CHAPTER 5

Spline Approximation of Functions and Data

This chapter introduces a number of methods for obtaining spline approximations to given functions, or more precisely, to data obtained by sampling a function. In Section 5.1, we focus on local methods where the approximation at a point x only depends on data values near x. Connecting neighbouring data points with straight lines is one such method where the value of the approximation at a point only depends on the two nearest data points.

In order to get smoother approximations, we must use splines of higher degree. With cubic polynomials we can prescribe, or *interpolate*, position and first derivatives at two points. Therefore, given a set of points with associated function values and first derivatives, we can determine a sequence of cubic polynomials that interpolate the data, joined together with continuous first derivatives. This is the cubic Hermite interpolant of Section 5.1.2.

In Section 5.2 we study global cubic approximation methods where we have to solve a system of equations involving all the data points in order to obtain the approximation. Like the local methods in Section 5.1, these methods *interpolate* the data, which now only are positions. The gain in turning to global methods is that the approximation may have more continuous derivatives and still be as accurate as the local methods.

The cubic spline interpolant with so called natural end conditions solves an interesting extremal problem. Among all functions with a continuous second derivative that interpolate a set of data, the natural cubic spline interpolant is the one whose integral of the square of the second derivative is the smallest. This is the foundation for various interpretations of splines, and is all discussed in Section 5.2.

Two approximation methods for splines of arbitrary degree are described in Section 5.3. The first method is spline interpolation with B-splines defined on some rather arbitrary knot vector. The disadvantage of using interpolation methods is that the approximations have a tendency to oscillate. If we reduce the dimension of the approximating spline space, and instead minimize the error at the data points this problem can be greatly reduced. Such *least squares methods* are studied in Section 5.3.2.

We end the chapter by a discussing a very simple approximation method, the *Variation Diminishing Spline Approximation*. This approximation scheme has the desirable ability to transfer the sign of some of the derivatives of a function to the approximation. This is important since many important characteristics of the shape of a function is closely related to the sign of the derivatives.

5.1 Local Approximation Methods

When we construct an approximation to data, it is usually an advantage if the approximation at a point x only depends on the data near x. If this is the case, changing the data in some small area will only affect the approximation in the same area. The variation diminishing approximation method and in particular piecewise linear interpolation has this property, it is a *local* method. In this section we consider another local approximation method.

5.1.1 Piecewise linear interpolation

The simplest way to obtain a continuous approximation to a set of ordered data points is to connect neighbouring data points with straight lines. This approximation is naturally enough called the *piecewise linear interpolant* to the data. It is clearly a linear spline and can therefore be written as a linear combination of B-splines on a suitable knot vector. The knots must be at the data points, and since the interpolant is continuous, each interior knot only needs to occur once in the knot vector. The construction is given in the following proposition.

Proposition 5.1. *Let $(x_i, y_i)_{i=1}^m$ be a set of data points with $x_i < x_{i+1}$ for $i = 1, \ldots, m-1$, and construct the 2-regular knot vector $\boldsymbol{t}$ as*

$$\boldsymbol{t} = (t_i)_{i=1}^{m+2} = (x_1, x_1, x_2, x_3, \ldots, x_{m-1}, x_m, x_m).$$

Then the linear spline g given by

$$g(x) = \sum_{i=1}^{m} y_i B_{i,1}(x)$$

satisfies the interpolation conditions

$$g(x_i) = y_i, \quad \text{for } i = 1, \ldots, m-1, \qquad \text{and} \quad \lim_{x \to x_m^-} g(x) = y_m. \tag{5.1}$$

The last condition states that the limit of g from the left at x_m is y_m. If the data are taken from a function f so that $y_i = f(x_i)$ for $i = 1, \ldots, m$, the interpolant g is often denoted by $I_1 f$.

Proof. From Example 2.2 in Chapter 2, we see that the B-spline $B_{i,1}$ for $1 \le i \le m$ is given by

$$B_{i,1}(x) = \begin{cases} (x - x_{i-1})/(x_i - x_{i-1}), & \text{if } x_{i-1} \le x < x_i, \\ (x_{i+1} - x)/(x_{i+1} - x_i), & \text{if } x_i \le x < x_{i+1}, \\ 0, & \text{otherwise,} \end{cases}$$

where we have set $x_0 = x_1$ and $x_{m+1} = x_m$. This means that $B_{i,1}(x_i) = 1$ for $i < m$ and $\lim_{x \to x_m^-} B_{m,1}(x) = 1$, while $B_{i,1}(x_j) = 0$ for all $j \neq i$, so the interpolation conditions (5.1) are satisfied. ∎

The piecewise linear interpolant preserves the shape of the data extremely well. The obvious disadvantage of this approximation is its lack of smoothness.

Intuitively, it seems reasonable that if f is continuous, it should be possible to approximate it to within any accuracy by piecewise linear interpolants, if we let the distance between the data points become small enough. This is indeed the case. Note that the symbol $C^j[a,b]$ denotes the set of all functions defined on $[a,b]$ with values in $\mathbb{R}$ whose first j derivatives are continuous.

Proposition 5.2. *Suppose that $a = x_1 < x_2 < \cdots < x_m = b$ are given points, and set $\Delta\boldsymbol{x} = \max_{1\leq i\leq m-1}\{x_{i+1} - x_i\}$.*

1. *If $f \in C[a,b]$, then for every $\epsilon > 0$ there is a $\delta > 0$ such that if $\Delta x < \delta$, then $|f(x) - I_1 f(x)| < \epsilon$ for all $x \in [a,b]$.*
2. *If $f \in C^2[a,b]$ then for all $x \in [a,b]$,*

$$|f(x) - (I_1 f)(x)| \leq \frac{1}{8}(\Delta\boldsymbol{x})^2 \max_{a\leq z\leq b} |f''(z)|, \tag{5.2}$$

$$|f'(x) - (I_1 f)'(x)| \leq \frac{1}{2}\Delta\boldsymbol{x} \max_{a\leq z\leq b} |f''(z)|. \tag{5.3}$$

Part (i) of Proposition 5.2 states that piecewise linear interpolation to a continuous function converges to the function when the distance between the data points goes to zero. More specifically, given a tolerance ϵ, we can make the error less than the tolerance by choosing $\Delta\boldsymbol{x}$ sufficiently small.

Part (ii) of Proposition 5.2 gives an upper bound for the error in case the function f is smooth, which in this case means that f and its first two derivatives are continuous. The inequality in (5.2) is often stated as "piecewise linear approximation has approximation order two", meaning that $\Delta\boldsymbol{x}$ is raised to the power of two in (5.2).

The bounds in Proposition 5.2 depend both on $\Delta\boldsymbol{x}$ and the size of the second derivative of f. Therefore, if the error is not small, it must be because one of these quantities are large. If in some way we can find an upper bound M for f'', i.e.,

$$|f''(x)| \leq M, \quad \text{for} \quad x \in [a,b], \tag{5.4}$$

we can determine a value of $\Delta\boldsymbol{x}$ such that the error, measured as in (5.2), is smaller than some given tolerance ϵ. We must clearly require $(\Delta\boldsymbol{x})^2 M/8 < \epsilon$. This inequality holds provided $\Delta\boldsymbol{x} < \sqrt{8\epsilon/M}$. We conclude that for any $\epsilon > 0$, we have the implication

$$\Delta\boldsymbol{x} < \sqrt{\frac{8\epsilon}{M}} \quad \Longrightarrow \quad |f(x) - I_1 f(x)| < \epsilon, \quad \text{for } x \in [x_1, x_m]. \tag{5.5}$$

This estimate tells us how densely we must sample f in order to have error smaller than ϵ everywhere.

We will on occasions want to compute the piecewise linear interpolant to a given higher degree spline f. A spline does not necessarily have continuous derivatives, but at least we know where the discontinuities are. The following proposition is therefore meaningful.

Proposition 5.3. *Suppose that $f \in \mathbb{S}_{d,\boldsymbol{t}}$ for some d and $\boldsymbol{t}$ with interior knots of multiplicity at most d (so f is continuous). If the break points $(x_i)_{i=1}^m$ are chosen so as to include all the knots in $\boldsymbol{t}$ where f' is discontinuous, the bounds in (5.2) and (5.3) continue to hold.*

5.1.2 Cubic Hermite interpolation

The piecewise linear interpolant has the nice property of being a local construction: The interpolant on an interval $[x_i, x_{i+1}]$ is completely defined by the value of f at x_i and x_{i+1}. The other advantage of f is that it does not oscillate between data points and therefore preserves the shape of f if $\Delta \boldsymbol{x}$ is small enough. In this section we construct an interpolant which, unlike the piecewise linear interpolant, has continuous first derivative, and which, like the piecewise linear interpolant, only depends on data values locally. The price of the smoothness is that this interpolant requires information about derivatives, and shape preservation in the strong sense of the piecewise linear interpolant cannot be guaranteed. The interpolant we seek is the solution of the following problem.

Problem 5.4 (Piecewise Cubic Hermite Interpolation)**.** *Let the discrete data* $(x_i, f(x_i), f'(x_i))_{i=1}^m$ *with* $a = x_1 < x_2 < \cdots < x_m = b$ *be given. Find a function* $g = H_3 f$ *that satisfies the following conditions:*

1. *On each subinterval* (x_i, x_{i+1}) *the function* g *is a cubic polynomial.*
2. *The given function* f *is interpolated by* g *in the sense that*

$$g(x_i) = f(x_i), \quad \text{and} \quad g'(x_i) = f'(x_i), \quad \text{for } i = 1, \ldots, m. \tag{5.6}$$

A spline g that solves Problem 5.4 must be continuous and have continuous first derivative since two neighbouring pieces meet with the same value $f(x_i)$ and first derivative $f'(x_i)$ at a join x_i. Since Hf should be a piecewise cubic polynomial, it is natural to try and define a knot vector so that Hf can be represented as a linear combination of B-splines on this knot vector. To get the correct smoothness, we need at least a double knot at each data point. Since $d = 3$ and we have $2m$ interpolation conditions, the length of the knot vector should be $2m + 4$, and we might as well choose to use a 4-regular knot vector. We achieve this by making each interior data point a knot of multiplicity two and place four knots at the two ends. This leads to the knot vector

$$\boldsymbol{t} = (t_i)_{i=1}^{2m+4} = (x_1, x_1, x_1, x_1, x_2, x_2, \ldots, x_{m-1}, x_{m-1}, x_m, x_m, x_m, x_m), \tag{5.7}$$

which we call *the Cubic Hermite knot vector* on $\boldsymbol{x} = (x_1, \ldots, x_m)$. This allows us to construct the solution to Problem 5.4.

Proposition 5.5. *Problem 5.4 has a unique solution* Hf *in the spline space* $\mathbb{S}_{3,\boldsymbol{t}}$*, where* $\boldsymbol{t}$ *is given in equation (5.7). More specifically, the solution is given by*

$$Hf = \sum_{i=1}^{2m} c_i B_{i,3}, \tag{5.8}$$

where

$$\left.\begin{aligned} c_{2i-1} &= f(x_i) - \frac{1}{3}\Delta x_{i-1} f'(x_i), \\ c_{2i} &= f(x_i) + \frac{1}{3}\Delta x_i f'(x_i), \end{aligned}\right\} \quad \text{for } i = 1, \ldots, m, \tag{5.9}$$

where $\Delta x_j = x_{j+1} - x_j$*, and the points* x_0 *and* x_{m+1} *are defined by* $x_0 = x_1$ *and* $x_{m+1} = x_m$*.*

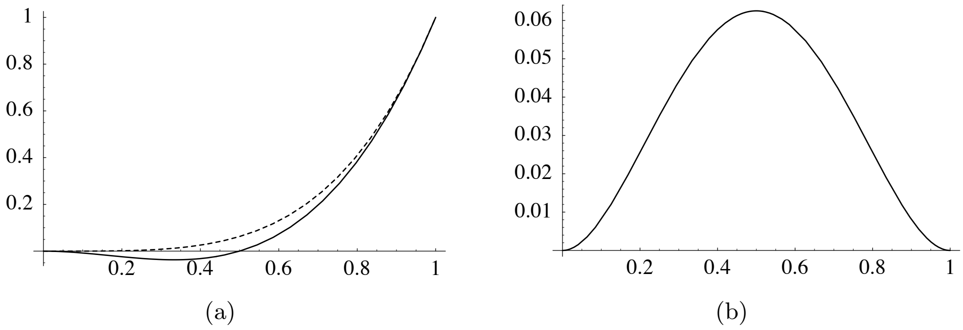


Figure 5.1. Figure (a) shows the cubic Hermite interpolant (solid) to $f(x) = x^4$ (dashed), see Example 5.6, while the error in this approximation is shown in (b).

Proof. We leave the proof that the spline defined by (5.9) satisfies the interpolation conditions in Problem 5.4 to the reader.

By construction, the solution is clearly a cubic polynomial. That there is only one solution follows if we can show that the only solution that solves the problem with $f(x_i) = f'(x_i) = 0$ for all i is the function that is zero everywhere. For if the general problem has two solutions, the difference between these must solve the problem with all the data equal to zero. If this difference is zero, the two solutions must be equal.

To show that the solution to the problem where all the data are zero is the zero function, it is clearly enough to show that the solution is zero in one subinterval. On each subinterval the function Hf is a cubic polynomial with value and derivative zero at both ends, and it therefore has four zeros (counting multiplicity) in the subinterval. But the only cubic polynomial with four zeros is the polynomial that is identically zero. From this we conclude that Hf must be zero in each subinterval and therefore identically zero. ∎

Let us see how this method of approximation behaves in a particular situation.

Example 5.6. We try to approximate the function $f(x) = x^4$ on the interval $[0, 1]$ with only one polynomial piece so that $m = 2$ and $[a, b] = [x_1, x_m] = [0, 1]$. Then the cubic Hermite knots are just the Bernstein knots. From (5.9) we find $(c_1, c_2, c_3, c_4) = (0, 0, -1/3, 1)$, and

$$(Hf)(x) = -\frac{1}{3}3x^2(1-x) + x^3 = 2x^3 - x^2.$$

The two functions f and Hf are shown in Figure 5.1.

Example 5.7. Let us again approximate $f(x) = x^4$ on $[0, 1]$, but this time we use two polynomial pieces so that $m = 3$ and $\boldsymbol{x} = (0, 1/2, 1)$. In this case the cubic Hermite knots are $\boldsymbol{t} = (0, 0, 0, 0, 1/2, 1/2, 1, 1, 1, 1)$, and we find the coefficients $\boldsymbol{c} = (0, 0, -1/48, 7/48, 1/3, 1)$. The two functions f and Hf are shown in Figure 5.1 (a). With the extra knots at $1/2$ (cf. Example 5.6), we get a much more accurate approximation to x^4. In fact, we see from the error plots in Figures 5.1 (b) and 5.1 (b) that the maximum error has been reduced from 0.06 to about 0.004, a factor of about 15.

Note that in Example 5.6 the approximation becomes negative even though f is nonnegative in all of $[0, 1]$. This shows that in contrast to the piecewise linear interpolant, the cubic Hermite interpolant Hf does not preserve the sign of f. However, it is simple to give conditions that guarantee Hf to be nonnegative.

Proposition 5.8. *Suppose that the function f to be approximated by cubic Hermite*

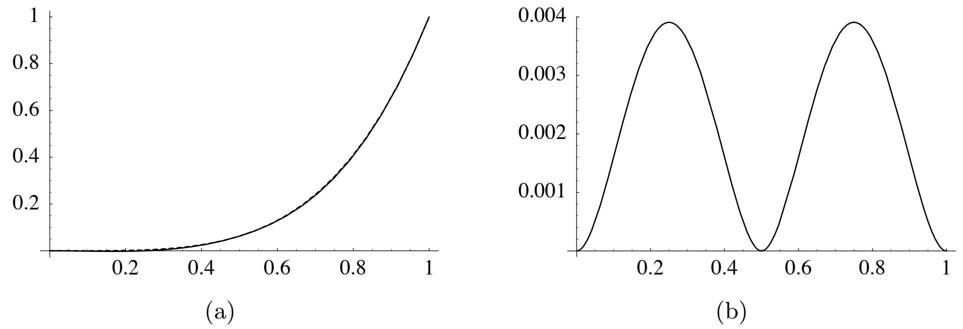


Figure 5.2. Figure (a) shows the cubic Hermite interpolant (solid) to $f(x) = x^4$ (dashed) with two polynomial pieces, see Example 5.7, while the error in the approximation is shown in (b).

interpolation satisfies the conditions

$$\left.\begin{aligned} f(x_i) - \frac{1}{3}\Delta x_{i-1} f'(x_i) \geq 0,\\ f(x_i) + \frac{1}{3}\Delta x_i f'(x_i) \geq 0, \end{aligned}\right\} \quad \textit{for } i = 1, \ldots, m.$$

Then the cubic Hermite interpolant Hf is nonnegative on $[a, b]$.

Proof. In this case, the spline approximation Hf given by Proposition 5.5 has nonnegative B-spline coefficients, so that $(Hf)(x)$ for each x is a sum of nonnegative quantities and therefore nonnegative. ∎

As for the piecewise linear interpolant, it is possible to relate the error to the spacing in $\boldsymbol{x}$ and the size of some derivative of f.

Proposition 5.9. *Suppose that f has continuous derivatives up to order four on the interval $[x_1, x_m]$. Then*

$$|f(x) - (Hf)(x)| \leq \frac{1}{384}(\Delta \boldsymbol{x})^4 \max_{a \leq z \leq b} |f^{(iv)}(z)|, \quad \textit{for } x \in [a, b]. \tag{5.10}$$

This estimate also holds whenever f is in some spline space $\mathbb{S}_{d,\boldsymbol{t}}$ provided f has a continuous derivative at all the x_i.

Proof. See a text on numerical analysis. ∎

The error estimate in (5.10) says that if we halve the distance between the interpolation points, then we can expect the error to decrease by a factor of $2^4 = 16$. This is usually referred to as “fourth order convergence”. This behaviour is confirmed by Examples 5.6 and 5.7 where the error was reduced by a factor of about 15 when $\Delta \boldsymbol{x}$ was halved.

From Proposition 5.9, we can determine a spacing between data points that guarantees that the error is smaller than some given tolerance. Suppose that

$$|f^{(iv)}(x)| \leq M, \quad \text{for } x \in [a, b].$$

For any $\epsilon > 0$ we then have

$$\Delta \boldsymbol{x} \leq \left(\frac{384\epsilon}{M}\right)^{1/4} \quad \Longrightarrow \quad |f(x) - (Hf)(x)| \leq \epsilon, \quad \text{for } x \in [a,b].$$

When $\epsilon \to 0$, the number $\epsilon^{1/4}$ goes to zero more slowly than the term $\epsilon^{1/2}$ in the corresponding estimate for piecewise linear interpolation. This means that when ϵ becomes small, we can usually use a larger $\Delta \boldsymbol{x}$ in cubic Hermite interpolation than in piecewise linear interpolation, or equivalently, we generally need fewer data points in cubic Hermite interpolation than in piecewise linear interpolation to obtain the same accuracy.

5.1.3 Estimating the derivatives

Sometimes we have function values available, but no derivatives, and we still want a smooth interpolant. In such cases we can still use cubic Hermite interpolation if we can somehow estimate the derivatives. This can be done in many ways, but one common choice is to use the slope of the parabola interpolating the data at three consecutive data-points. To find this slope we observe that the parabola p_i such that $p_i(x_j) = f(x_j)$, for $j = i-1$, i and $i+1$, is given by

$$p_i(x) = f(x_{i-1}) + (x - x_{i-1})\delta_{i-1} + (x - x_{i-1})(x - x_i)\frac{\delta_i - \delta_{i-1}}{\Delta x_{i-1} + \Delta x_i},$$

where

$$\delta_j = \big(f(x_{j+1}) - f(x_j)\big)/\Delta x_j.$$

We then find that

$$p_i'(x_i) = \delta_{i-1} + \Delta x_{i-1}\frac{\delta_i - \delta_{i-1}}{\Delta x_{i-1} + \Delta x_i}.$$

After simplification, we obtain

$$p_i'(x_i) = \frac{\Delta x_{i-1}\delta_i + \Delta x_i \delta_{i-1}}{\Delta x_{i-1} + \Delta x_i}, \quad \text{for } i = 2, \ldots, m-1, \tag{5.11}$$

and this we use as an estimate for $f'(x_i)$. Using cubic Hermite interpolation with the choice (5.11) for derivatives is known as *cubic Bessel interpolation*. It is equivalent to a process known as *parabolic blending*. The end derivatives $f'(x_1)$ and $f'(x_m)$ must be estimated separately. One possibility is to use the value in (5.11) with $x_0 = x_3$ and $x_{m+1} = x_{m-2}$.

5.2 Cubic Spline Interpolation

Cubic Hermite interpolation works well in many cases, but it is inconvenient that the derivatives have to be specified. In Section 5.1.3 we saw one way in which the derivatives can be estimated from the function values. There are many other ways to estimate the derivatives at the data points; one possibility is to demand that the interpolant should have a continuous second derivative at each interpolation point. As we shall see in this section, this leads to a system of linear equations for the unknown derivatives so the locality of the construction is lost, but we gain one more continuous derivative which is important in some applications. A surprising property of this interpolant is that it has the smallest second derivative of all C^2-functions that satisfy the interpolation conditions. The cubic spline interpolant therefore has a number of geometric and physical interpretations that we discuss briefly in Section 5.2.1.

Our starting point is m points $a = x_1 < x_2 < \cdots < x_m = b$ with corresponding values $y_i = f(x_i)$. We are looking for a piecewise cubic polynomial that interpolates the given values and belongs to $C^2[a, b]$. In this construction, it turns out that we need two extra conditions to specify the interpolant uniquely. One of the following boundary conditions is often used.

$$\begin{array}{rll} \text{(i)} & g'(a) = f'(a) \text{ and } g'(b) = f'(b); & \text{H(ermite)} \\ \text{(ii)} & g''(a) = g''(b) = 0; & \text{N(atural)} \\ \text{(iii)} & g''' \text{ is continuous at } x_2 \text{ and } x_{m-1}. & \text{F(ree)} \\ \text{(iv)} & D^j g(a) = D^j g(b) \text{ for } j = 1,\, 2. & \text{P(eriodic)} \end{array} \tag{5.12}$$

The periodic boundary conditions are suitable for closed parametric curves where $f(x_1) = f(x_m)$.

In order to formulate the interpolation problems more precisely, we will define the appropriate spline spaces. Since we want the splines to have continuous derivatives up to order two, we know that all interior knots must be simple. For the boundary conditions H, N, and F, we therefore define the 4-regular knot vectors

$$\begin{aligned} \boldsymbol{t}_H = \boldsymbol{t}_N = (t_i)_{i=1}^{m+6} &= (x_1, x_1, x_1, x_1, x_2, x_3, \ldots, x_{m-1}, x_m, x_m, x_m, x_m), \\ \boldsymbol{t}_F = (t_i)_{i=1}^{m+4} &= (x_1, x_1, x_1, x_1, x_3, x_4, \ldots, x_{m-2}, x_m, x_m, x_m, x_m). \end{aligned} \tag{5.13}$$

This leads to three cubic spline spaces $\mathbb{S}_{3,\boldsymbol{t}_H}$, $\mathbb{S}_{3,\boldsymbol{t}_N}$ and $\mathbb{S}_{3,\boldsymbol{t}_F}$, all of which will have two continuous derivatives at each interior knot. Note that x_2 and x_{m-1} are missing in $\boldsymbol{t}_F$. This means that any $h \in \mathbb{S}_{3,\boldsymbol{t}_F}$ will automatically satisfy the free boundary conditions.

We consider the following interpolation problems.

Problem 5.10. *Let the data $(x_i, f(x_i))_{i=1}^m$ with $a = x_1 < x_2 < \cdots < x_m = b$ be given, together with $f'(x_1)$ and $f'(x_m)$ if they are needed. For Z denoting one of H, N, or F, we seek a spline $g = g_Z = I_Z f$ in the spline space $\mathbb{S}_{3,\boldsymbol{t}_Z}$, such that $g(x_i) = f(x_i)$ for $i = 1, 2, \ldots, m$, and such that boundary condition Z holds.*

We consider first Problem 5.10 in the case of Hermite boundary conditions. Our aim is to show that the problem has a unique solution, and this requires that we study it in some detail.

It turns out that any solution of Problem 5.10 H has a remarkable property. It is the interpolant which, in some sense, has the smallest second derivative. To formulate this, we need to work with integrals of the splines. An interpretation of these integrals is that they are generalizations of the dot product or *inner product* for vectors. Recall that if $\boldsymbol{u}$ and $\boldsymbol{v}$ are two vectors in $\mathbb{R}^n$, then their inner product is defined by

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \boldsymbol{u} \cdot \boldsymbol{v} = \sum_{i=1}^{n} u_i v_i,$$

and the length or *norm* of $\boldsymbol{u}$ can be defined in terms of the inner product as

$$||\boldsymbol{u}|| = \langle \boldsymbol{u}, \boldsymbol{u} \rangle^{1/2} = \Big(\sum_{i=1}^{n} u_i^2 \Big)^{1/2}.$$

The corresponding inner product and norm for functions are

$$\langle u, v\rangle = \int_a^b u(x)v(x)dx = \int_a^b uv$$

and

$$||u|| = \Bigl(\int_a^b u(t)^2 dt\Bigr)^{1/2} = \Bigl(\int_a^b u^2\Bigr)^{1/2}.$$

It therefore makes sense to say that two functions u and v are orthogonal if $\langle u, v\rangle = \int uv = 0$.

The first result that we prove says that the error $f - I_H f$ is orthogonal to a family of linear splines.

Lemma 5.11. *Denote the error in cubic spline interpolation with Hermite end conditions by $e = f - I_H f$, and let $\boldsymbol{t}$ be the 1-regular knot vector*

$$\boldsymbol{t} = (t_i)_{i=1}^{m+2} = (x_1, x_1, x_2, x_3, \ldots, x_{m-1}, x_m, x_m).$$

Then the second derivative of e is orthogonal to the spline space $\mathbb{S}_{1,\boldsymbol{t}}$. In other words

$$\int_a^b e''(x)h(x)\,dx = 0, \quad \textit{for all} \quad h \in \mathbb{S}_{1,\boldsymbol{t}}.$$

Proof. Dividing $[a, b]$ into the subintervals $[x_i, x_{i+1}]$ for $i = 1, \ldots, m-1$, and using integration by parts, we find

$$\int_a^b e''h = \sum_{i=1}^{m-1} \int_{x_i}^{x_{i+1}} e''h = \sum_{i=1}^{m-1} \Bigl(e'h\Big|_{x_i}^{x_{i+1}} - \int_{x_i}^{x_{i+1}} e'h' \Bigr).$$

Since $e'(a) = e'(b) = 0$, the first term is zero,

$$\sum_{i=1}^{m-1} e'h\Big|_{x_i}^{x_{i+1}} = e'(b)h(b) - e'(a)h(a) = 0. \tag{5.14}$$

For the second term, we observe that since h is a linear spline, its derivative is equal to some constant h_i in the subinterval (x_i, x_{i+1}), and therefore can be moved outside the integral. Because of the interpolation conditions we have $e(x_{i+1}) = e(x_i) = 0$, so that

$$\sum_{i=1}^{m-1} \int_{x_i}^{x_{i+1}} e'h' = \sum_{i=1}^{m-1} h_i \int_{x_i}^{x_{i+1}} e'(x)\,dx = 0.$$

This completes the proof. ∎

We can now show that the cubic spline interpolant solves a minimisation problem. In any minimisation problem, we must specify the space over which we minimise. The space in this case is $\mathbb{E}_H(f)$, which is defined in terms of the related space $\mathbb{E}(f)$

$$\begin{aligned} \mathbb{E}(f) &= \bigl\{ g \in C^2[a, b] \mid g(x_i) = f(x_i) \text{ for } i = 1, \ldots, m \bigr\}, \\ \mathbb{E}_H(f) &= \bigl\{ g \in \mathbb{E}(f) \mid g'(a) = f'(a) \text{ and } g'(b) = f'(b) \bigr\}. \end{aligned} \tag{5.15}$$

The space $\mathbb{E}(f)$ is the set of all functions with continuous derivatives up to the second order that interpolate f at the data points. If we restrict the derivatives at the ends to coincide with the derivatives of f we obtain $\mathbb{E}_H(f)$.

The following theorem shows that the second derivative of a cubic interpolating spline has the smallest second derivative of all functions in $\mathbb{E}_H(f)$.

Theorem 5.12. *Suppose that $g = I_H f$ is the solution of Problem 5.10 H. Then*

$$\int_a^b \bigl(g''(x)\bigr)^2\,dx \le \int_a^b \bigl(h''(x)\bigr)^2\,dx \quad \text{for all } h \text{ in } \mathbb{E}_H(f), \tag{5.16}$$

with equality if and only if $h = g$.

Proof. Select some $h \in \mathbb{E}_H(f)$ and set $e = h - g$. Then we have

$$\int_a^b h''^2 = \int_a^b \bigl(e'' + g''\bigr)^2 = \int_a^b e''^2 + 2\int_a^b e''g'' + \int_a^b g''^2. \tag{5.17}$$

Since $g \in \mathbb{S}_{3,\boldsymbol{t}_H}$ we have $g'' \in \mathbb{S}_{1,\boldsymbol{t}}$, where $\boldsymbol{t}$ is the knot vector given in Lemma 5.11. Since $g = I_H h = I_H f$, we have $e = h - I_H h$ so we can apply Lemma 5.11 and obtain $\int_a^b e''g'' = 0$. We conclude that $\int_a^b h''^2 \ge \int_a^b g''^2$.

To show that we can only have equality in (5.16) when $h = g$, suppose that $\int_a^b h''^2 = \int_a^b g''^2$. Using (5.17), we observe that we must have $\int_a^b e''^2 = 0$. But since e'' is continuous, this means that we must have $e'' = 0$. Since we also have $e(a) = e'(a) = 0$, we conclude that $e = 0$. This can be shown by using Taylor's formula

$$e(x) = e(a) + (x-a)e'(a) + \int_a^x e''(t)(x-t)\,dt.$$

Since $e = 0$, we end up with $g = h$. ∎

Lemma 5.11 and Theorem 5.12 allow us to show that the Hermite problem has a unique solution.

Theorem 5.13. *Problem 5.10 H has a unique solution.*

Proof. We seek a function

$$g = I_H f = \sum_{i=1}^{m+2} c_i B_{i,3}$$

in $\mathbb{S}_{3,\boldsymbol{t}_H}$ such that

$$\begin{aligned} \sum_{j=1}^{m+2} c_j B_{j,3}(x_i) &= f(x_i), && \text{for } i = 1, \ldots, m, \\ \sum_{j=1}^{m+2} c_j B'_{j,3}(x_i) &= f'(x_i), && \text{for } i = 1 \text{ and } m. \end{aligned} \tag{5.18}$$

This is a linear system of $m+2$ equations in the $m+2$ unknown B-spline coefficients. From linear algebra we know that such a system has a unique solution if and only if the corresponding system with zero right-hand side only has the zero solution. This means that existence and uniqueness of the solution will follow if we can show that Problem 5.10 H with zero data only has the zero solution. Suppose that $g \in \mathbb{S}_{3,\boldsymbol{t}_H}$ solves Problem 5.10 H with zero data. Clearly $g = 0$ is a solution. According to Theorem 5.12, any other solution must also minimise the integral of the second derivative. By the uniqueness assertion in Theorem 5.12, we conclude that $g = 0$ is the only solution. ∎

We have similar results for the "natural" case.

Lemma 5.14. *If $e = f - I_N f$ and $\boldsymbol{t}$ the knot vector*

$$\boldsymbol{t} = (t_i)_{i=1}^m = (x_1, x_2, x_3, \ldots, x_{m-1}, x_m),$$

the second derivative of e is orthogonal to $\mathbb{S}_{1,\boldsymbol{t}}$,

$$\int_a^b e''(x)h(x)\,dx = 0, \quad \textit{for all } h \textit{ in } \mathbb{S}_{1,\boldsymbol{t}}.$$

Proof. The proof is similar to Lemma 5.11. The relation in (5.14) holds since every $h \in \mathbb{S}_{1,\boldsymbol{t}}$ now satisfies $h(a) = h(b) = 0$. ∎

Lemma 5.14 allows us to prove that the cubic spline interpolation problem with natural boundary conditions has a unique solution.

Theorem 5.15. *Problem 5.10 N has a unique solution $g = I_N f$. The solution is the unique function in $C^2[a,b]$ with the smallest possible second derivative in the sense that*

$$\int_a^b \big(g''(x)\big)^2\,dx \le \int_a^b \big(h''(x)\big)^2\,dx, \quad \textit{for all} \quad h \in \mathbb{E}(f),$$

with equality if and only if $h = g$.

Proof. The proof of Theorem 5.12 carries over to this case. We only need to observe that the natural boundary conditions imply that $g'' \in \mathbb{S}_{1,\boldsymbol{t}}$. ∎

From this it should be clear that the cubic spline interpolants with Hermite and natural end conditions are extraordinary functions. If we consider all continuous functions with two continuous derivatives that interpolate f at the x_i, the cubic spline interpolant with natural end conditions is the one with the smallest second derivative in the sense that the integral of the square of the second derivative is minimised. This explains why the N boundary conditions in (5.12) are called natural. If we restrict the interpolant to have the same derivative as f at the ends, the solution is still a cubic spline.

For the free end interpolant we will show existence and uniqueness in the next section. No minimisation property is known for this spline.

5.2.1 Interpretations of cubic spline interpolation

Today engineers use computers to fit curves through their data points; this is one of the main applications of splines. But splines have been used for this purpose long before computers were available, except that at that time the word spline had a different meaning. In industries like for example ship building, a thin flexible ruler was used to draw curves.

The ruler could be clamped down at fixed data points and would then take on a nice smooth shape that interpolated the data and minimised the bending energy in accordance with the physical laws. This allowed the user to interpolate the data in a visually pleasing way. This flexible ruler was known as a *draftmans spline.*

The physical laws governing the classical spline used by ship designers tell us that the ruler will take on a shape that minimises the total bending energy. The linearised bending energy is given by $\int g''^2$, where $g(x)$ is the position of the centreline of the ruler. Outside the first and last fixing points the ruler is unconstrained and will take the shape of a straight line. From this we see that the natural cubic spline models such a linearised ruler. The word spline was therefore a natural choice for the cubic interpolants we have considered here when they were first studied systematically in 1940's.

The cubic spline interpolant also has a related, geometric interpretation. From differential geometry we know that the curvature of a function $g(x)$ is given by

$$\kappa(x) = \frac{g''(x)}{\Big(1 + (g'(x))^2\Big)^{3/2}}.$$

The curvature $\kappa(x)$ measures how much the function curves at x and is important in the study of parametric curves. If we assume that $1 + g'^2 \approx 1$ on $[a, b]$, then $\kappa(x) \approx g''(x)$. The cubic spline interpolants $I_H f$ and $I_N f$ can therefore be interpreted as the interpolants with the smallest linearised curvature.

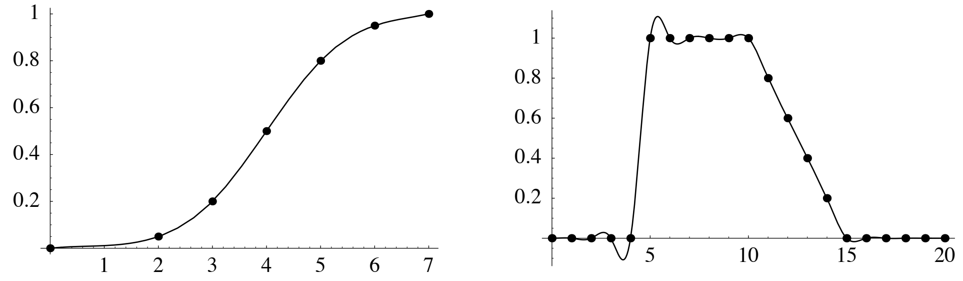
5.2.2 Numerical solution and examples

If we were just presented with the problem of finding the C^2 function that interpolate a given function at some points and have the smallest second derivative, without the knowledge that we obtained in Section 5.2, we would have to work very hard to write a reliable computer program that could solve the problem. With Theorem 5.15, the most difficult part of the work has been done, so that in order to compute the solution to say Problem 5.10 H, we only have to solve the linear system of equations (5.18). Let us take a closer look at this system. We order the equations so that the boundary conditions correspond to the first and last equation, respectively. Because of the local support property of the B-splines, only a few unknowns appear in each equation, in other words we have a banded linear system. Indeed, since $t_{i+3} = x_i$, we see that only $\{B_{j,3}\}_{j=i}^{i+3}$ can be nonzero at x_i. But we note also that x_i is located at the first knot of $B_{i+3,3}$, which means that $B_{i+3,3}(x_i) = 0$. Since we also have $B'_{j,3}(x_1) = 0$ for $j \geq 3$ and $B'_{j,3}(x_m) = 0$ for $j \leq m$, we conclude that the system can be written in the tridiagonal form

$$\boldsymbol{A}\boldsymbol{c} = \begin{pmatrix} \alpha_1 & \gamma_1 & & & \\ \beta_2 & \alpha_2 & \gamma_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{m+1} & \alpha_{m+1} & \gamma_{m+1} \\ & & & \beta_{m+2} & \alpha_{m+2} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{m+1} \\ c_{m+2} \end{pmatrix} = \begin{pmatrix} f'(x_1) \\ f(x_1) \\ \vdots \\ f(x_m) \\ f'(x_m) \end{pmatrix} = \boldsymbol{f}, \tag{5.19}$$

where the elements of $\boldsymbol{A}$ are given by

$$\begin{aligned} \alpha_1 &= B'_{1,3}(x_1), \quad \alpha_{m+2} = B'_{m+2,3}(x_m), \\ \gamma_1 &= B'_{2,3}(x_1), \quad \beta_{m+2} = B'_{m+1,3}(x_m), \\ \beta_{i+1} &= B_{i,3}(x_i), \quad \alpha_{i+1} = B_{i+1,3}(x_i), \quad \gamma_{i+1} = B_{i+2,3}(x_i). \end{aligned} \tag{5.20}$$



(a) (b)

Figure 5.3. Cubic spline interpolation to smoothly varying data (a) and data with sharp corners (b).

The elements of $\boldsymbol{A}$ can be computed by one of the triangular algorithms for B-bases.

For $H_3 f$ we had explicit formulas for the B-spline coefficients that only involved a few function values and derivatives, in other words the approximation was local. In cubic spline interpolation the situation is quite different. All the equations in (5.19) are coupled and we have to solve a linear system of equations. Each coefficient will therefore in general depend on all the given function values which means that the value of the interpolant at a point also depends on all the given function values. This means that cubic spline interpolation is not a local process.

Numerically it is quite simple to solve (5.19). It follows from the proof of Theorem 5.13 that the matrix $\boldsymbol{A}$ is nonsingular, since otherwise the solution could not be unique. Since it has a tridiagonal form it is recommended to use Gaussian elimination. It can be shown that the elimination can be carried out without changing the order of the equations (pivoting), and a detailed error analysis shows that this process is numerically stable .

In most cases, the underlying function f is only known through the data $y_i = f(x_i)$, for $i = 1, \ldots, m$. We can still use Hermite end conditions if we estimate the end slopes $f'(x_1)$ and $f'(x_m)$. A simple estimate is $f'(a) = d_1$ and $f'(b) = d_2$, where

$$d_1 = \frac{f(x_2) - f(x_1)}{x_2 - x_1} \quad \text{and} \quad d_2 = \frac{f(x_m) - f(x_{m-1})}{x_m - x_{m-1}}. \tag{5.21}$$

More elaborate estimates like those in Section 5.1.3 are of course also possible.

Another possibility is to turn to natural and free boundary conditions which also lead to linear systems similar to the one in equation (5.19), except that the first and last equations which correspond to the boundary conditions must be changed appropriately. For natural end conditions we know from Theorem 5.15 that there is a unique solution. Existence and uniqueness of the solution with free end conditions is established in Corollary 5.19.

The free end condition is particularly attractive in a B-spline formulation, since by not giving any knot at x_2 and x_{m-1} these conditions take care of themselves. The free end conditions work well in many cases, but extra wiggles can sometimes occur near the ends of the range. The Hermite conditions give us better control in this respect.

Example 5.16. In Figure 5.3 (a) and 5.3 (b) we show two examples of cubic spline interpolation. In both cases we used the Hermite boundary conditions with the estimate in (5.21) for the slopes. The data to be interpolated is shown as bullets. Note that in Figure 5.3 (a) the interpolant behaves very nicely and predictably between the data points.

In comparison, the interpolant in Figure 5.3 (b) has some unexpected wiggles. This is a characteristic feature of spline interpolation when the data have sudden changes or sharp corners. For such data, least squares approximation by splines usually gives better results, see Section 5.3.2.

5.3 General Spline Approximation

So far, we have mainly considered spline approximation methods tailored to specific degrees. In practise, cubic splines are undoubtedly the most common, but there is an obvious advantage to have methods available for splines of all degrees. In this section we first consider spline interpolation for splines of arbitrary degree. The optimal properties of the cubic spline interpolant can be generalised to spline interpolants of any odd degree, but here we only focus on the practical construction of the interpolant. Least squares approximation, which we study in Section 5.3.2, is a completely different approximation procedure that often give better results than interpolation, especially when the data changes abruptly like in Figure 1.6 (b).

5.3.1 Spline interpolation

Given points $(x_i, y_i)_{i=1}^m$, we again consider the problem of finding a spline g such that

$$g(x_i) = y_i, \qquad i = 1, \ldots, m.$$

In the previous section we used cubic splines where the knots of the spline were located at the data points. This works well if the data points are fairly evenly spaced, but can otherwise give undesirable effects. In such cases the knots should not be chosen at the data points. However, how to choose good knots in general is difficult.

In some cases we might also be interested in doing interpolation with splines of degree higher than three. We could for example be interested in a smooth representation of the second derivative of f. However, if we want f''' to be continuous, say, then the degree d must be higher than three. We therefore consider the following interpolation problem.

Problem 5.17. *Let there be given data $\big(x_i, y_i\big)_{i=1}^m$ and a spline space $\mathbb{S}_{d,\boldsymbol{t}}$ whose knot vector $\boldsymbol{t} = (t_i)_{i=1}^{m+d+1}$ satisfies $t_{i+d+1} > t_i$, for $i = 1, \ldots, m$. Find a spline g in $\mathbb{S}_{d,\boldsymbol{t}}$ such that*

$$g(x_i) = \sum_{j=1}^{m} c_j B_{j,d}(x_i) = y_i, \qquad \text{for } i = 1, \ldots, m. \tag{5.22}$$

The equations in (5.22) form a system of m equations in m unknowns. In matrix form these equations can be written

$$\boldsymbol{Ac} = \begin{pmatrix} B_{1,d}(x_1) & \ldots & B_{m,d}(x_1) \\ \vdots & \ddots & \vdots \\ B_{1,d}(x_m) & \ldots & B_{m,d}(x_m) \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_m \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} = \boldsymbol{y}. \tag{5.23}$$

Theorem 5.18 gives necessary and sufficient conditions for this system to have a unique solution, in other words for $\boldsymbol{A}$ to be nonsingular.

Theorem 5.18. *The matrix $\boldsymbol{A}$ in (5.23) is nonsingular if and only if the diagonal elements $a_{i,i} = B_{i,d}(x_i)$ are positive for $i = 1, \ldots m$.*

Proof. See Theorem 10.6 in Chapter 10. ∎

The condition that the diagonal elements of $\boldsymbol{A}$ should be nonzero can be written

$$t_i < x_i < t_{i+d+1}, \quad i = 1, 2, \ldots, m, \tag{5.24}$$

provided we allow $x_i = t_i$ if $t_i = \cdots = t_{i+d}$. Conditions (5.24) are known as the *Schoenberg-Whitney nesting conditions.*

As an application of Theorem 5.18, let us verify that the coefficient matrix for cubic spline interpolation with free end conditions is nonsingular.

Corollary 5.19. *Cubic spline interpolation with free end conditions (Problem 5.10 F) has a unique solution.*

Proof. The coefficients of the interpolant are found by solving a linear system of equations of the form (5.22). Recall that the knot vector $\boldsymbol{t} = \boldsymbol{t}_F$ is given by

$$\boldsymbol{t} = (t_i)_{i=1}^{m+4} = (x_1, x_1, x_1, x_1, x_3, x_4, \ldots, x_{m-2}, x_m, x_m, x_m, x_m).$$

From this we note that $B_1(x_1)$ and $B_2(x_2)$ are both positive. Since $t_{i+2} = x_i$ for $i = 3$, $\ldots$, $m-2$, we also have $t_i < x_{i-1} < t_{i+4}$ for $3 \leq i \leq m-2$. The last two conditions follow similarly, so the coefficient matrix is nonsingular. ∎

For implementation of general spline interpolation, it is important to make use of the fact that at most $d+1$ B-splines are nonzero for a given x, just like we did for cubic spline interpolation. This means that in any row of the matrix $\boldsymbol{A}$ in (5.22), at most $d+1$ entries are nonzero, and those entries are consecutive. This gives $\boldsymbol{A}$ a band structure that can be exploited in Gaussian elimination. It can also be shown that nothing is gained by rearranging the equations or unknowns in Gaussian elimination, so the equations can be solved without pivoting.

5.3.2 Least squares approximation

In this chapter we have described a number of spline approximation techniques based on interpolation. If it is an absolute requirement that the spline should pass exactly through the data points, there is no alternative to interpolation. But such perfect interpolation is only possible if all computations can be performed without any round-off error. In practise, all computations are done with floating

point numbers, and round-off errors are inevitable. A small error is

therefore always present and must be tolerable whenever computers are used for approximation. The question is what is a tolerable error? Often the data are results of measurements with a certain known resolution. To interpolate such data is not recommended since it means that the error is also approximated. If it is known that the underlying function is smooth, it is usually better to use a method that will only approximate the data, but approximate in such a way that the error at the data points is minimised. Least squares approximation is a general and simple approximation method for accomplishing this. The problem can be formulated as follows.

Problem 5.20. *Given data* $(x_i, y_i)_{i=1}^m$ *with* $x_1 < \cdots < x_m$, *positive real numbers* w_i *for* $i = 1, \ldots, m$, *and an* n*-dimensional spline space* $\mathbb{S}_{d,\boldsymbol{t}}$, *find a spline* g *in* $\mathbb{S}_{d,\boldsymbol{t}}$ *which solves the minimization problem*

$$\min_{h \in \mathbb{S}_{d,\boldsymbol{t}}} \sum_{i=1}^{m} w_i \left(y_i - h(x_i)\right)^2. \tag{5.25}$$

The expression (5.25) that is minimized is a sum of the squares of the errors at each data point, weighted by the numbers w_i which are called *weights*. This explains the name *least squares approximation*, or more precisely *weighted least squares approximation*. If w_i is large in comparison to the other weights, the error $y_i - h(x_i)$ will count more in the minimization. As the the weight grows, the error at this data point will go to zero. On the other hand, if the weight is small in comparison to the other weights, the error at that data point gives little contribution to the total least squares deviation. If the weight is zero, the approximation is completely independent of the data point. Note that the actual value of the weights is irrelevant, it is the relative size that matters. The weights therefore provides us with the opportunity to attach a measure of confidence to each data point. If we know that y_i is a very accurate data value we can give it a large weight, while if y_i is very inaccurate we can give it a small weight. Note that it is the relative size of the weights that matters, a natural 'neutral' value is therefore $w_i = 1$.

From our experience with interpolation, we see that if we choose the spline space $\mathbb{S}_{d,\boldsymbol{t}}$ so that the number of B-splines equals the number of data points and such that $B_i(x_i) > 0$ for all i, then the least squares approximation will agree with the interpolant and give zero error, at least in the absence of round-off errors. Since the

whole point of introducing the least squares approximation is to avoid interpolation of the data, we must make sure that n is smaller than m and that the knot vector is appropriate. This all means that the spline space $\mathbb{S}_{d,\boldsymbol{t}}$ must be chosen appropriately, but this is not easy. Of course we would like the spline space to be such that a "good" approximation g can be found. Good, will have different interpretations for different applications. A statistician would like g to have certain statistical properties. A designer would like an aesthetically pleasing curve, and maybe some other shape and tolerance requirements to be satisfied. In practise, one often starts with a small spline space, and then adds knots in problematic areas until hopefully a satisfactory approximation is obtained.

Different points of view are possible in order to analyse Problem 5.20 mathematically. Our approach is based on linear algebra. Our task is to find the vector $\boldsymbol{c} = (c_1, \ldots, c_n)$ of B-spline coefficients of the spline g solving Problem 5.20. The following matrix-vector formulation is convenient.

Lemma 5.21. *Problem 5.20 is equivalent to the linear least squares problem*

$$\min_{\boldsymbol{c}\in\mathbb{R}^n} \|\boldsymbol{A}\boldsymbol{c} - \boldsymbol{b}\|^2,$$

where $\boldsymbol{A} \in \mathbb{R}^{m,n}$ and $\boldsymbol{b} \in \mathbb{R}^m$ have components

$$a_{i,j} = \sqrt{w_i} B_j(x_i) \qquad \text{and} \qquad b_i = \sqrt{w_i} y_i, \tag{5.26}$$

and for any $\boldsymbol{u} = (u_1, \ldots, u_m)$,

$$\|\boldsymbol{u}\| = \sqrt{u_1^2 + \cdots + u_m^2},$$

is the usual Euclidean length of a vector in $\mathbb{R}^m$.

Proof. Suppose $\boldsymbol{c} = (c_1, \ldots, c_n)$ are the B-spline coefficients of some $h \in \mathbb{S}_{d,\boldsymbol{t}}$. Then

$$\begin{aligned}\|\boldsymbol{A}\boldsymbol{c} - \boldsymbol{b}\|_2^2 &= \sum_{i=1}^{m}\Bigl(\sum_{j=1}^{n} a_{i,j}c_j - b_i\Bigr)^2 \\ &= \sum_{i=1}^{m}\Bigl(\sum_{j=1}^{n} \sqrt{w_i}B_j(x_i)c_j - \sqrt{w_i}y_i\Bigr)^2 \\ &= \sum_{i=1}^{m} w_i\Bigl(h(x_i) - y_i\Bigr)^2.\end{aligned}$$

This shows that the two minimisation problems are equivalent. ∎

In the next lemma, we collect some facts about the general linear least squares problem. Recall that a symmetric matrix $\boldsymbol{N}$ is positive semidefinite if $\boldsymbol{c}^T\boldsymbol{N}\boldsymbol{c} \geq 0$ for all $\boldsymbol{c} \in \mathbb{R}^n$, and positive definite if in addition $\boldsymbol{c}^T\boldsymbol{N}\boldsymbol{c} > 0$ for all nonzero $\boldsymbol{c} \in \mathbb{R}^n$.

Lemma 5.22. *Suppose m and n are positive integers with $m \geq n$, and let the matrix $\boldsymbol{A}$ in $\mathbb{R}^{m,n}$ and the vector $\boldsymbol{b}$ in $\mathbb{R}^m$ be given. The linear least squares problem*

$$\min_{\boldsymbol{c}\in\mathbb{R}^n} \|\boldsymbol{A}\boldsymbol{c} - \boldsymbol{b}\|^2 \tag{5.27}$$

always has a solution $\boldsymbol{c}^*$ which can be found by solving the linear set of equations

$$\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{c}^* = \boldsymbol{A}^T\boldsymbol{b}. \tag{5.28}$$

The coefficient matrix $\boldsymbol{N} = \boldsymbol{A}^T\boldsymbol{A}$ is symmetric and positive semidefinite. It is positive definite, and therefore nonsingular, and the solution of (5.27) is unique if and only if $\boldsymbol{A}$ has linearly independent columns.

Proof. Let span$(\boldsymbol{A})$ denote the n-dimensional linear subspace of $\mathbb{R}^m$ spanned by the columns of $\boldsymbol{A}$,

$$\operatorname{span}(\boldsymbol{A}) = \{\boldsymbol{A}\boldsymbol{c} \mid \boldsymbol{c} \in \mathbb{R}^n\}.$$

From basic linear algebra we know that a vector $\boldsymbol{b} \in \mathbb{R}^m$ can be written uniquely as a sum $\boldsymbol{b} = \boldsymbol{b}_1 + \boldsymbol{b}_2$, where $\boldsymbol{b}_1$ is a linear combination of the columns of $\boldsymbol{A}$ so that $\boldsymbol{b}_1 \in \operatorname{span}(\boldsymbol{A})$, and $\boldsymbol{b}_2$ is orthogonal to span$(\boldsymbol{A})$, i.e., we have $\boldsymbol{b}_2^T\boldsymbol{d} = 0$ for all $\boldsymbol{d}$ in span$(\boldsymbol{A})$. Using this decomposition of $\boldsymbol{b}$, and the Pythagorean theorem, we have for any $\boldsymbol{c} \in \mathbb{R}^n$,

$$\|\boldsymbol{A}\boldsymbol{c} - \boldsymbol{b}\|^2 = \|\boldsymbol{A}\boldsymbol{c} - \boldsymbol{b}_1 - \boldsymbol{b}_2\|^2 = \|\boldsymbol{A}\boldsymbol{c} - \boldsymbol{b}_1\|^2 + \|\boldsymbol{b}_2\|^2.$$

It follows that $\|\boldsymbol{A}\boldsymbol{c} - \boldsymbol{b}\|_2^2 \geq \|\boldsymbol{b}_2\|_2^2$ for any $\boldsymbol{c} \in \mathbb{R}^n$, with equality if $\boldsymbol{A}\boldsymbol{c} = \boldsymbol{b}_1$. A $\boldsymbol{c} = \boldsymbol{c}^*$ such that $\boldsymbol{A}\boldsymbol{c}^* = \boldsymbol{b}_1$ clearly exists since $\boldsymbol{b}_1$ is in span$(\boldsymbol{A})$, and $\boldsymbol{c}^*$ is unique if and only if $\boldsymbol{A}$ has linearly independent columns. To derive the linear system for $\boldsymbol{c}^*$, we note that any $\boldsymbol{c}$ that is minimising satisfies $\boldsymbol{A}\boldsymbol{c} - \boldsymbol{b} = -\boldsymbol{b}_2$. Since we also know that $\boldsymbol{b}_2$ is orthogonal to span$(\boldsymbol{A})$, we must have

$$\boldsymbol{d}^T(\boldsymbol{A}\boldsymbol{c} - \boldsymbol{b}) = \boldsymbol{c}_1^T\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{c} - \boldsymbol{b}) = 0$$

for all $\boldsymbol{d} = \boldsymbol{A}\boldsymbol{c}_1$ in span$(\boldsymbol{A})$, i.e., for all $\boldsymbol{c}_1$ in $\mathbb{R}^n$. But this is only possible if $\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{c}-\boldsymbol{b}) = 0$. This proves (5.28).

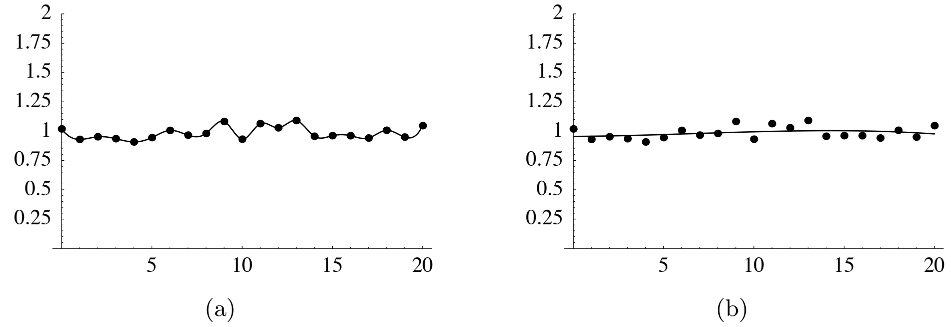


Figure 5.4. Figure (a) shows the cubic spline interpolation to the noisy data of Example 5.24, while least squares approximation to the same data is shown in (b).

The $n \times n$-matrix $\boldsymbol{N} = \boldsymbol{A}^T\boldsymbol{A}$ is clearly symmetric and

$$\boldsymbol{c}^T\boldsymbol{N}\boldsymbol{c} = \|\boldsymbol{A}\boldsymbol{c}\|_2^2 \geq 0, \tag{5.29}$$

for all $\boldsymbol{c} \in \mathbb{R}^n$, so that $\boldsymbol{N}$ is positive semi-definite. From (5.29) we see that we can find a nonzero $\boldsymbol{c}$ such that $\boldsymbol{c}^T\boldsymbol{N}\boldsymbol{c} = 0$ if and only if $\boldsymbol{A}\boldsymbol{c} = 0$, i.e., if and only if $\boldsymbol{A}$ has linearly dependent columns . We conclude that $\boldsymbol{N}$ is positive definite if and only if $\boldsymbol{A}$ has linearly independent columns. ∎

Applying these results to Problem 5.20 we obtain.

Theorem 5.23. *Problem 5.20 always has a solution. The solution is unique if and only if we can find a sub-sequence $(x_{i_\ell})_{\ell=1}^n$ of the data abscissa such that*

$$B_\ell(x_{i_\ell}) \neq 0 \qquad \text{for } \ell = 1, \ldots, n.$$

Proof. By Lemma 5.21 and Lemma 5.22 we conclude that Problem 5.20 always has a solution, and the solution is unique if and only if the matrix $\boldsymbol{A}$ in Lemma 5.21 has linearly independent columns. Now $\boldsymbol{A}$ has linearly independent columns if and only if we can find a subset of n rows of $\boldsymbol{A}$ such that the square submatrix consisting of these rows and all columns of $\boldsymbol{A}$ is nonsingular. But such a matrix is of the form treated in Theorem 5.18. Therefore, the submatrix is nonsingular if and only if the diagonal elements are nonzero. But the diagonal elements are given by $B_\ell(x_{i_\ell})$. ∎

Theorem 5.23 provides a nice condition for checking that we have a unique least squares spline approximation to a given data set; we just have to check that each B-spline has its 'own' x_{i_ℓ} in its support. To find the B-spline coefficients of the approximation, we must solve the linear system of equations (5.28). These equations are called the *normal equations* of the least squares system and can be solved by Cholesky factorisation of a banded matrix followed by back substitution. The least squares problem can also be solved by computing a QR-factorisation of the matrix $\boldsymbol{A}$; for both methods we refer to a standard text on numerical linear algebra for details.

Example 5.24. Least squares approximation is especially appropriate when the data is known to be noisy. Consider the data represented as bullets in Figure 5.4 (a). These data were obtained by adding random perturbations in the interval $[-0.1, 0.1]$ to the function $f(x) = 1$. In Figure 5.4 (a) we show the cubic spline interpolant (with free end conditions) to the data, while Figure 5.4 (b) shows the cubic least squares approximation to the same data, using no interior knots. We see that the least squares approximation smooths out the data nicely. We also see that the cubic spline interpolant gives a nice approximation to the given data, but it also reproduces the noise that was added artificially.

Once we have made the choice of approximating the data in Example 5.24 using cubic splines with no interior knots, we have no chance of representing the noise in the data. The flexibility of cubic polynomials is nowhere near rich enough to represent all the oscillations that we see in Figure 5.4 (a), and this gives us the desired smoothing effect in Figure 5.4 (b). The advantage of the method of least squares is that it gives a reasonably simple method for computing a reasonably good approximation to quite arbitrary data on quite arbitrary knot vectors. But it is largely the knot vector that decides how much the approximation is allowed to oscillate, and good methods for choosing the knot vector is therefore of fundamental importance. Once the knot vector is given there are in fact many approximation methods that will provide good approximations.

5.4 The Variation Diminishing Spline Approximation

In this section we describe a simple, but very useful method for obtaining spline approximations to a function f defined on an interval $[a, b]$. This method is a generalisation of piecewise linear interpolation and has a nice shape preserving behaviour. For example, if the function f is positive, then the spline approximation will also be positive.

Definition 5.25. *Let f be a given continuous function on the interval $[a, b]$, let d be a given positive integer, and let $\boldsymbol{t} = (t_1, \ldots, t_{n+d+1})$ be a $d+1$-regular knot vector with boundary knots $t_{d+1} = a$ and $t_{n+1} = b$. The spline given by*

$$(Vf)(x) = \sum_{j=1}^{n} f(t_j^*) B_{j,d}(x) \tag{5.30}$$

where $t_j^* = (t_{j+1} + \cdots + t_{j+d})/d$ are the knot averages, is called the Variation Diminishing Spline Approximation *of degree d to f on the knot vector $\boldsymbol{t}$.*

The approximation method that assigns to f the spline approximation Vf is about the simplest method of approximation that one can imagine. Unlike some of the other methods discussed in this chapter there is no need to solve a linear system. To obtain Vf, we simply evaluate f at certain points and use these function values as B-spline coefficients directly.

Note that if all interior knots occur less than $d+1$ times in $\boldsymbol{t}$, then

$$a = t_1^* < t_2^* < \ldots < t_{n-1}^* < t_n^* = b. \tag{5.31}$$

This is because t_1 and t_{n+d+1} do not occur in the definition of t_1^* and t_n^* so that all selections of d consecutive knots must be different.

Example 5.26. Suppose that $d = 3$ and that the interior knots of $\boldsymbol{t}$ are uniform in the interval $[0, 1]$, say

$$\boldsymbol{t} = (0, 0, 0, 0, 1/m, 2/m, \ldots, 1 - 1/m, 1, 1, 1, 1). \tag{5.32}$$

For $m \geq 2$ we then have

$$\boldsymbol{t}^* = (0, 1/(3m), 1/m, 2/m, \ldots, 1 - 1/m, 1 - 1/(3m), 1). \tag{5.33}$$

Figure 5.5 (a) shows the cubic variation diminishing approximation to the exponential function on the knot vector in (5.32) with $m = 5$, and the error is shown in Figure 5.5 (b). The error is so small that it is difficult to distinguish between the two functions in Figure 5.5 (a).

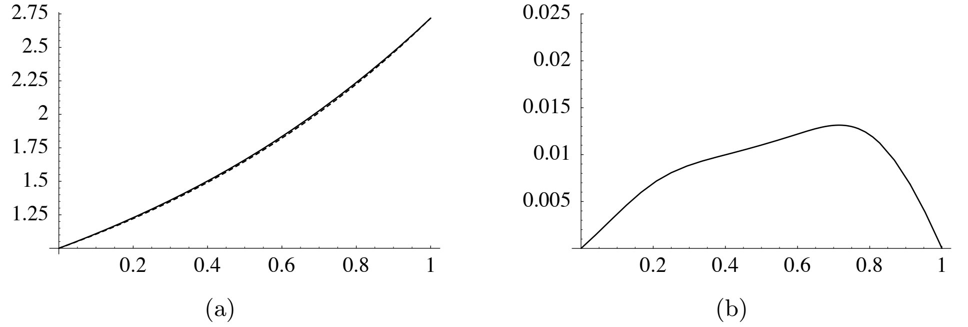


Figure 5.5. The exponential function together with the cubic variation diminishing approximation of Example 5.26 in the special case $m = 5$ is shown in (a). The error in the approximation is shown in (b).

The variation diminishing spline can also be used to approximate functions with singularities, that is, functions with discontinuities in a derivative of first or higher orders.

Example 5.27. Suppose we want to approximate the function

$$f(x) = 1 - e^{-50|x|}, \qquad x \in [-1, 1], \tag{5.34}$$

by a cubic spline Vf. In order to construct a suitable knot vector, we take a closer look at the function, see Figure 5.6 (a). The graph of f has a cusp at the origin, so f' is discontinuous and changes sign there. Our spline approximation should therefore also have some kind of singularity at the origin. Recall from Theorem 3.19 that a B-spline can have a discontinuous first derivative at a knot provided the knot has multiplicity at least d. Since we are using cubic splines, we therefore place a triple knot at the origin. The rest of the interior knots are placed uniformly in $[-1, 1]$. A suitable knot vector is therefore

$$\boldsymbol{t} = (-1, -1, -1, -1, -1 + 1/m, \ldots, -1/m, 0, 0, 0, 1/m, \ldots, 1 - 1/m, 1, 1, 1, 1). \tag{5.35}$$

The integer m is a parameter which is used to control the number of knots and thereby the accuracy of the approximation. The spline Vf is shown in Figure 5.6 (a) for $m = 4$ together with the function f itself. The error is shown in Figure 5.6 (b), and we note that the error is zero at $x = 0$, but quite large just outside the origin.

Figures 5.6 (c) and 5.6 (d) show the first and second derivatives of the two functions, respectively. Note that the sign of f and its derivatives seem to be preserved by the variation diminishing spline approximation.

The variation diminishing spline approximation is a very simple procedure for obtaining spline approximations. In Example 5.27 we observed that the approximation has the same sign as f everywhere, and more than this, even the sign of the first two derivatives is preserved in passing from f to the approximation Vf. This is important since the sign of the derivative gives important information about the shape of the graph of the function. A nonnegative derivative for example, means that the function is nondecreasing, while a nonnegative second derivative roughly means that the function is convex, in other words it curves in the same direction everywhere. Approximation methods that preserve the sign of the derivative are therefore important in practical modelling of curves. We will now study such *shape preservation* in more detail.

5.4.1 Preservation of bounds on a function

Sometimes it is important that the maximum and minimum values of a function are preserved under approximation. Splines have some very useful properties in this respect.

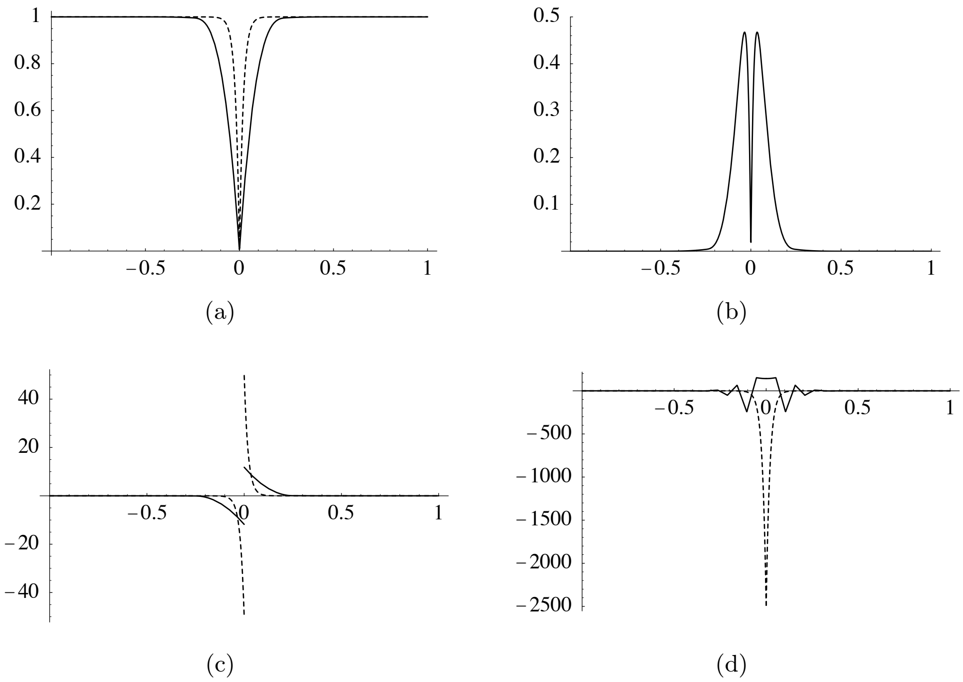


Figure 5.6. Figure (a) shows the function $f(x) = 1 - e^{-50|x|}$ (dashed) and its cubic variation diminishing spline approximation (solid) on the knot vector described in Example 5.27, and the error in the approximation is shown in Figure (b). The first derivative of the two functions is shown in (c), and the second derivatives in (d).

Lemma 5.28. *Let g be a spline in some spline space $\mathbb{S}_{d,\boldsymbol{t}}$ of dimension n. Then g is bounded by its smallest and largest B-spline coefficients,*

$$\min_i\{c_i\} \leq \sum_i c_i B_i(x) \leq \max_i\{c_i\}, \qquad \text{for all } x \in [t_{d+1}, t_{n+1}). \tag{5.36}$$

Proof. Let $c_{\max}$ be the largest coefficient. Then we have

$$\sum_i c_i B_i(x) \leq \sum_i c_{\max} B_i(x) = c_{\max} \sum_i B_i(x) = c_{\max},$$

since $\sum_i B_i(x) = 1$. This proves the second inequality in (5.36). The proof of the first inequality is similar. ∎

Note that this lemma only says something interesting if $n \geq d+1$. Any plot of a spline with its control polygon will confirm the inequalities (5.36), see for example Figure 5.7.

With Lemma 5.28 we can show that bounds on a function are preserved by its variation diminishing approximation.

Proposition 5.29. *Let f be a function that satisfies*

$$f_{\min} \leq f(x) \leq f_{\max} \qquad \text{for all } x \in \mathbb{R}.$$

Then the variation diminishing spline approximation to f from some spline space $\mathbb{S}_{d,\boldsymbol{t}}$ has the same bounds,

$$f_{\min} \leq (Vf)(x) \leq f_{\max} \qquad \text{for all } x \in \mathbb{R}. \tag{5.37}$$

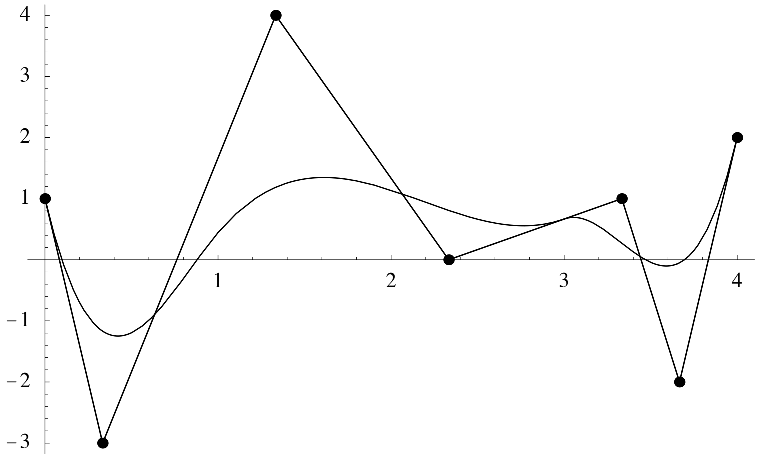


Figure 5.7. A cubic spline with its control polygon. Note how the extrema of the control polygon bound the extrema of the spline.

Proof. Recall that the B-spline coefficients c_i of Vf are given by

$$c_i = f(t_i^*) \qquad \text{where } t_i^* = (t_{i+1} + \cdots + t_{i+d})/d.$$

We therefore have that all the B-spline coefficients of Vf are bounded below by $f_{\min}$ and above by $f_{\max}$. The inequalities in (5.37) therefore follow as in Lemma 5.28. ∎

5.4.2 Preservation of monotonicity

Many geometric properties of smooth functions can be characterised in terms of the derivative of the function. In particular, the sign of the derivative tells us whether the function is increasing or decreasing. The variation diminishing approximation also preserves information about the derivatives in a very convenient way. Let us first define exactly what we mean by increasing and decreasing functions.

Definition 5.30. *A function f defined on an interval $[a, b]$ is increasing if the inequality $f(x_0) \leq f(x_1)$ holds for all x_0 and x_1 in $[a, b]$ with $x_0 < x_1$. It is decreasing if $f(x_0) \geq f(x_1)$ for all x_0 and x_1 in $[a, b]$ with $x_0 < x_1$. A function that is increasing or decreasing is said to be monotone.*

The two properties of being increasing and decreasing are clearly completely symmetric and we will only prove results about increasing functions.

If f is differentiable, monotonicity can be characterized in terms of the derivative.

Proposition 5.31. *A differentiable function is increasing if and only if its derivative is nonnegative.*

Proof. Suppose that f is increasing. Then $(f(x + h) - f(x))/h \geq 0$ for all x and positive h such that both x and $x + h$ are in $[a, b]$. Taking the limit as h tends to zero, we must have $f'(x) \geq 0$ for an increasing differentiable function. At $x = b$ a similar argument with negative h may be used.

If the derivative of f is nonnegative, let x_0 and x_1 be two distinct points in $[a, b]$ with $x_0 < x_1$. The mean value theorem then states that

$$\frac{f(x_1) - f(x_0)}{x_1 - x_0} = f'(\theta)$$

for some $\theta \in (x_0, x_1)$. Since $f'(\theta) \geq 0$, we conclude that $f(x_1) \geq f(x_0)$. ∎

Monotonicity of a spline can be characterized in terms of some simple conditions on its B-spline coefficients.

Proposition 5.32. *Let $\boldsymbol{t}$ be a $d+1$-extended knot vector for splines on the interval $[a, b] = [t_{d+1}, t_{n+1}]$, and let $g = \sum_{i=1}^{n} c_i B_i$ be a spline in $\mathbb{S}_{d,\boldsymbol{t}}$. If the coefficients are increasing, that is $c_i \leq c_{i+1}$ for $i = 1, \ldots, n-1$, then g is increasing. If the coefficients are decreasing then g is decreasing.*

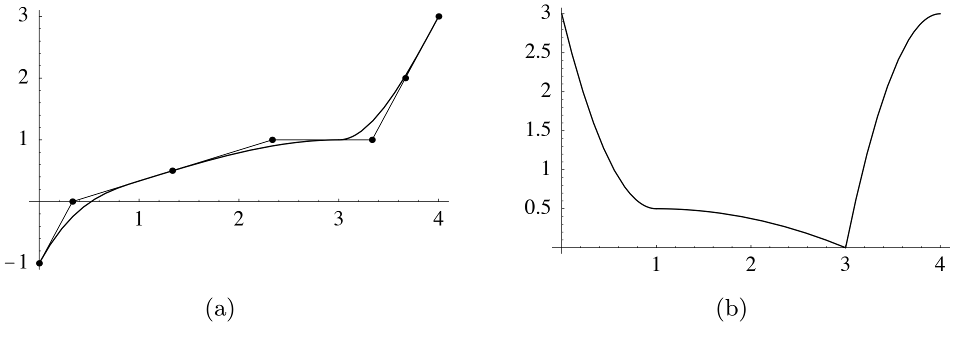
Proof. The proposition is certainly true for $d = 0$, so we can assume that $d \geq 1$. Suppose first that there are no interior knots in $\boldsymbol{t}$ of multiplicity $d+1$. If we differentiate g we find $g'(x) = \sum_{i=1}^{n} \Delta c_i B_{i,d-1}(x)$ for $x \in [a, b]$, where

$$\Delta c_i = d \frac{c_i - c_{i-1}}{t_{i+d} - t_i}.$$

Since all the coefficients of g' are nonnegative we must have $g'(x) \geq 0$ (really the one sided derivative from the right) for $x \in [a, b]$. Since we have assumed that there are no knots of multiplicity $d+1$ in (a, b), we know that g is continuous and hence that it must be increasing.

If there is an interior knot at $t_i = t_{i+d}$ of multiplicity $d+1$, we conclude from the above that g is increasing on both sides of t_i. But we also know that $g(t_i) = c_i$ while the limit of g from the left is c_{i-1}. The jump is therefore $c_i - c_{i-1}$ which is nonnegative so g increases across the jump. ∎

An example of an increasing spline with its control polygon is shown in Figure 5.8 (a). Its derivative is shown in Figure 5.8 (b) and is, as expected, positive.



(a) (b)

Figure 5.8. An increasing cubic spline (a) and its derivative (b).

From Proposition 5.32 it is now easy to deduce that Vf preserves monotonicity in f.

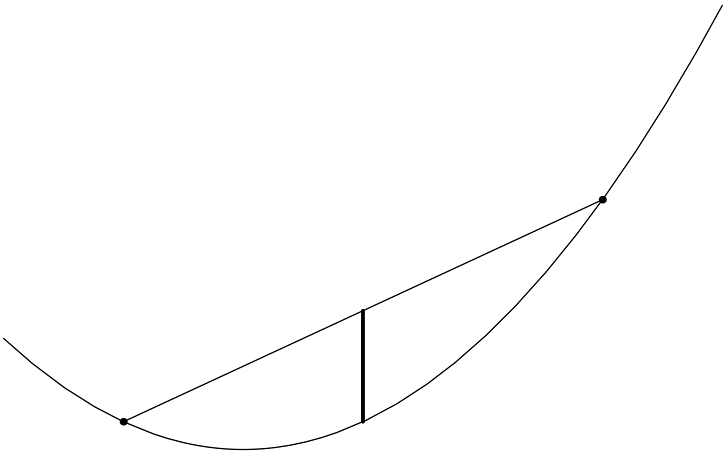


Figure 5.9. A convex function and the cord connecting two points on the graph.

Proposition 5.33. *Let f be function defined on the interval $[a, b]$ and let $\boldsymbol{t}$ be a $d+1$-extended knot vector with $t_{d+1} = a$ and $t_{n+1} = b$. If f is increasing (decreasing) on $[a, b]$, then the variation diminishing approximation Vf is also increasing (decreasing) on $[a, b]$.*

Proof. The variation diminishing approximation Vf has as its i'th coefficient $c_i = f(t_i^*)$, and since f is increasing these coefficients are also increasing. Proposition 5.32 then shows that Vf is increasing. ∎

That Vf preserves monotonicity means that the oscillations we saw could occur in spline interpolation are much less pronounced in the variation diminishing spline approximation. In fact, we shall also see that Vf preserves the sign of the second derivative of f which reduces further the possibility of oscillations.

5.4.3 Preservation of convexity

From elementary calculus, we know that the sign of the second derivative of a function tells us in whether the function curves upward or downwardsupward, or whether the function is *convex* or *concave*. These concepts can be defined for functions that have no a priori smoothness.

Definition 5.34. *A function f is said to be convex on an interval (a, b) if*

$$f\big((1-\lambda)x_0 + \lambda x_2\big) \leq (1-\lambda)f(x_0) + \lambda f(x_2) \tag{5.38}$$

for all x_0 and x_2 in $[a, b]$ and for all λ in $[0, 1]$. If $-f$ is convex then f is said to be concave.

From Definition 5.34 we see that f is convex if the line between two points on the graph of f is always above the graph, see Figure 5.9. It therefore 'curves upward' just like smooth functions with nonnegative second derivative.

Convexity can be characterised in many different ways, some of which are listed in the following lemma.

Lemma 5.35. *Let f be a function defined on the open interval (a, b).*

1. *The function f is convex if and only if*
$$\frac{f(x_1)-f(x_0)}{x_1-x_0} \le \frac{f(x_2)-f(x_1)}{x_2-x_1} \tag{5.39}$$
for all x_0, x_1 and x_2 in (a,b) with $x_0 < x_1 < x_2$.

2. *If f is differentiable on (a,b), it is convex if and only if $f'(y_0) \le f'(y_1)$ when $a < y_0 < y_1 < b$, that is, the derivative of f is increasing.*

3. *If f is two times differentiable it is convex if and only if $f''(x) \ge 0$ for all x in (a,b).*

Proof. Let $\lambda = (x_1-x_0)/(x_2-x_0)$ so that $x_1 = (1-\lambda)x_0+\lambda x_2$. Then (5.38) is equivalent to the inequality
$$(1-\lambda)\bigl(f(x_1)-f(x_0)\bigr) \le \lambda\bigl(f(x_2)-f(x_1)\bigr).$$
Inserting the expression for λ gives (5.39), so (i) is equivalent to Definition 5.34.

To prove (ii), suppose that f is convex and let y_0 and y_1 be two points in (a,b) with $y_0 < y_1$. From (5.39) we deduce that
$$\frac{f(y_0)-f(x_0)}{y_0-x_0} \le \frac{f(y_1)-f(x_1)}{y_1-x_1},$$
for any x_0 and x_1 with $x_0 < y_0 < x_1 < y_1$. Letting x_0 and x_1 tend to y_0 and y_1 respectively, we see that $f'(y_0) \le f'(y_1)$.

For the converse, suppose that f' is increasing, and let $x_0 < x_1 < x_2$ be three points in (a,b). By the mean value theorem we have
$$\frac{f(x_1)-f(x_0)}{x_1-x_0} = f'(\theta_0) \qquad \text{and} \qquad \frac{f(x_2)-f(x_1)}{x_2-x_1} = f'(\theta_1),$$
where $x_0 < \theta_0 < x_1 < \theta_1 < x_2$. Since f' is increasing, conclude that (5.39) holds and therefore that f is convex.

For part (iii) we use part (ii) and Proposition 5.31. From (ii) we know that f is convex if and only if f' is increasing, and by Proposition 5.31 we know that f' is increasing if and only if f'' is nonnegative. ∎

It may be a bit confusing that the restrictions on $x_0 < x_1 < x_2$ in Lemma 5.35 are stronger than the restrictions on x_0, x_2 and λ in Definition 5.34. But this is only superficial since in the special cases $x_0 = x_2$, and $\lambda = 0$ and $\lambda = 1$, the inequality (5.38) is automatically satisfied.

It is difficult to imagine a discontinuous convex function. This is not so strange since all convex functions are in fact continuous.

Proposition 5.36. *A function that is convex on an open interval is continuous on that interval.*

Proof. Let f be a convex function on (a,b), and let x and y be two points in some subinterval (c,d) of (a,b). Using part (i) of Lemma 5.35 repeatedly, we find that
$$\frac{f(c)-f(a)}{c-a} \le \frac{f(y)-f(x)}{y-x} \le \frac{f(b)-f(d)}{b-d}. \tag{5.40}$$

Set $M = \max\{(f(c) - f(a))/(c - a), (f(b) - f(d))/(b - d)\}$. Then (5.40) is equivalent to

$$|f(y) - f(x)| \leq M|y - x|.$$

But this means that f is continuous at each point in (c, d). For if z is in (c, d) we can choose $x = z$ and $y > z$ and obtain that f is continuous from the right at z. Similarly, we can also choose $y = z$ and $x < z$ to find that f is continuous from the left as well. Since (c, d) was arbitrary in (a, b), we have showed that f is continuous in all of (a, b). ∎

The assumption in Proposition 5.36 that f is defined on an open interval is essential. On the interval $(0, 1]$ for example, the function f that is identically zero except that $f(1) = 1$, is convex, but clearly discontinuous at $x = 1$. For splines however, this is immaterial if we assume a spline to be continuous from the right at the left end of the interval of interest and continuous from the left at the right end of the interval of interest. In addition, since a polynomial never is infinite, we see that our results in this section remain true for splines defined on some closed interval $[a, b]$.

We can now give a simple condition that ensures that a spline function is convex.

Proposition 5.37. *Let $\boldsymbol{t}$ be a $d + 1$-extended knot vector for some $d \geq 1$, and let $g = \sum_{i=1}^{n} c_i B_{i,d}$ be a spline in $\mathbb{S}_{d,\boldsymbol{t}}$. Define Δc_i by*

$$\Delta c_i = \begin{cases} (c_i - c_{i-1})/(t_{i+d} - t_i), & \text{if } t_i < t_{i+d}, \\ \Delta c_{i-1}, & \text{if } t_i = t_{i+d}; \end{cases}$$

for $i = 2, \ldots, n$. Then g is convex on $[t_{d+1}, t_{n+1}]$ if it is continuous and

$$\Delta c_{i-1} \leq \Delta c_i \qquad \text{for } i = 2, \ldots, n. \tag{5.41}$$

Proof. Note that $(\Delta c_i)_{i=2}^{n}$ are the B-spline coefficients of g' on the interval $[t_{d+1}, t_{n+1}]$, bar the constant d. Since (5.41) ensures that these are increasing, we conclude from Proposition 5.32 that g' is increasing. If g is also differentiable everywhere in $[t_{d+1}, t_{n+1}]$, part (ii) of Lemma 5.35 shows that g is convex.

In the rest of the proof, the short hand notation

$$\delta(u, v) = \frac{g(v) - g(u)}{v - u}$$

will be convenient. Suppose now that there is only one point z where g is not differentiable, and let $x_0 < x_1 < x_2$ be three points in $[t_{d+1}, t_{n+1}]$. We must show that

$$\delta(x_0, x_1) \leq \delta(x_1, x_2). \tag{5.42}$$

The case where all three x's are on one side of z are covered by the first part of the proof. Suppose therefore that $x_0 < z \leq x_1 < x_2$. Since $\delta(u, v) = g'(\theta)$ with $u < \theta < v$ when g is differentiable on $[a, b]$, and since g' is increasing, we certainly have $\delta(x_0, z) \leq \delta(z, x_2)$, so that (5.42) holds in the special case where $x_1 = z$. When $x_1 > z$ we use the simple identity

$$\delta(x_0, x_1) = \delta(x_0, z)\frac{z - x_0}{x_1 - x_0} + \delta(z, x_1)\frac{x_1 - z}{x_1 - x_0},$$

which shows that $\delta(x_0, x_1)$ is a weighted average of $\delta(x_0, z)$ and $\delta(z, x_1)$. But then we have

$$\delta(x_0, x_1) \leq \delta(z, x_1) \leq \delta(x_1, x_2),$$

the first inequality being valid since $\delta(x_0, z) \leq \delta(z, x_1)$ and the second one because g is convex to the right of z. This shows that g is convex.

The case where $x_0 < x_1 < z < x_2$ and the case of several discontinuities can be treated similarly. ∎

An example of a convex spline is shown in Figure 5.10, together with its first and second derivatives in.

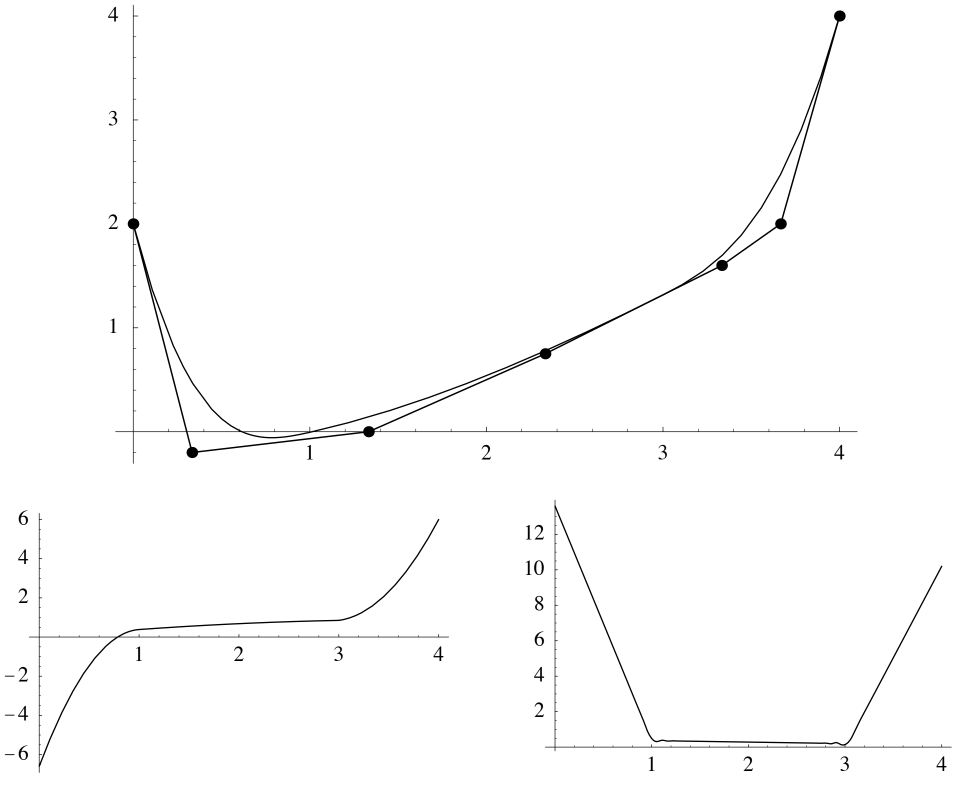


Figure 5.10. A convex spline with its control polygon (a), its first derivative (b) and its second derivative (c).

With Proposition 5.37 at hand, it is simple to show that the variation diminishing spline approximation preserves convexity.

Proposition 5.38. *Let f be a function defined on the interval $[a, b]$, let $d \geq 1$ be an integer, and let $\boldsymbol{t}$ be a $d+1$-extended knot vector with $t_{d+1} = a$ and $t_{n+1} = b$. If f is convex on $[a, b]$ then Vf is also convex on $[a, b]$.*

Proof. Recall that the coefficients of Vf are $\big(f(t_i^*)\big)_{i=1}^n$ so that the differences in Proposition 5.37 are

$$\Delta c_i = \frac{f(t_i^*) - f(t_{i-1}^*)}{t_{i+d} - t_i} = \frac{f(t_i^*) - f(t_{i-1}^*)}{(t_i^* - t_{i-1}^*)d},$$

if $t_i < t_{i+d}$. Since f is convex, these differences must be increasing. Proposition 5.37 then shows that Vf is convex. ∎

At this point, we can undoubtedly say that the variation diminishing spline approximation is a truly remarkable method of approximation. In spite of its simplicity, it preserves the shape of f both with regards to convexity, monotonicity and bounds on the function values. This makes it very attractive as an approximation method in for example design where the shape of a curve is more important than how accurately it approximates given data.

It should be noted that the shape preserving properties of the variation diminishing approximation is due to the properties of B-splines. When we determine Vf we give its control polygon directly by sampling f at the knot averages, and the results that we have obtained about the shape preserving properties of Vf are all consequences of relationships between a spline and its control polygon: *A spline is bounded by the extrema of its control polygon, a spline is monotone if the control polygon is monotone, a spline is convex if the control polygon is convex.* In short: *A spline is a smoothed out version of its control polygon.* We will see many more realisations of this general principle in later chapters