
[x_0, \ldots, x_{n+1}]f &= \frac{[x_1, \ldots, x_{n+1}]f - [x_0, \ldots, x_n]f}{x_{n+1} - x_0} \\
&= \frac{(-1)^n \frac{1}{x_1 \cdots x_{n+1}} - (-1)^n \frac{1}{x_0 \cdots x_n}}{x_{n+1} - x_0} \\
&= \frac{(-1)^n \frac{1}{x_{n+1}} - (-1)^n \frac{1}{x_0}}{(x_{n+1} - x_0) x_1 \cdots x_n} \\
&= \frac{(-1)^n \frac{x_0 - x_{n+1}}{x_0 x_{n+1}}}{(x_{n+1} - x_0) x_1 \cdots x_n} \\
&= (-1)^{n+1} \frac{x_{n+1} - x_0}{(x_{n+1} - x_0) x_0 x_1 \cdots x_n x_{n+1}} \\
&= (-1)^{n+1} \frac{1}{x_0 x_1 \cdots x_{n+1}},
\end{aligned}$$

proving the formula by induction.

Exercise 3.3. Prove the Leibniz rule for divided differences:

$$[x_0, x_1, \ldots, x_k](fg) = \sum_{i=0}^{k} [x_0, \ldots, x_i]f [x_i, \ldots, x_k]g.$$

Hint: let p and q be the polynomials of degree $\leq k$ that interpolate f and g respectively at $x_0, x_1, \ldots, x_k$, and express p and q as

$$p(x) = \sum_{i=0}^{k} (x - x_0) \cdots (x - x_{i-1})[x_0, \ldots, x_i]f,$$
$$q(x) = \sum_{j=0}^{k} (x - x_{j+1}) \cdots (x - x_k)[x_j, \ldots, x_k]g.$$

Now consider the polynomial pq.

Solution 3.3. As the hint suggests, we consider the polynomials p and q of degree $\leq k$ that interpolate f and g respectively at $x_0, x_1, \ldots, x_k$, expressed as

$$p(x) = \sum_{i=0}^{k} (x - x_0) \cdots (x - x_{i-1})[x_0, \ldots, x_i]f,$$
$$q(x) = \sum_{j=0}^{k} (x - x_{j+1}) \cdots (x - x_k)[x_j, \ldots, x_k]g.$$

In addition, let r be the polynomial of degree $\leq k$ that interpolates fg at $x_0, x_1, \ldots, x_k$, expressed as

$$r(x) = \sum_{m=0}^{k} (x - x_0) \cdots (x - x_{m-1})[x_0, \ldots, x_m](fg).$$

We now consider the polynomial pq. In order to illustrate the idea, let $k = 1$ for now. We have

$$p(x) = [x_0]f + (x - x_0)[x_0, x_1]f,$$
$$q(x) = (x - x_1)[x_0, x_1]g + [x_1]g.$$

The polynomial pq is then

$$pq(x) = [x_0]f[x_1]g + (x - x_0)[x_0, x_1]f[x_1]g + (x - x_1)[x_0]f[x_0, x_1]g \\ + (x - x_0)(x - x_1)[x_0, x_1]f[x_0, x_1]g.$$

We can see that the polynomial pq is of degree ≤ 2 and interpolates fg at x_0 and x_1. However, note that the rightmost term is zero at both the interpolation points, so removing it still gives us an interpolating polynomial, now of degree ≤ 1. It is therefore the unique interpolating polynomial, of the form

$$\overline{pq}(x) = [x_0]f[x_1]g + (x - x_0)[x_0, x_1]f[x_1]g + (x - x_1)[x_0]f[x_0, x_1]g.$$

The leading coefficient of $\overline{pq}$ is then

$$\text{l.c.}(\overline{pq}) = [x_0, x_1]f[x_1]g + [x_0]f[x_0, x_1]g.$$

Now, we consider the polynomial r. We have

$$r(x) = [x_0](fg) + (x - x_0)[x_0, x_1](fg),$$

which interpolates fg at x_0, x_1. Due to the uniqueness of the interpolating polynomial, we must have $r = \overline{pq}$. As we can easily see, the leading coefficient of r is

$$\text{l.c.}(r) = [x_0, x_1](fg).$$

As the polynomials are the same, the leading coefficients must be the same, so we have

$$[x_0, x_1](fg) = [x_0, x_1]f[x_1]g + [x_0]f[x_0, x_1]g,$$

proving the case when $k = 1$.

We now consider the general case. The polynomial pq is now of the form

$$pq(x) = \sum_{i=0}^{k}\sum_{j=0}^{k}[x_0, \dots, x_i]f[x_j, \dots, x_k]g \prod_{a=0}^{i-1}(x - x_a) \prod_{b=j+1}^{k}(x - x_b).$$

When $j \leq i - 1$, the products on the right side will be zero at $x_0, x_1, \dots, x_k$, so we can safely remove these terms while still having an interpolating polynomial. This gives us $\overline{pq}$, of the form

$$\overline{pq}(x) = \sum_{i=0}^{k}\sum_{j=i}^{k}[x_0, \dots, x_i]f[x_j, \dots, x_k]g \prod_{a=0}^{i-1}(x - x_a) \prod_{b=j+1}^{k}(x - x_b),$$

which interpolates fg at $x_0, x_1, \dots, x_k$ and has degree $\leq k$. The leading coefficient of $\overline{pq}$ is found when $i = j$, as it includes the term

$$\prod_{a=0}^{i-1}(x - x_a) \prod_{b=i+1}^{k}(x - x_b) = \prod_{\substack{a=0\\a\neq i}}^{k}(x - x_a),$$

with the leading coefficient

$$\text{l.c.}(\overline{pq}) = \sum_{i=0}^{k}[x_0, \dots, x_i]f[x_i, \dots, x_k]g.$$

From r we can again read the leading coefficient simply as $[x_0, \dots, x_k](fg)$, proving Leibniz' rule.

Exercise 3.4. Use the recursion formula (Theorem 3.4) to show that

a) $B[0,0,0,1](x) = (1-x)^2 B[0,1](x)$,

b) $B[0,0,1,2](x) = x(2-3x/2)B[0,1](x) + \frac{1}{2}(2-x)^2 B[1,2](x)$,

c) $B[0,1,1,2](x) = x^2 B[0,1](x) + (2-x)^2 B[1,2](x)$.

Solution 3.4. The recursion formula states that for $d \geq 1$,

$$B_{i,d}(x) = \frac{x - t_i}{t_{i+d} - t_i} B_{i,d-1}(x) + \frac{t_{i+d+1} - x}{t_{i+d+1} - t_{i+1}} B_{i+1,d-1}(x).$$

I'm not entirely sure which knots the B-splines are defined on, however I guess the exercise is to relate these ones to the cardinal B-splines.

a) In this case we have $t_i = t_d$, meaning that the first term in the recursion formula is zero. We then have

$$B[0,0,0,1](x) = 0 + \frac{1-x}{1-0} B[0,0,1](x).$$

Recursing on $B[0,0,1](x)$, again the first term is zero, so we have

$$(1-x)B[0,0,1](x) = (1-x)\frac{1-x}{1-0} B[0,1](x) = (1-x)^2 B[0,1](x).$$

b) Next, we have

$$\begin{aligned} B[0,0,1,2](x) &= \frac{x-0}{1-0} B[0,0,1](x) + \frac{2-x}{2-0} B[0,1,2](x) \\ &= x(1-x)B[0,1](x) + \frac{2-x}{2}\left(xB[0,1](x) + (2-x)B[1,2](x)\right) \\ &= x\left(1 - x + \frac{2-x}{2}\right) B[0,1](x) + \frac{1}{2}(2-x)^2 B[1,2](x) \\ &= x\left(2 - \frac{3}{2}x\right) B[0,1](x) + \frac{1}{2}(2-x)^2 B[1,2](x). \end{aligned}$$

c) Finally, for the last expression we have

$$B[0,1,1,2](x) = \frac{x-0}{1-0} B[0,1,1](x) + \frac{2-x}{2-1} B[1,1,2](x).$$

Considering the two terms separately, we have

$$B[0,1,1](x) = xB[0,1](x)$$

as the second term vanishes, and

$$B[1,1,2](x) = \frac{2-x}{2-1} B[1,2](x) = (2-x)^2 B[1,2](x),$$

where the first term vanishes. Combining these, we find

$$B[0,1,1,2](x) = x^2 B[0,1](x) + (2-x)^2 B[1,2](x),$$

as desired.

Exercise 3.5. a) Prove Theorem 3.5, and b) use it to show that for distinct knots,

$$B_{i,2}(t_{i+1}) = \frac{t_{i+1}-t_i}{t_{i+2}-t_i}, \quad B_{i,2}(t_{i+2}) = \frac{t_{i+3}-t_{i+2}}{t_{i+3}-t_{i+1}}.$$

Solution 3.5. Theorem 3.5 states that for any $j = i, i+1, \ldots, i+d+1$,

$$B[t_i, \ldots, t_{i+d+1}](t_j) = B[t_i, \ldots, t_{j-1}, t_{j+1}, \ldots, t_{i+d+1}](t_j),$$

that is, the spline with the knot t_j removed.

Let

$$f(x) = (\cdot - x) \quad \text{and} \quad g(x) = (\cdot - x)_+^{d-1},$$

as then

$$\begin{aligned} B_{i,d}(x) &= (t_{i+d+1} - t_i)[t_i, \ldots, t_{i+d+1}](\cdot - x)_+^d \\ &= (t_{i+d+1} - t_i)[t_i, \ldots, t_{i+d+1}](fg). \end{aligned}$$

Dropping the initial coefficient $(t_{i+d+1} - t_i)$ for now, we have by Leibniz' rule

$$\begin{aligned} [t_i, \ldots, t_{i+d+1}](fg) &= [t_j, t_i, \ldots, \hat{t}_j, \ldots, t_{i+d+1}](fg) \\ &= [t_j]f[t_j, t_i, \ldots, \hat{t}_j, \ldots, t_{i+d+1}](g) \\ &\quad + [t_j, t_i]f[t_i, \ldots, \hat{t}_j, \ldots, t_{i+d+1}](g) \\ &\quad + [t_j, t_i, t_{i+1}]f[t_{i+1}, \ldots, \hat{t}_j, \ldots, t_{i+d+1}](g) \\ &\quad + \cdots, \end{aligned}$$

where the hat denotes the omitted knot. As f is linear, we have

$$\underbrace{[t_j, t_i, \ldots, t_k]}_{\geq 3} f = 0 \quad \text{and} \quad [t_j, t_i]f = 1.$$

We then have

$$\begin{aligned} [t_i, \ldots, t_{i+d+1}](fg) &= f(t_j)[t_j, t_i, \ldots, \hat{t}_j, \ldots, t_{i+d+1}](g) \\ &\quad + [t_i, \ldots, \hat{t}_j, \ldots, t_{i+d+1}](g) \\ &= (t_j - x)[t_j, t_i, \ldots, \hat{t}_j, \ldots, t_{i+d+1}](g) \\ &\quad + [t_i, \ldots, \hat{t}_j, \ldots, t_{i+d+1}](g). \end{aligned}$$

At $x = t_j$, we thus have, instering back the coefficient,

$$B[t_i, \dots, t_{i+d+1}](t_j) = B[t_i, \dots, \hat{t}_j, \dots, t_{i+d+1}](t_j),$$

as we wanted to show.

For the second part, we have

$$\begin{aligned}
B_{i,2}(t_{i+1}) &= B[t_i, t_{i+1}, t_{i+2}, t_{i+3}](t_{i+1}) = B[t_i, t_{i+2}, t_{i+3}](t_{i+1}) \\
&= \frac{t_{i+1} - t_i}{t_{i+2} - t_i} B[t_i, t_{i+2}](t_{i+1}) + \frac{t_{i+3} - t_{i+1}}{t_{i+3} - t_{i+2}} B[t_{i+2}, t_{i+3}](t_{i+1}) \\
&= \frac{t_{i+1} - t_i}{t_{i+2} - t_i}
\end{aligned}$$

Here, we've used that $B[a, b](x) = 0$ for $x \notin [a, b)$, and $B[a, b](x) = 1$ for $x \in [a, b)$. For the second expression we similarly have

$$\begin{aligned}
B_{i,2}(t_{i+2}) &= B[t_i, t_{i+1}, t_{i+2}, t_{i+3}](t_{i+2}) = B[t_i, t_{i+1}, t_{i+3}](t_{i+2}) \\
&= \frac{t_{i+2} - t_i}{t_{i+3} - t_i} B[t_i, t_{i+1}](t_{i+2}) + \frac{t_{i+3} - t_{i+2}}{t_{i+3} - t_{i+1}} B[t_{i+1}, t_{i+3}](t_{i+2}) \\
&= \frac{t_{i+3} - t_{i+2}}{t_{i+3} - t_{i+1}}.
\end{aligned}$$

Exercise 3.6. Show that

$$\begin{aligned}
B[\underbrace{a, \dots, a}_{d+1}, b](x) &= \left(\frac{b - x}{b - a}\right)^d B[a, b](x), \\
B[a, \underbrace{b, \dots, b}_{d+1}](x) &= \left(\frac{x - a}{b - a}\right)^d B[a, b](x).
\end{aligned}$$

Use this to show that

$$B[a, \underbrace{b, \dots, b}_{d}, c](x) = \left(\frac{x - a}{b - a}\right)^d B[a, b](x) + \left(\frac{c - x}{c - b}\right)^d B[b, c](x).$$

Show that this B-spline is continuous on $\mathbb{R}$.

Solution 3.6. We begin by showing the first two expressions. Note that when $t_i = t_d$, then

$$[t_i, \dots, t_d](\cdot - x)_+^{d-1} = 0 \quad \text{for } x \neq t_i,$$

and when we additionally define the term to be zero when $x = t_i$, we have

$$B[\underbrace{a, \dots, a}_{d+1}, b](x) = \frac{b - x}{b - a} B[\underbrace{a, \dots, a}_{d}, b](x) = \dots = \left(\frac{b - x}{b - a}\right)^d B[a, b](x).$$

Similarly, we find that

$$B[a,\underbrace{b,\dots,b}_{d+1}](x) = \left(\frac{x-a}{b-a}\right)^d B[a,b](x).$$

Next, we then have by the recursive formula

$$\begin{aligned} B[a,\underbrace{b,\dots,b}_{d},c](x) &= \frac{x-a}{b-a}B[a,\underbrace{b,\dots,b}_{d}](x) + \frac{c-x}{c-b}B[\underbrace{b,\dots,b}_{d},c](x) \\ &= \frac{x-a}{b-a}\left(\frac{x-a}{b-a}\right)^{d-1} B[a,b](x) + \frac{c-x}{c-b}\left(\frac{c-x}{c-b}\right)^{d-1} B[b,c](x) \\ &= \left(\frac{x-a}{b-a}\right)^{d} B[a,b](x) + \left(\frac{c-x}{c-b}\right)^{d} B[b,c](x). \end{aligned}$$

Using again that

$$B[a,b](x) = \begin{cases} 0, & x \notin [a,b), \\ 1, & x \in [a,b), \end{cases}$$

we can rewrite the spline as

$$B[a,\underbrace{b,\dots,b}_{d},c](x) = \begin{cases} \left(\frac{x-a}{b-a}\right)^d & x \in [a,b), \\ \left(\frac{c-x}{c-b}\right)^d & x \in [b,c), \\ 0 & \text{otherwise.} \end{cases}$$

At the knot b, the spline is continuous, as

$$\frac{b-a}{b-a} = 1 = \frac{c-b}{c-b},$$

while they are both zero at $x = a$ and $x = c$ respectively.

Exercise 3.7. Show that

$$B_{i,d}(x) = (-1)^{d+1}(t_{i+d+1} - t_i)[t_i, t_{i+1}, \dots, t_{i+d+1}](x - \cdot)_+^d, \quad x \in \mathbb{R}.$$

Hint: Use the fact that

$$(t-x)_+^d - (-1)^{d+1}(x-t)_+^d = (t-x)^d.$$

Use this to show that for distinct knots,

$$B_{i,d}(x) = (t_{i+d+1} - t_i) \sum_{j=i}^{i+d+1} \frac{(x-t_j)_+^d}{\prod_{\substack{k=i \\ k\neq j}}^{i+d+1}(t_k - t_j)}.$$

Solution 3.7. A reasonable place to start baced on the hint seems to note that

$$(t-x)_+^d = (-1)^{d+1}(x-t)_+^d - (t-x)^d.$$

Then, if we can show that

$$[t_i, t_{i+1}, \dots, t_{i+d+1}](\cdot - x)^d = 0$$

we are practically done with the first part. As $(t-x)^d$ is a polynomial of degree d, and we have $d+1$ knots, the term vanishes.

We then have

$$\begin{aligned}
B_{i,d}(x) &= (t_{i+d+1} - t_i)[t_i, t_{i+1}, \dots, t_{i+d+1}](\cdot - x)_+^d \\
&= (t_{i+d+1} - t_i)[t_i, t_{i+1}, \dots, t_{i+d+1}]\left((-1)^{d+1}(x - \cdot)_+^d - (t_{i+d+1} - x)^d\right) \\
&= (-1)^{d+1}(t_{i+d+1} - t_i)[t_i, t_{i+1}, \dots, t_{i+d+1}](x - \cdot)_+^d
\end{aligned}$$

as desired.

We already know by the forward difference formula that we can write

$$B_{i,d}(x) = (t_{i+d+1} - t_i) \sum_{j=1}^{i+d+1} \frac{(t_j - x)_+^d}{\prod_{\substack{k=i \\ k\neq j}}^{i+d+1} (t_j - t_k)}.$$

We can rewrite the denominator as

$$\prod_{\substack{k=i \\ k\neq j}}^{i+d+1} (t_j - t_k) = \prod_{\substack{k=1 \\ k\neq j}}^{i+d+1} (-1)(t_k - t_j) = (-1)^{d+1} \prod_{\substack{k=1 \\ k\neq j}}^{i+d+1} (t_k - t_j).$$

Using the forward difference formula on $(-1)^{d+1}(x-t)_+^d$ we then find

$$\begin{aligned}
B_{i,d}(x) &= (t_{i+d+1} - t_i)(-1)^{d+1} \sum_{j=1}^{i+d+1} \frac{(x - t_j)_+^d}{\prod_{\substack{k=i \\ k\neq j}}^{i+d+1} (t_j - t_k)} \\
&= (t_{i+d+1} - t_i)(-1)^{d+1} \sum_{j=1}^{i+d+1} \frac{(x - t_j)_+^d}{(-1)^{d+1}\prod_{\substack{k=i \\ k\neq j}}^{i+d+1} (t_k - t_j)} \\
&= (t_{i+d+1} - t_i) \sum_{j=1}^{i+d+1} \frac{(x - t_j)_+^d}{\prod_{\substack{k=i \\ k\neq j}}^{i+d+1} (t_k - t_j)},
\end{aligned}$$

as desired.

Exercise 3.8. Use the recursion formula to show that (for $d \geq 1$)

$$\sum_{i=1}^{n} B_{i,d}(x) = 1, \quad x \in [t_{d+1}, t_{n+1}].$$

Solution 3.8. We have

$$\begin{aligned}
\sum_{i=1}^{n} B_{i,d}(x) &= \sum_{i=1}^{n} \left(\frac{x - t_i}{t_{i+d} - t_i} B_{i,d-1}(x) + \frac{t_{i+d+1} - x}{t_{i+d+1} - t_{i+1}} B_{i+1,d-1}(x) \right) \\
&= \sum_{i=1}^{n} \frac{x - t_i}{t_{i+d} - t_i} B_{i,d-1}(x) + \sum_{i=1}^{n} \frac{t_{i+d+1} - x}{t_{i+d+1} - t_{i+1}} B_{i+1,d-1}(x) \\
&= \sum_{i=1}^{n} \frac{x - t_i}{t_{i+d} - t_i} B_{i,d-1}(x) + \sum_{i=2}^{n+1} \frac{t_{i+d} - x}{t_{i+d} - t_i} B_{i,d-1}(x) \\
&= \frac{x - t_1}{t_{1+d} - t_1} B_{1,d-1}(x) + \sum_{i=2}^{n} \left(\left(\frac{x - t_i}{t_{i+d} - t_i} + \frac{t_{i+d} - x}{t_{i+d} - t_i} \right) B_{i,d-1}(x) \right) \\
&\quad + \frac{t_{n+d+1} - x}{t_{n+d+1} - t_{n+1}} B_{n+1,d-1}(x) \\
&= \frac{x - t_1}{t_{1+d} - t_1} B_{1,d-1}(x) + \sum_{i=2}^{n} \left(\frac{t_{i+d} - t_i}{t_{i+d} - t_i} \right) B_{i,d-1}(x) \\
&\quad + \frac{t_{n+d+1} - x}{t_{n+d+1} - t_{n+1}} B_{n+1,d-1}(x) \\
&= \frac{x - t_1}{t_{1+d} - t_1} B_{1,d-1}(x) + \sum_{i=2}^{n} B_{i,d-1}(x) + \frac{t_{n+d+1} - x}{t_{n+d+1} - t_{n+1}} B_{n+1,d-1}(x).
\end{aligned}$$

The first term can be written more explicitly as

$$\frac{x - t_1}{t_{1+d} - t_1} B_{1,d-1}(x) = \frac{x - t_1}{t_{1+d} - t_1} B[t_1, \ldots, t_{d+1}](x)$$

which is zero for $x \geq t_{d+1}$. The last term can be written as

$$\frac{t_{n+d+1} - x}{t_{n+d+1} - t_{n+1}} B_{n+1,d-1}(x) = \frac{t_{n+d+1} - x}{t_{n+d+1} - t_{n+1}} B[t_{n+1}, \ldots, t_{d+n+1}](x)$$

which similarly is zero for $x \leq t_{n+1}$.

Therefore, for $x \in [t_{d+1}, t_{n+1}]$, we have

$$\sum_{i=1}^{n} B_{i,d}(x) = \sum_{i=2}^{n} B_{i,d-1}(x) = \cdots = \sum_{i=d+1}^{n} B_{i,0}(x) = 1.$$

Exercise 3.9. Prove Theorem 3.5. Hint: Derive an analogy of (3.16) by first expressing $[t_i, \ldots, t_{i+d+1}]$ in the form

$$[t_j, t_i, \ldots, \hat{t}_j, \ldots, t_{i+d+1}].$$

Solution 3.9. This was already done as the first part of Exercise 3.5.

Exercise 3.10. Given a knot vector $\mathbf{t} = (t_i)_{i=1}^{n+d+1}$ and a real number x, with $x \in [t_{d+1}, t_{n+1}]$, write a routine to determine an index μ such that $x \in [t_\mu, t_{\mu+1}]$ with $[t_\mu, t_{\mu+1}]$ non-empty. A call to this routine is always needed before running Algorithms 1 and 2 of Section 3.9. By letting μ be an input parameter as well as an output parameter you can minimize the searching, for example during plotting when the routine is called with many values of x in the same know interval.

Solution 3.10. This simply involves implementing a binary search, assuming that $\mathbf{t}$ is sorted.

Exercise 3.11. Implement Algorithm 1 of Section 3.9 in your favourite programming lanuage.

Solution 3.11. Implementing the algorithm requires a little bit of setup, in order to facilitate an efficient implementation. I'll therefore work through the first couple of steps. We begin by fixing $x \in [t_\mu, t_{\mu+1}]$ and $B_{\mu,0}(x) = 1$. We then have

$$\begin{aligned}
B_{\mu,1}(x) &= \frac{x - t_\mu}{t_{\mu+1} - t_\mu} B_{\mu,0}(x) + \frac{t_{\mu+2} - x}{t_{\mu+2} - t_{\mu+1}} B_{\mu+1,0}(x) \\
&= \frac{x - t_\mu}{t_{\mu+1} - t_\mu} + 0 = \frac{x - t_\mu}{t_{\mu+1} - t_\mu} \\
B_{\mu-1,1}(x) &= \frac{x - t_{\mu-1}}{t_\mu - t_{\mu-1}} B_{\mu-1,0}(x) + \frac{t_{\mu+1} - x}{t_{\mu+1} - t_\mu} B_{\mu,0}(x) \\
&= 0 + \frac{t_{\mu+1} - x}{t_{\mu+1} - t_\mu} = \frac{t_{\mu+1} - x}{t_{\mu+1} - t_\mu}.
\end{aligned}$$

The computational graph for the first couple steps is then illustrated in Figure 3. The illustration on the right more explicitly encodes the dependence on previous values.

NOTE: Implement the algorithm without recursion.

Exercise 3.12. Implement Algorithm 2 of Section 3.9 in your favourite programming language.

Solution 3.12. A crucial detail of this algorithm is that for the interval $[t_\mu, t_{\mu+1}]$, we require the value of $t_{\mu+d+1}$, which is not entirely clear when we are in the last interval. To overcome this, we can simply append the knot vector with d additional knots, where the last d knots are equal to t_{n+1}. This way, we can always find the value of $t_{\mu+d+1}$. Doing this, we generate Figure 4.

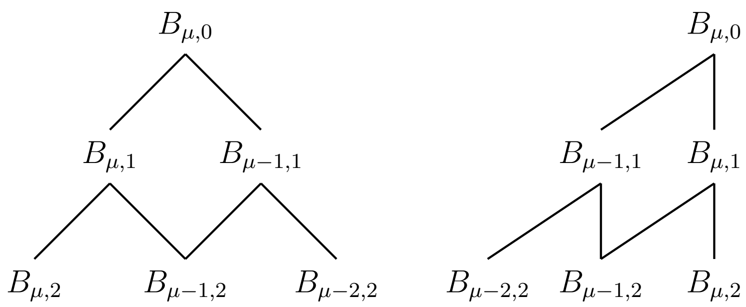

Figure 3: Computational graph for the first couple steps of Algorithm 1.

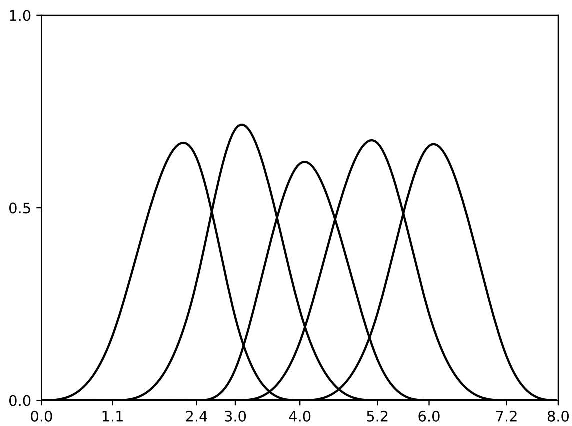

Figure 4: Cubic B-splines on distinct knots computed with de Boor's algorithm without recursion. $n = 5$, with $\mathbf{t} = (0.0, 1.1, 2.4, 3, 4, 5.2, 6.0, 7.2, 8)$.