6.11 *Moving-Average Filters with Prescribed Moments.* The predictive FIR filter of Eq. (6.16.3) has lag equal to $\bar{n} = -\tau$ by design. Show that its second moment is not independently specified but is given by,

$$\overline{n^2} = \sum_{n=0}^{N-1} n^2 h(n) = -\frac{1}{6}(N-1)(N-2+6\tau) \tag{6.25.5}$$

The construction of the predictive filters (6.16.3) can be generalized to allow arbitrary specification of the first and second moments, that is, the problem is to design a length-N FIR filter with the prescribed moments,

$$\overline{n^0} = \sum_{n=0}^{N-1} h(n) = 1\,, \quad \overline{n^1} = \sum_{n=0}^{N-1} n\,h(n) = -\tau_1\,, \quad \overline{n^2} = \sum_{n=0}^{N-1} n^2 h(n) = \tau_2 \tag{6.25.6}$$

Show that such filter is given by an expression of the form,

$$h(n) = c_0 + c_1 n + c_2 n^2\,, \quad n = 0, 1, \dots, N-1$$

where the coefficients c_0, c_1, c_2 are the solutions of the linear system,

$$\begin{bmatrix} S_0 & S_1 & S_2 \\ S_1 & S_2 & S_3 \\ S_2 & S_3 & S_4 \end{bmatrix} \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -\tau_1 \\ \tau_2 \end{bmatrix}$$

where

$$S_p = \sum_{n=0}^{N-1} n^p\,, \quad p = 0, 1, 2, 3, 4$$

Then, show that the S_p are given explicitly by,

$$S_0 = N\,, \quad S_1 = \frac{1}{2}N(N-1)\,, \quad S_2 = \frac{1}{6}N(N-1)(2N-1)$$

$$S_3 = \frac{1}{4}N^2(N-1)^2\,, \quad S_4 = \frac{1}{30}N(N-1)(2N-1)(3N^2-3N-1)$$

and that the coefficients are given by,

$$c_0 = \frac{3(3N^2-3N+2) + 18(2N-1)\tau_1 + 30\tau_2}{N(N+1)(N+2)}$$

$$c_1 = -\frac{18(N-1)(N-2)(2N-1) + 12(2N-1)(8N-11)\tau_1 + 180(N-1)\tau_2}{N(N^2-1)(N^2-4)}$$

$$c_2 = \frac{30(N-1)(N-2) + 180(N-1)\tau_1 + 180\tau_2}{N(N^2-1)(N^2-4)}$$

Finally, show that the condition $c_2 = 0$ recovers the predictive FIR case of Eq. (6.16.3) with second moment given by Eq. (6.25.5).

6.12 Consider the Butterworth filter of Eq. (6.20.2). Show that the lag of the first-order section and the lag of the ith second-order section are given by,

$$\bar{n}_0 = \frac{1}{2\Omega_0}\,, \quad \bar{n}_i = -\frac{\cos\theta_i}{\Omega_0}\,, \quad i = 1, 2, \dots, K$$

Using these results, prove Eq. (6.20.8) for the full lag $\bar{n}$, and show that it is valid for both even and odd filter orders M.

7

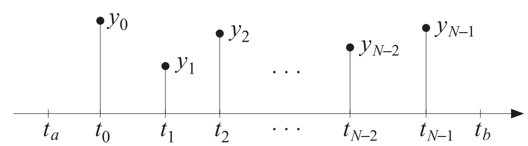
Smoothing Splines

7.1 Interpolation versus Smoothing

Besides their extensive use in drafting and computer graphics, splines have many other applications. A large online bibliography can be found in [350]. A small subset of references on interpolating and smoothing splines and their applications is [351–404].

We recall from Sec. 4.2 that the minimum-R_s filters had the property of maximizing the smoothness of the filtered output signal by minimizing the mean-square value of the s-differenced output, that is, the quantity $E\big[(\nabla^s \hat{x}_n)^2\big]$ in the notation of Eq. (4.2.11). Because of their finite span, minimum-R_s filters belong to the class of local smoothing methods. Smoothing splines are global methods in the sense that their design criterion involves the entire data signal to be smoothed, but their objective is similar, that is, to maximize smoothness.

We assume an observation model of the form $y(t) = x(t) + v(t)$, where $x(t)$ is a smooth trend to be estimated on the basis of N noisy observations $y_n = y(t_n)$ measured at N time instants t_n, for $n = 0, 1, \dots, N-1$, as shown below.



The times t_n, called the *knots*, are not necessarily equally-spaced, but are in increasing order and are assumed to lie within a slightly larger interval $[t_a, t_b]$, that is,

$$t_a < t_0 < t_1 < t_2 < \cdots < t_{N-1} < t_b$$

A smoothing spline fits a continuous function $x(t)$, taken to be the estimate of the underlying smooth trend, by solving the optimization problem:

$$\mathcal{J} = \sum_{n=0}^{N-1} w_n \big(y_n - x(t_n)\big)^2 + \lambda \int_{t_a}^{t_b} \big[x^{(s)}(t)\big]^2 dt = \min \tag{7.1.1}$$

where $x^{(s)}(t)$ denotes the s-th derivative of $x(t)$, λ is a positive "smoothing parameter," and w_n are given non-negative weights.

The performance index strikes a balance between interpolation and smoothing. The first term attempts to interpolate the data by $x(t)$, while the second attempts to minimize the roughness or maximize the smoothness of $x(t)$. The balance between the two terms is controlled by the parameter λ; larger λ increases smoothing, smaller λ interpolates the data more closely.

Schoenberg [357] has shown that the solution to the problem (7.1.1) is a so-called *natural smoothing spline* of polynomial order $2s-1$, that is, $x(t)$ has $2s-2$ continuous derivatives, it is a polynomial of degree $2s-1$ within each subinterval (t_n, t_{n+1}), for $n = 0, 1, \dots, N-2$, and it is a polynomial of order $s-1$ within the end subintervals $[t_a, t_0)$ and $(t_{N-1}, t_b]$.

For discrete-time sampled data, the problem was originally posed and solved for special cases of s by Thiele, Bohlmann, Whittaker, and Henderson [405–412], and is referred to as *Whittaker-Henderson* smoothing. We will consider it in Sec. 8.1. In this case, the performance index becomes:

$$\mathcal{J} = \sum_{n=0}^{N-1} w_n \big(y_n - x_n\big)^2 + \lambda \sum_{n=s}^{N-1} \big[\nabla^s x_n\big]^2 = \min \tag{7.1.2}$$

In this chapter, we concentrate on the case $s = 2$ for the problem (7.1.1), but allow an arbitrary s for problem (7.1.2). For $s = 2$, the performance index (7.1.1) reads:

$$\mathcal{J} = \sum_{n=0}^{N-1} w_n \big(y_n - x(t_n)\big)^2 + \lambda \int_{t_a}^{t_b} \big[\ddot{x}(t)\big]^2 dt = \min \tag{7.1.3}$$

Eq. (7.1.3) will be minimized under the assumption that the desired $x(t)$ and its first and second derivatives $\dot{x}(t), \ddot{x}(t)$ are continuous over $[t_a, t_b]$.

In the next section we solve the problem from a variational point of view and derive the solution as a natural cubic spline.

7.2 Variational Approach

We begin with a short review of variational calculus [354]. Consider first a Lagrangian $\mathcal{L}(x, \dot{x})$ that depends on a function $x(t)$ and its first derivative $\dot{x}(t)$.†

A prototypical variational problem is to find the function $x(t)$ that maximizes or minimizes the "action" functional:

$$\mathcal{J}(x) = \int_{t_a}^{t_b} \mathcal{L}(x, \dot{x})\, dt = \text{extremum} \tag{7.2.1}$$

The optimum function $x(t)$ is found by solving the Euler-Lagrange equation for (7.2.1):

$$\frac{\partial \mathcal{L}}{\partial x} - \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{x}} = 0 \tag{7.2.2}$$

This can be derived as follows. Consider a small deviation from the optimum solution, $x(t) \to x(t) + \delta x(t)$. Then, the corresponding first-order variation of the functional

†$\mathcal{L}$ can also have an explicit dependence on t, but we suppress it in the notation.

(7.2.1) will be:

$$\begin{aligned}\delta\mathcal{J} &= \mathcal{J}(x+\delta x) - \mathcal{J}(x) = \int_{t_a}^{t_b} \big[\mathcal{L}(x+\delta x, \dot{x}+\delta\dot{x}) - \mathcal{L}(x,\dot{x})\big]\, dt \\ &= \int_{t_a}^{t_b} \left[\frac{\partial \mathcal{L}}{\partial x}\delta x + \frac{\partial \mathcal{L}}{\partial \dot{x}}\delta \dot{x}\right] dt = \int_{t_a}^{t_b} \left[\frac{\partial \mathcal{L}}{\partial x}\delta x - \left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right)' \delta x + \left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\delta x\right)'\right] dt\end{aligned}$$

where we used the differential identity‡

$$\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\delta x\right)' = \frac{\partial \mathcal{L}}{\partial \dot{x}}\delta \dot{x} + \left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right)' \delta x \tag{7.2.3}$$

Integrating the last term in $\delta\mathcal{J}$, we obtain:

$$\delta\mathcal{J} = \int_{t_a}^{t_b} \left[\frac{\partial \mathcal{L}}{\partial x} - \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{x}}\right] \delta x\, dt + \frac{\partial \mathcal{L}}{\partial \dot{x}}\delta x\bigg|_{t_b} - \frac{\partial \mathcal{L}}{\partial \dot{x}}\delta x\bigg|_{t_a} \tag{7.2.4}$$

The boundary terms can be removed by assuming the condition:

$$\frac{\partial \mathcal{L}}{\partial \dot{x}}\delta x\bigg|_{t_b} - \frac{\partial \mathcal{L}}{\partial \dot{x}}\delta x\bigg|_{t_a} = 0 \tag{7.2.5}$$

It follows that

$$\delta\mathcal{J} = \int_{t_a}^{t_b} \left[\frac{\partial \mathcal{L}}{\partial x} - \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{x}}\right] \delta x\, dt \tag{7.2.6}$$

which defines the *functional derivative* of $\mathcal{J}(x)$:

$$\frac{\delta\mathcal{J}}{\delta x} = \frac{\partial \mathcal{L}}{\partial x} - \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{x}} \tag{7.2.7}$$

The Euler-Lagrange equation (7.2.2) is obtained by requiring the vanishing of the functional derivative, or the vanishing of the first-order variation $\delta\mathcal{J}$ around the optimum solution for any choice of δx subject to (7.2.5).

The boundary condition (7.2.5) can be achieved in a number of ways. The typical one is to assume that the variation $\delta x(t)$ vanish at the endpoints, $\delta x(t_a) = \delta x(t_b) = 0$. Alternatively, if no restrictions are to be made on $\delta x(t)$, then one must assume the so-called *natural* boundary conditions [354]:

$$\frac{\partial \mathcal{L}}{\partial \dot{x}}\bigg|_{t_a} = \frac{\partial \mathcal{L}}{\partial \dot{x}}\bigg|_{t_b} = 0 \tag{7.2.8}$$

A mixed case is also possible in which at one end one assumes the vanishing of δx and at the other end, the vanishing of $\partial\mathcal{L}/\partial\dot{x}$.

The above results can be extended to the case when the Lagrangian is also a function of the second derivative $\ddot{x}$, that is, $\mathcal{L}(x, \dot{x}, \ddot{x})$. Using Eq. (7.2.3) and the identity,

$$\frac{\partial \mathcal{L}}{\partial \ddot{x}}\delta \ddot{x} = \left(\frac{\partial \mathcal{L}}{\partial \ddot{x}}\delta \dot{x} - \left(\frac{\partial \mathcal{L}}{\partial \ddot{x}}\right)' \delta x\right)' + \left(\frac{\partial \mathcal{L}}{\partial \ddot{x}}\right)'' \delta x$$

‡primes and dots denote differentiation with respect to t.

the first-order variation of $\mathcal{J}$ becomes

$$\delta\mathcal{J} = \mathcal{J}(x+\delta x) - \mathcal{J}(x) = \int_{t_a}^{t_b} \big[\mathcal{L}(x+\delta x,\, \dot{x}+\delta\dot{x},\, \ddot{x}+\delta\ddot{x}) - \mathcal{L}(x,\dot{x},\ddot{x})\big]\,dt$$
$$= \int_{t_a}^{t_b}\left[\frac{\partial\mathcal{L}}{\partial x}\delta x + \frac{\partial\mathcal{L}}{\partial \dot{x}}\delta\dot{x} + \frac{\partial\mathcal{L}}{\partial \ddot{x}}\delta\ddot{x}\right]dt = \int_{t_a}^{t_b}\left[\frac{\partial\mathcal{L}}{\partial x} - \frac{d}{dt}\frac{\partial\mathcal{L}}{\partial \dot{x}} + \frac{d^2}{dt^2}\frac{\partial\mathcal{L}}{\partial \ddot{x}}\right]\delta x\,dt$$
$$+ \left(\frac{\partial\mathcal{L}}{\partial \dot{x}} - \frac{d}{dt}\frac{\partial\mathcal{L}}{\partial \ddot{x}}\right)\delta x\Big|_{t_a}^{t_b} + \frac{\partial\mathcal{L}}{\partial \ddot{x}}\delta\dot{x}\Big|_{t_a}^{t_b}$$

To eliminate the boundary terms, we must assume that

$$\left(\frac{\partial\mathcal{L}}{\partial \dot{x}} - \frac{d}{dt}\frac{\partial\mathcal{L}}{\partial \ddot{x}}\right)\delta x\Big|_{t_a}^{t_b} + \frac{\partial\mathcal{L}}{\partial \ddot{x}}\delta\dot{x}\Big|_{t_a}^{t_b} = 0 \tag{7.2.9}$$

Then, the first-order variation and functional derivative of $\mathcal{J}$ become:

$$\delta\mathcal{J} = \int_{t_a}^{t_b}\left[\frac{\partial\mathcal{L}}{\partial x} - \frac{d}{dt}\frac{\partial\mathcal{L}}{\partial \dot{x}} + \frac{d^2}{dt^2}\frac{\partial\mathcal{L}}{\partial \ddot{x}}\right]\delta x\,dt\,, \quad \frac{\delta\mathcal{J}}{\delta x} = \frac{\partial\mathcal{L}}{\partial x} - \frac{d}{dt}\frac{\partial\mathcal{L}}{\partial \dot{x}} + \frac{d^2}{dt^2}\frac{\partial\mathcal{L}}{\partial \ddot{x}} \tag{7.2.10}$$

Their vanishing leads to the Euler-Lagrange equation for this case:

$$\frac{\partial\mathcal{L}}{\partial x} - \frac{d}{dt}\frac{\partial\mathcal{L}}{\partial \dot{x}} + \frac{d^2}{dt^2}\frac{\partial\mathcal{L}}{\partial \ddot{x}} = 0 \tag{7.2.11}$$

subject to the condition (7.2.9). In the spline problem, because the endpoints t_a, t_b lie slightly outside the knot range, we do not want to impose any restrictions on the values of δx and $\delta\dot{x}$ there. Therefore, to satisfy (7.2.9), we will assume the four *natural* boundary conditions:

$$\frac{\partial\mathcal{L}}{\partial \dot{x}} - \frac{d}{dt}\frac{\partial\mathcal{L}}{\partial \ddot{x}}\Big|_{t_a} = 0\,, \quad \frac{\partial\mathcal{L}}{\partial \ddot{x}}\Big|_{t_a} = 0\,, \quad \frac{\partial\mathcal{L}}{\partial \dot{x}} - \frac{d}{dt}\frac{\partial\mathcal{L}}{\partial \ddot{x}}\Big|_{t_b} = 0\,, \quad \frac{\partial\mathcal{L}}{\partial \ddot{x}}\Big|_{t_b} = 0 \tag{7.2.12}$$

The spline problem (7.1.3) can be put in a variational form as follows,

$$\mathcal{J} = \sum_{n=0}^{N-1} w_n\big(y_n - x(t_n)\big)^2 + \lambda\int_{t_a}^{t_b}\big[\ddot{x}(t)\big]^2 dt = \int_{t_a}^{t_b}\mathcal{L}\,dt = \min \tag{7.2.13}$$

where the Lagrangian depends only on x and $\ddot{x}$,

$$\mathcal{L} = \sum_{n=0}^{N-1} w_n\big(y_n - x(t)\big)^2\delta(t-t_n) + \lambda\big[\ddot{x}(t)\big]^2 \tag{7.2.14}$$

The Euler-Lagrange equation (7.2.11) then reads:

$$\frac{\partial\mathcal{L}}{\partial x} - \frac{d}{dt}\frac{\partial\mathcal{L}}{\partial \dot{x}} + \frac{d^2}{dt^2}\frac{\partial\mathcal{L}}{\partial \ddot{x}} = -2\sum_{n=0}^{N-1} w_n\big(y_n - x(t)\big)\delta(t-t_n) + 2\lambda\ddddot{x}(t) = 0\,, \quad \text{or,}$$

$$\boxed{\ddddot{x}(t) = \lambda^{-1}\sum_{n=0}^{N-1} w_n\big(y_n - x(t_n)\big)\delta(t-t_n)} \tag{7.2.15}$$

where we replaced $\big(y_n - x(t)\big)\delta(t-t_n)$ by $\big(y_n - x(t_n)\big)\delta(t-t_n)$ in the right-hand side. The natural boundary conditions (7.2.12) become:

$$\dddot{x}(t_a) = 0\,, \quad \ddot{x}(t_a) = 0\,, \quad \dddot{x}(t_b) = 0\,, \quad \ddot{x}(t_b) = 0 \tag{7.2.16}$$

7.3 *Natural Cubic Smoothing Splines*

Eq. (7.2.15) implies that $\ddddot{x}(t) = 0$ for all t except at the knot times t_n. This means that $x(t)$ must be a cubic polynomial in t. Within each knot interval $[t_n, t_{n+1}]$, for $n = 0, 1, \dots, N-2$, and within the end-point intervals $[t_a, t_0]$ and $[t_{N-1}, t_b]$, the function $x(t)$ must be a cubic polynomial, albeit with different coefficients in each interval.

Specifically, the boundary conditions (7.2.16) imply that within $[t_a, t_0]$ and $[t_{N-1}, t_b]$, the third-degree polynomials must actually be polynomials of first-degree. Thus, $x(t)$ will have the form:

$$x(t) = \begin{cases} p_{-1}(t) = a_{-1} + b_{-1}(t-t_a)\,, & t_a \le t \le t_0 \\ p_n(t) = a_n + b_n(t-t_n) + \frac{1}{2}c_n(t-t_n)^2 + \frac{1}{6}d_n(t-t_n)^3\,, & t_n \le t \le t_{n+1} \\ p_{N-1}(t) = a_{N-1} + b_{N-1}(t-t_{N-1})\,, & t_{N-1} \le t \le t_b \end{cases} \tag{7.3.1}$$

where $n = 0, 1, \dots, N-2$ for the interval $[t_n, t_{n+1}]$, and we have referred the time origin to the left end of each subinterval. We note that $a_n = x(t_n) = p_n(t_n)$, $b_n = \dot{p}_n(t_n)$, $c_n = \ddot{p}_n(t_n)$, and $d_n = \dddot{p}_n(t_n)$, for $n = 0, 1, \dots, N-1$. The a_n are the smoothed values.

The polynomial pieces join continuously at the knots. The term "natural" cubic spline refers to the property that $x(t)$ is a linear function of t outside the knot range, and consists of cubic polynomial pieces that are continuous and have continuous first and second derivatives at the knot times. Fig. 7.3.1 illustrates the case of $N = 5$ and the numbering convention that we follow.

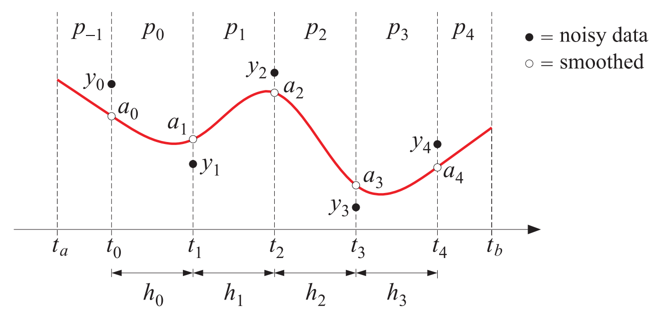


Fig. 7.3.1 Smoothing with natural cubic splines.

Although $x(t), \dot{x}(t), \ddot{x}(t)$ are continuous at the knots, Eq. (7.2.15) implies that the third derivatives $\dddot{x}(t)$ must be discontinuous. Indeed, integrating (7.2.15) around the interval $[t_n - \epsilon, t_n + \epsilon]$ and taking the limit $\epsilon \to 0$, we obtain the N discontinuity conditions:

$$\dddot{x}(t_n)_+ - \dddot{x}(t_n)_- = \lambda^{-1}w_n(y_n - a_n)\,, \quad n = 0, 1, \dots, N-1 \tag{7.3.2}$$

where $\dddot{x}(t_n)_\pm = \lim_{\epsilon\to 0}\dddot{x}(t_n \pm \epsilon)$, and $a_n = x(t_n)$. Expressed in terms of the polynomial pieces, the continuity and discontinuity conditions can be stated as follows:

$$
\begin{aligned}
p_n(t_n) &= p_{n-1}(t_n)\,, \quad n = 0, 1, \dots, N-1 \\
\dot{p}_n(t_n) &= \dot{p}_{n-1}(t_n) \\
\ddot{p}_n(t_n) &= \ddot{p}_{n-1}(t_n) \\
\dddot{p}_n(t_n) - \dddot{p}_{n-1}(t_n) &= \lambda^{-1} w_n (y_n - a_n)
\end{aligned}
\tag{7.3.3}
$$

These provide $4N$ equations. The number of unknown coefficients is also $4N$. Indeed, there are $N-1$ strictly cubic polynomials plus the two linear polynomials at the ends, thus, the total number of coefficients is $4(N-1)+2\cdot 2 = 4N$.

In solving these equations, we follow Reinsch's procedure [358] that eliminates b_n, d_n in favor of a_n, c_n. We begin by applying the continuity conditions (7.3.3) at $t = t_0$,

$$
\begin{aligned}
a_0 &= a_{-1} + b_{-1}(t_0 - t_a) \\
b_0 &= b_{-1} \\
c_0 &= 0 \\
d_0 &= \lambda^{-1} w_0 (y_0 - a_0)
\end{aligned}
\tag{7.3.4}
$$

where in the last two we used $c_{-1} = d_{-1} = 0$. From the first two, it follows that the left-most polynomial can be referred to time origin t_0 and written alternatively as,

$$
p_{-1}(t) = a_{-1} + b_{-1}(t - t_a) = a_0 + b_0(t - t_0) \tag{7.3.5}
$$

For $n = 1, 2, \dots, N-1$, defining $h_{n-1} = t_n - t_{n-1}$, conditions (7.3.3) read:

$$
\begin{aligned}
a_n &= a_{n-1} + b_{n-1}h_{n-1} + \frac{1}{2}c_{n-1}h_{n-1}^2 + \frac{1}{6}d_{n-1}h_{n-1}^3 \\
b_n &= b_{n-1} + c_{n-1}h_{n-1} + \frac{1}{2}d_{n-1}h_{n-1}^2 \\
c_n &= c_{n-1} + d_{n-1}h_{n-1} \\
d_n - d_{n-1} &= \lambda^{-1} w_n (y_n - a_n)
\end{aligned}
\tag{7.3.6}
$$

Since $c_{N-1} = d_{N-1} = 0$, we have at $n = N-1$:

$$
\begin{aligned}
a_{N-1} &= a_{N-2} + b_{N-2}h_{N-2} + \frac{1}{2}c_{N-2}h_{N-2}^2 + \frac{1}{6}d_{N-2}h_{N-2}^3 \\
b_{N-1} &= b_{N-2} + c_{N-2}h_{N-2} + \frac{1}{2}d_{N-2}h_{N-2}^2 \\
0 &= c_{N-2} + d_{N-2}h_{N-2} \\
0 - d_{N-2} &= \lambda^{-1} w_{N-1} (y_{N-1} - a_{N-1})
\end{aligned}
$$

Using the third into the first two equations, we may rewrite them as,

$$
\begin{aligned}
a_{N-1} &= a_{N-2} + b_{N-2}h_{N-2} + \frac{1}{3}c_{N-2}h_{N-2}^2 \\
b_{N-1} &= b_{N-2} + \frac{1}{2}c_{N-2}h_{N-2} \\
c_{N-2} &= -d_{N-2}h_{N-2} \\
d_{N-2} &= -\lambda^{-1} w_{N-1} (y_{N-1} - a_{N-1})
\end{aligned}
\tag{7.3.7}
$$

From the third of Eq. (7.3.6), we have

$$
d_{n-1} = \frac{c_n - c_{n-1}}{h_{n-1}}\,, \quad n = 1, 2, \dots, N-1 \tag{7.3.8}
$$

In particular, we obtain at $n = 1$ and $n = N-1$,

$$
\begin{aligned}
d_0 &= \frac{c_1 - c_0}{h_0} = \frac{c_1}{h_0} = \lambda^{-1} w_0 (y_0 - a_0) \\
-d_{N-2} &= -\frac{c_{N-1} - c_{N-2}}{h_{N-2}} = \frac{c_{N-2}}{h_{N-2}} = \lambda^{-1} w_{N-1} (y_{N-1} - a_{N-1})
\end{aligned}
\tag{7.3.9}
$$

where we used Eqs. (7.3.4) and (7.3.7). Inserting Eq. (7.3.8) into the last of (7.3.6), we obtain for $n = 1, 2, \dots, N-2$:

$$
d_n - d_{n-1} = \frac{c_{n+1} - c_n}{h_n} - \frac{c_n - c_{n-1}}{h_{n-1}} = \lambda^{-1} w_n (y_n - a_n) \tag{7.3.10}
$$

Thus, combining these with (7.3.9), we obtain an $N\times(N-2)$ tridiagonal system of equations that relates the $(N-2)$-dimensional vector $\mathbf{c} = [c_1, c_2, \dots, c_{N-2}]^T$ to the N-dimensional vector $\mathbf{a} = [a_0, a_1, \dots, a_{N-1}]^T$:

$$
\begin{aligned}
&\frac{c_1}{h_0} = \lambda^{-1} w_0 (y_0 - a_0) \\
&\frac{1}{h_{n-1}}c_{n-1} - \left(\frac{1}{h_{n-1}} + \frac{1}{h_n}\right)c_n + \frac{1}{h_n}c_{n+1} = \lambda^{-1} w_n (y_n - a_n)\,, \quad n = 1, 2, \dots, N-2 \\
&\frac{c_{N-2}}{h_{N-2}} = \lambda^{-1} w_{N-1} (y_{N-1} - a_{N-1})
\end{aligned}
\tag{7.3.11}
$$

where we must use $c_0 = c_{N-1} = 0$. These may be written in a matrix form by defining the vector $\mathbf{y} = [y_0, y_1, \dots, y_{N-1}]^T$ and weight matrix $W = \mathrm{diag}\big([w_0, w_1, \dots, w_{N-1}]\big)$,

$$
Q\mathbf{c} = \lambda^{-1} W(\mathbf{y} - \mathbf{a}) \tag{7.3.12}
$$

The $N\times(N-2)$ tridiagonal matrix Q has non-zero matrix elements:

$$
Q_{n-1,n} = \frac{1}{h_{n-1}}\,, \quad Q_{n,n} = -\left(\frac{1}{h_{n-1}} + \frac{1}{h_n}\right), \quad Q_{n+1,n} = \frac{1}{h_n} \tag{7.3.13}
$$

for $n = 1, 2, \dots, N-2$. We note that the matrix elements Q_{ni} were assumed to be indexed such that $0 \le n \le N-1$ and $1 \le i \le N-2$. Next, we determine another relationship between a_n and c_n. Substituting Eq. (7.3.8) into the first and second of (7.3.6), we obtain:

$$\begin{aligned} a_n - a_{n-1} &= b_{n-1}h_{n-1} + \frac{1}{6}(c_n + 2c_{n-1})h_{n-1}^2\,, \quad n = 1,2,\dots,N-1 \\ b_n - b_{n-1} &= \frac{1}{2}(c_n + c_{n-1})h_{n-1} \end{aligned} \tag{7.3.14}$$

The first of these can be solved for b_{n-1} in terms of a_n:

$$\begin{aligned} b_{n-1} &= \frac{a_n - a_{n-1}}{h_{n-1}} - \frac{1}{6}(c_n + 2c_{n-1})h_{n-1}\,, \qquad n = 1,2,\dots,N-1 \\ b_n &= \frac{a_{n+1} - a_n}{h_n} - \frac{1}{6}(c_{n+1} + 2c_n)h_n\,, \quad n = 0,1,\dots,N-2 \end{aligned} \tag{7.3.15}$$

Substituting these into the second of (7.3.14), we obtain for $n = 1, 2, \dots, N-2$:

$$\frac{1}{h_{n-1}}a_{n-1} - \left(\frac{1}{h_{n-1}} + \frac{1}{h_n}\right)a_n + \frac{1}{h_n}a_{n+1} = \frac{1}{6}h_{n-1}c_{n-1} + \frac{1}{3}(h_{n-1} + h_n)c_n + \frac{1}{6}h_n c_{n+1} \tag{7.3.16}$$

This an $(N-2)\times N$ tridiagonal system with the transposed of Q appearing on the left, and the following $(N-2)\times(N-2)$ symmetric tridiagonal matrix on the right,

$$\begin{aligned} T_{n,n} &= \frac{1}{3}(h_{n-1} + h_n)\,, \qquad 1 \le n \le N-2 \\ T_{n+1,n} = T_{n,n+1} &= \frac{1}{6}h_n\,, \qquad 1 \le n \le N-3 \end{aligned} \tag{7.3.17}$$

Thus, the system (7.3.16) can be written compactly as,

$$Q^T\mathbf{a} = T\mathbf{c} \tag{7.3.18}$$

To summarize, the optimal coefficients $\mathbf{a}, \mathbf{c}$ are coupled by

$$\boxed{\begin{aligned} Q^T\mathbf{a} &= T\mathbf{c} \\ Q\mathbf{c} &= \lambda^{-1}W(\mathbf{y} - \mathbf{a}) \end{aligned}} \tag{7.3.19}$$

To clarify the nature of the matrices Q, T, consider the case $N = 6$ with data vector $\mathbf{y} = [y_0, y_1, y_2, y_3, y_4, y_5]^T$. The matrix equations (7.3.19) read explicitly,

$$\begin{bmatrix} h_0^{-1} & -(h_0^{-1} + h_1^{-1}) & h_1^{-1} & 0 & 0 & 0 \\ 0 & h_1^{-1} & -(h_1^{-1} + h_2^{-1}) & h_2^{-1} & 0 & 0 \\ 0 & 0 & h_2^{-1} & -(h_2^{-1} + h_3^{-1}) & h_3^{-1} & 0 \\ 0 & 0 & 0 & h_3^{-1} & -(h_3^{-1} + h_4^{-1}) & h_4^{-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix}$$

$$= \frac{1}{6}\begin{bmatrix} 2(h_0 + h_1) & h_1 & 0 & 0 \\ h_1 & 2(h_1 + h_2) & h_2 & 0 \\ 0 & h_2 & 2(h_2 + h_3) & h_3 \\ 0 & 0 & h_3 & 2(h_3 + h_4) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix}$$

$$\begin{bmatrix} h_0^{-1} & 0 & 0 & 0 \\ -(h_0^{-1} + h_1^{-1}) & h_1^{-1} & 0 & 0 \\ h_1^{-1} & -(h_1^{-1} + h_2^{-1}) & h_2^{-1} & 0 \\ 0 & h_2^{-1} & -(h_2^{-1} + h_3^{-1}) & h_3^{-1} \\ 0 & 0 & h_3^{-1} & -(h_3^{-1} + h_4^{-1}) \\ 0 & 0 & 0 & h_4^{-1} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \lambda^{-1} \begin{bmatrix} w_0(y_0 - a_0) \\ w_1(y_1 - a_1) \\ w_2(y_2 - a_2) \\ w_3(y_3 - a_3) \\ w_4(y_4 - a_4) \\ w_5(y_5 - a_5) \end{bmatrix}$$

In order to have a non-trivial vector $\mathbf{c}$, we will assume that $N \ge 3$. Eqs. (7.3.19) can be solved in a straightforward way. Since T is square and invertible, we may solve the first for $\mathbf{c} = T^{-1}Q^T\mathbf{a}$ and substitute into the second,

$$QT^{-1}Q^T\mathbf{a} = \lambda^{-1}W(\mathbf{y} - \mathbf{a}) \quad \Rightarrow \quad (W + \lambda QT^{-1}Q^T)\mathbf{a} = W\mathbf{y} \quad \text{or,}$$

$$\boxed{\mathbf{a} = (W + \lambda QT^{-1}Q^T)^{-1}W\mathbf{y}} \tag{7.3.20}$$

so that the filtering (the so-called "hat") matrix for the smoothing operation $\mathbf{a} = H\mathbf{y}$ is

$$H = (W + \lambda QT^{-1}Q^T)^{-1}W \tag{7.3.21}$$

Although both Q and T are banded matrices with bandwidth three, the inverse T^{-1} is not banded and neither is $(W + \lambda QT^{-1}Q^T)$. Therefore, the indicated matrix inverse is computationally expensive, requiring $O(N^3)$ operations. However, there is an alternative algorithm due to Reinsch [358] that reduces the computational cost to $O(N)$ operations. From the second of (7.3.19), we have after multiplying it by Q^TW^{-1},

$$\lambda W^{-1}Q\mathbf{c} = \mathbf{y} - \mathbf{a} \quad \Rightarrow \quad \lambda Q^TW^{-1}Q\mathbf{c} = Q^T\mathbf{y} - Q^T\mathbf{a} = Q^T\mathbf{y} - T\mathbf{c}$$

which may be solved for $\mathbf{c}$

$$(T + \lambda Q^TW^{-1}Q)\mathbf{c} = Q^T\mathbf{y} \quad \Rightarrow \quad \mathbf{c} = (T + \lambda Q^TW^{-1}Q)^{-1}Q^T\mathbf{y}$$

where now because W is diagonal, the matrix $R = T + \lambda Q^TW^{-1}Q$ is banded with bandwidth five, and therefore it can be inverted in $O(N)$ operations. This leads to Reinsch's efficient computational algorithm:

$$\boxed{\begin{aligned} R &= T + \lambda Q^TW^{-1}Q \\ \mathbf{c} &= R^{-1}Q^T\mathbf{y} \\ \mathbf{a} &= \mathbf{y} - \lambda W^{-1}Q\mathbf{c} \end{aligned}} \tag{7.3.22}$$

This implies an alternative expression for the matrix H. Eliminating $\mathbf{c}$, we have,

$$\mathbf{a} = \mathbf{y} - \lambda W^{-1}QR^{-1}Q^T\mathbf{y} \quad \Rightarrow \quad \mathbf{a} = (I - \lambda W^{-1}QR^{-1}Q^T)\mathbf{y}\,, \quad \text{or,}$$

$$H = I - \lambda W^{-1}QR^{-1}Q^T = I - \lambda W^{-1}Q(T + \lambda Q^TW^{-1}Q)^{-1}Q^T \tag{7.3.23}$$

The equivalence of Eqs. (7.3.21) and (7.3.23) follows from the matrix inversion lemma. Once the polynomial coefficients $\mathbf{c} = [c_1, c_2, \dots, c_{N-2}]^T$ and $\mathbf{a} = [a_0, a_1, \dots, a_{N-1}]^T$

have been computed, the b_n and d_n coefficients can be obtained from Eqs. (7.3.8) and (7.3.14), and (7.3.7), with $c_0 = c_{N-1} = 0$,

$$\begin{aligned} d_n &= \frac{c_{n+1} - c_n}{h_n}\,, \quad n = 0,1,\dots,N-2\,, \quad \text{and} \quad d_{N-1} = 0 \\ b_n &= \frac{a_{n+1} - a_n}{h_n} - \frac{1}{6}(c_{n+1} + 2c_n)h_n\,, \quad n = 0,1,\dots,N-2 \\ b_{N-1} &= b_{N-2} + \frac{1}{2}c_{N-2}h_{N-2} \end{aligned} \tag{7.3.24}$$

Eqs. (7.3.22) and (7.3.24) provide the complete solution for the coefficients for all the polynomial pieces. We note two particular limits of the solution. For $\lambda = 0$, Eq. (7.3.22) gives $R = T$ and

$$\mathbf{c} = T^{-1}Q^T\mathbf{y}\,, \quad \mathbf{a} = \mathbf{y} \tag{7.3.25}$$

Thus, the smoothing spline interpolates the data, that is, $x(t_n) = a_n = y_n$. Interpolating splines are widely used in image processing and graphics applications.

For $\lambda \to \infty$, the solution corresponds to fitting a straight line to the entire data set. In this case, Eq. (7.3.23) has a well-defined limit,

$$H = I - \lambda W^{-1}Q(T + \lambda Q^T W^{-1}Q)^{-1}Q^T \to I - W^{-1}Q(Q^T W^{-1}Q)^{-1}Q^T \tag{7.3.26}$$

and Eqs. (7.3.22) become:

$$\mathbf{c} = 0\,, \quad \mathbf{a} = \mathbf{y} - W^{-1}Q(Q^T W^{-1}Q)^{-1}Q^T\mathbf{y} \tag{7.3.27}$$

Since $\mathbf{c} = 0$, Eqs. (7.3.24) imply that $d_n = 0$, therefore, the polynomial pieces $p_n(t)$ are first-order polynomials, and we also have $b_n = (a_{n+1} - a_n)/h_n$. The vector $\mathbf{a}$ lies in the null space of Q^T. Indeed, multiplying by Q^T, we have from (7.3.27),

$$Q^T\mathbf{a} = Q^T\mathbf{y} - (Q^T W^{-1}Q)(Q^T W^{-1}Q)^{-1}Q^T\mathbf{y} = Q^T\mathbf{y} - Q^T\mathbf{y} = 0$$

Component-wise this means that the slopes b_n of the $p_n(t)$ polynomials are the same,

$$(Q^T\mathbf{a})_n = \frac{a_{n+1} - a_n}{h_n} - \frac{a_n - a_{n-1}}{h_{n-1}} = b_n - b_{n-1} = 0 \tag{7.3.28}$$

Thus, the polynomials $p_n(t)$ represent pieces of the same straight line. Indeed, setting $b_n \equiv \beta$, and using $a_n = a_{n-1} + \beta h_{n-1}$, we obtain,

$$p_n(t) = a_n + \beta(t - t_n) = a_{n-1} + \beta h_{n-1} + \beta(t - t_n) = a_{n-1} + \beta(t - t_{n-1}) = p_{n-1}(t)$$

This line corresponds to a weighted least-squares straight-line fit through the data y_n, that is, fitting a polynomial $p(t) = \alpha + \beta t$ to

$$\mathcal{J} = \sum_{n=0}^{N-1} w_n\big(y_n - p(t_n)\big)^2 = (\mathbf{y} - \hat{\mathbf{y}})^T W(\mathbf{y} - \hat{\mathbf{y}}) = \min$$

It is easily verified that the coefficients and fitted values $\hat{\mathbf{y}} = [p(t_0), p(t_1), \dots, p(t_n)]^T$ are given by

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = (S^T W S)^{-1}S^T W\mathbf{y}\,, \quad \hat{\mathbf{y}} = S(S^T W S)^{-1}S^T W\mathbf{y} \tag{7.3.29}$$

where S is the $N{\times}2$ polynomial basis matrix defined by

$$S = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ t_0 & t_1 & \cdots & t_{N-1} \end{bmatrix}^T \tag{7.3.30}$$

The fitted values $\hat{\mathbf{y}}$ are exactly equal to those of (7.3.27), as can be verified using the following projection matrix identity, which can be proved using the property $Q^T S = 0$,

$$W^{-1}Q(Q^T W^{-1}Q)^{-1}Q^T + S(S^T W S)^{-1}S^T W = I \tag{7.3.31}$$

7.4 Optimality of Natural Splines

The smoothing spline solution just derived is not only an extremum of the performance index (7.1.3), but also a minimum. To show this, consider a deviation from the optimum solution, $x(t)+f(t)$, where $x(t)$ is the solution (7.3.1) and $f(t)$ an arbitrary twice differentiable function. Then, we must show that $\mathcal{J}(x+f) \ge \mathcal{J}(x)$. Noting that $a_n = x(t_n)$ and denoting $f_n = f(t_n)$, we have,

$$\begin{aligned} \mathcal{J}(x+f) &= \sum_{n=0}^{N-1} w_n(y_n - a_n - f_n)^2 + \lambda\int_{t_a}^{t_b}[\ddot{x}(t) + \ddot{f}(t)]^2 dt \\ \mathcal{J}(x) &= \sum_{n=0}^{N-1} w_n(y_n - a_n)^2 + \lambda\int_{t_a}^{t_b}[\ddot{x}(t)]^2 dt \end{aligned}$$

and by subtracting,

$$\begin{aligned} \mathcal{J}(x+f) - \mathcal{J}(x) &= \sum_{n=0}^{N-1} w_n[f_n^2 - 2(y_n - a_n)f_n] + \lambda\int_{t_a}^{t_b}[\ddot{f}(t)^2 + 2\ddot{x}(t)\ddot{f}(t)]dt \\ &= \sum_{n=0}^{N-1} w_n f_n^2 + \lambda\int_{t_a}^{t_b}\ddot{f}(t)^2\,dt - 2\sum_{n=0}^{N-1} w_n(y_n - a_n)f_n + 2\lambda\int_{t_a}^{t_b}\ddot{x}(t)\ddot{f}(t)\,dt \end{aligned}$$

The first two terms are non-negative. Therefore, the desired result $\mathcal{J}(x+f) \ge \mathcal{J}(x)$ would follow if we can show that the last two terms that are linear in f cancel each other. Indeed, this follows from the optimality conditions (7.3.3). Splitting the integration range as a sum over the subintervals, and replacing $\ddot{x}(t)$ by $\ddot{p}_n(t)$ over the nth subinterval, we have,

$$\begin{aligned} \int_{t_a}^{t_b}\ddot{x}(t)\ddot{f}(t)dt &= \int_{t_a}^{t_0}\ddot{p}_{-1}(t)\ddot{f}(t)dt + \sum_{n=0}^{N-2}\int_{t_n}^{t_{n+1}}\ddot{p}_n(t)\ddot{f}(t)dt + \int_{t_{N-1}}^{t_b}\ddot{p}_{N-1}(t)\ddot{f}(t)dt \\ &= \sum_{n=0}^{N-2}\int_{t_n}^{t_{n+1}}\Big[\big(\ddot{p}_n(t)\dot{f}(t)\big)' - \dddot{p}_n(t)\dot{f}(t)\Big]dt \end{aligned}$$

where we dropped the first an last integrals because $p_{-1}(t)$ and $p_{N-1}(t)$ are linear and have vanishing second derivatives, and used the identity $\ddot{p}_n\ddot{f} = (\ddot{p}_n\dot{f})' - \dddot{p}_n\dot{f}$. The first term is a complete derivative and can be integrated simply. In the second term, we may use $\dddot{p}_n(t)=d_n$ over the nth subinterval to obtain,

$$\int_{t_a}^{t_b}\ddot{x}(t)\ddot{f}(t)\,dt = \ddot{p}_{N-2}(t_{N-1})\dot{f}(t_{N-1})-\ddot{p}_0(t_0)\dot{f}(t_0)-\sum_{n=0}^{N-2}\int_{t_n}^{t_{n+1}}d_n\dot{f}(t)\,dt$$

$$= \ddot{p}_{N-2}(t_{N-1})\dot{f}(t_{N-1})-\ddot{p}_0(t_0)\dot{f}(t_0)-\sum_{n=0}^{N-2}d_n(f_{n+1}-f_n)$$

From the continuity at $t = t_0$ and $t = t_{N-1}$, we have $\ddot{p}_{N-2}(t_{N-1})=\ddot{p}_{N-1}(t_{N-1})=0$ and $\ddot{p}_0(t_0)=\ddot{p}_{-1}(t_0)=0$. Thus, we find,

$$\int_{t_a}^{t_b}\ddot{x}(t)\ddot{f}(t)\,dt = -\sum_{n=0}^{N-2}d_n(f_{n+1}-f_n)=d_0f_0+\sum_{n=1}^{N-2}(d_n-d_{n-1})f_n-d_{N-2}f_{N-1}$$

Using Eqs. (7.3.4), (7.3.6), and (7.3.7), we obtain

$$\int_{t_a}^{t_b}\ddot{x}(t)\ddot{f}(t)\,dt = \lambda^{-1}\sum_{n=0}^{N-1}w_n(y_n-a_n)f_n$$

Thus, these two terms cancel in the difference of the performance indices,

$$\mathcal{J}(x+f)-\mathcal{J}(x)=\sum_{n=0}^{N-1}w_nf_n^2+\lambda\int_{t_a}^{t_b}\ddot{f}(t)^2\,dt \tag{7.4.1}$$

Hence, $\mathcal{J}(x+f)\geq\mathcal{J}(x)$, with equality achieved when $\ddot{f}(t)=0$ and $f_n = f(t_n)=0$, which imply that $f(t)=0$.

Although we showed that the *interpolating* spline case corresponds to the special case $\lambda = 0$, it is worth looking at its optimality properties from a variational point of view. Simply setting $\lambda = 0$ into the performance index (7.1.3) is not useful because it only implies the interpolation property $x(t_n)=y_n$. An alternative point of view is to consider the following constrained variational problem:

$$\mathcal{J}=\int_{t_a}^{t_b}[\ddot{x}(t)]^2dt=\min \qquad \text{subject to } x(t_n)=y_n\,,\quad n=0,1,\dots,N-1 \tag{7.4.2}$$

The constraints can be incorporated using a set of Lagrange multipliers μ_n,

$$\mathcal{J}=\sum_{n=0}^{N-1}2\mu_n(y_n-x(t_n))+\int_{t_a}^{t_b}[\ddot{x}(t)]^2dt=\min \tag{7.4.3}$$

The corresponding effective Lagrangian is,

$$\mathcal{L}=\sum_{n=0}^{N-1}2\mu_n(y_n-x(t))\delta(t-t_n)+[\ddot{x}(t)]^2 \tag{7.4.4}$$

The Euler-Lagrange equation (7.2.11) then gives,

$$\ddddot{x}(t)=\sum_{n=0}^{N-1}\mu_n\delta(t-t_n) \tag{7.4.5}$$

which is to be solved subject to the same natural boundary conditions as (7.2.16),

$$\dddot{x}(t_a)=0\,,\quad \ddot{x}(t_a)=0\,,\quad \dddot{x}(t_b)=0\,,\quad \ddot{x}(t_b)=0 \tag{7.4.6}$$

This is identical to the smoothing spline case with the replacement of $\lambda^{-1}(y_n-a_n)$ by μ_n, or, vectorially $\lambda^{-1}W(\mathbf{y}-\mathbf{a})\to\boldsymbol{\mu}$. Therefore, the solution will be a natural spline with Eq. (7.3.19) replaced by

$$Q^T\mathbf{y}=T\mathbf{c}\,,\quad Q\mathbf{c}=\boldsymbol{\mu}$$

which is the same as the $\lambda = 0$ smoothing spline case. Thus, the interpolating spline solution is defined by $\mathbf{a}=\mathbf{y}$ and $\mathbf{c}=T^{-1}Q^T\mathbf{y}$, with the equation $Q\mathbf{c}=\boldsymbol{\mu}$ fixing the Lagrange multiplier vector.

7.5 Generalized Cross Validation

The cross-validation and generalized cross-validation criteria are popular ways of choosing the smoothing parameter λ. We encountered these criteria in sections 4.5 and 5.2.

The cross-validation criterion selects the λ that minimizes the weighted sum of squared errors [352]:

$$\text{CV}(\lambda)=\frac{1}{N}\sum_{i=0}^{N-1}w_i(y_i-a_i^-)^2=\min \tag{7.5.1}$$

where a_i^- is the estimate of the sample y_i obtained by *deleting* the ith observation y_i and basing the spline smoothing on the remaining observations. As was the case in Sec. 5.2, we may show that

$$y_i-a_i^-=\frac{y_i-a_i}{1-H_{ii}} \tag{7.5.2}$$

where H_{ii} is the ith diagonal element of the filtering matrix H of the smoothing problem with the observation y_i included, and a_i, the corresponding estimate of y_i. Thus, the CV index can be expressed as:

$$\text{CV}(\lambda)=\frac{1}{N}\sum_{i=0}^{N-1}w_i(y_i-a_i^-)^2=\frac{1}{N}\sum_{i=0}^{N-1}w_i\left(\frac{y_i-a_i}{1-H_{ii}}\right)^2=\min \tag{7.5.3}$$

The generalized cross-validation criterion replaces H_{ii} by its average over i, that is,

$$\text{GCV}(\lambda)=\frac{1}{N}\sum_{i=0}^{N-1}w_i\left(\frac{y_i-a_i}{1-\bar{H}}\right)^2=\min\,,\quad \bar{H}=\frac{1}{N}\sum_{i=0}^{N-1}H_{ii}=\frac{1}{N}\text{tr}(H) \tag{7.5.4}$$

The GCV can be evaluated efficiently with $O(N)$ operations for each value of λ using the algorithm of [377]. Noting that $1-\bar{H}=\big(N-\text{tr}(H)\big)/N=\text{tr}(I-H)/N$, and defining $\mathbf{e}=\mathbf{y}-\mathbf{a}=(I-H)\mathbf{y}$, the GCV can be written in a slightly different form,

$$\frac{1}{N}\,\mathrm{GCV}(\lambda)=\frac{\sum_{i=0}^{N-1}w_i(y_i-a_i)^2}{\big[\mathrm{tr}(I-H)\big]^2}=\frac{\mathbf{e}^TW\mathbf{e}}{\big[\mathrm{tr}(I-H)\big]^2}=\min \tag{7.5.5}$$

To show Eq. (7.5.2), consider the index (7.1.3) with the i-th observation y_i deleted:

$$\mathcal{J}_-=\sum_{\substack{n=0\\n\neq i}}^{N-1}w_n\big(y_n-x(t_n)\big)^2+\lambda\int_{t_a}^{t_b}\big[\ddot{x}(t)\big]^2dt=\min \tag{7.5.6}$$

The i-th term can be included provided we attach zero weight to it, that is, we may define $w_n^-=w_n$, if $n\neq i$, and $w_i^-=0$:

$$\mathcal{J}_-=\sum_{n=0}^{N-1}w_n^-\big(y_n-x(t_n)\big)^2+\lambda\int_{t_a}^{t_b}\big[\ddot{x}(t)\big]^2dt=\min \tag{7.5.7}$$

It follows from Eq. (7.3.20) that the optimum solutions with and without the i observation are given by

$$\begin{aligned}\mathbf{a}&=H\mathbf{y}=F^{-1}W\mathbf{y}\,, &\quad F&=W+\lambda QT^{-1}Q^T\\ \mathbf{a}^-&=H_-\mathbf{y}=F_-^{-1}W_-\mathbf{y}\,, &\quad F_-&=W_-+\lambda QT^{-1}Q^T\end{aligned} \tag{7.5.8}$$

where W_- is the diagonal matrix of the w_n^-. Defining the i-th unit vector that has one in its i-th slot, $\mathbf{u}_i=[0,\dots,0,1,0,\dots,0]^T$, then W_- is related to the original W by

$$W_-=W-w_i\,\mathbf{u}_i\mathbf{u}_i^T\quad\Rightarrow\quad F_-=F-w_i\,\mathbf{u}_i\mathbf{u}_i^T$$

It follows from Eq. (7.5.8) that,

$$F_-\mathbf{a}^-=W_-\mathbf{y}\quad\Rightarrow\quad(F-w_i\,\mathbf{u}_i\mathbf{u}_i^T)\mathbf{a}^-=(W-w_i\,\mathbf{u}_i\mathbf{u}_i^T)\mathbf{y}$$

Noting that $y_i=\mathbf{u}_i^T\mathbf{y}$ and $a_i^-=\mathbf{u}_i^T\mathbf{a}^-$, we have after multiplying by F^{-1},

$$\mathbf{a}^--w_iF^{-1}\mathbf{u}_i\,a_i^-=\mathbf{a}-w_iF^{-1}\mathbf{u}_iy_i\quad\Rightarrow\quad\mathbf{a}-\mathbf{a}^-=w_iF^{-1}\mathbf{u}_i(y_i-a_i^-)$$

Multiplying by $\mathbf{u}_i^T$ and noting that $H_{ii}=\mathbf{u}_i^TH\mathbf{u}_i=\mathbf{u}_i^TF^{-1}W\mathbf{u}_i=(\mathbf{u}_i^TF^{-1}\mathbf{u}_i)w_i$, we find,

$$a_i-a_i^-=H_{ii}(y_i-a_i^-) \tag{7.5.9}$$

which is equivalent to (7.5.2). An intuitive interpretation [352] of $\mathbf{a}^-$ is that it is obtainable by the original filtering matrix H acting on a modified observation vector $\mathbf{y}^*$ whose i-th entry has been replaced by the estimated value $y_i^*=a_i^-$, and whose other entries agree with those of $\mathbf{y}$. To show it, we note that $W\mathbf{y}^*=W_-\mathbf{y}+w_i\mathbf{u}_iy_i^*$. Then, we have

$$F_-\mathbf{a}^-=W_-\mathbf{y}\quad\Rightarrow\quad(F-w_i\,\mathbf{u}_i\mathbf{u}_i^T)\mathbf{a}^-=W\mathbf{y}^*-w_i\mathbf{u}_iy_i^*\quad\Rightarrow\quad F\mathbf{a}^-=W\mathbf{y}^*-w_i\mathbf{u}_i(y_i^*-a_i^-)$$

Thus, if we choose $y_i^*=a_i^-$, we have $F\mathbf{a}^-=W\mathbf{y}^*$, which gives $\mathbf{a}^-=F^{-1}W\mathbf{y}^*=H\mathbf{y}^*$. A similar result was obtained in Sec. 4.5.

7.6 Repeated Observations

We discussed how to handle repeated observations in local polynomial modeling in Sec. 5.5, replacing the repeated observations by their averages and using their multiplicities to modify the weighting function. A similar procedure can be derived for the spline smoothing case.

Assuming that at each knot time t_n there are m_n observations, y_{ni} with weights w_{ni}, $i=1,2,\dots,m_n$, the performance index (7.1.3) may be modified as follows:

$$\mathcal{J}=\sum_{n=0}^{N-1}\sum_{i=1}^{m_n}w_{ni}\big(y_{ni}-x(t_n)\big)^2+\lambda\int_{t_a}^{t_b}\big[\ddot{x}(t)\big]^2dt=\min \tag{7.6.1}$$

Let us define the weighted-averaged observations and corresponding weights by:

$$\boxed{\bar{y}_n=\frac{1}{\bar{w}_n}\sum_{i=1}^{m_n}w_{ni}y_{ni}\,,\quad \bar{w}_n=\sum_{i=1}^{m_n}w_{ni}} \tag{7.6.2}$$

If the weights w_{ni} are unity, $\bar{y}_n$ and $\bar{w}_n$ reduce to ordinary averages and multiplicities. It is easily verified that $\mathcal{J}$ can be written in the alternative form:

$$\mathcal{J}=\sum_{n=0}^{N-1}\bar{w}_n\big(\bar{y}_n-x(t_n)\big)^2+\lambda\int_{t_a}^{t_b}\big[\ddot{x}(t)\big]^2dt+\text{const.}=\min \tag{7.6.3}$$

up to a constant that does not depend on the unknown function $x(t)$ to be determined. Thus, the case of multiple observations may be reduced to an ordinary spline smoothing problem.

7.7 Equivalent Filter

The filtering equation of a smoothing spline, $\mathbf{a}=H\mathbf{y}$, raises the question of whether it is possible to view it as an ordinary convolutional filtering operation. Such a viewpoint indeed arises if we replace the performance index (7.1.3) with the following one, which assumes the availability of continuous-time observations $y(t)$ for $-\infty<t<\infty$,

$$\mathcal{J}=\int_{-\infty}^{\infty}\big|y(t)-x(t)\big|^2dt+\lambda\int_{-\infty}^{\infty}\big|\ddot{x}(t)\big|^2dt=\min \tag{7.7.1}$$

The solution can be carried out easily in the frequency domain. Using Parseval's identity and denoting the Fourier transforms of $y(t),x(t)$ by $Y(\omega),X(\omega)$, and noting that the transform of $\ddot{x}(t)$ is $-\omega^2X(\omega)$, we obtain the equivalent criterion,

$$\mathcal{J}=\int_{-\infty}^{\infty}\big|Y(\omega)-X(\omega)\big|^2\frac{d\omega}{2\pi}+\lambda\int_{-\infty}^{\infty}\omega^4\big|X(\omega)\big|^2\frac{d\omega}{2\pi}=\min \tag{7.7.2}$$

Setting the functional derivative of $\mathcal{J}$ with respect to $X^*(\omega)$ to zero,† we obtain the Euler-Lagrange equation in this case:‡

$$\frac{\delta\mathcal{J}}{\delta X^*(\omega)}=-\big[Y(\omega)-X(\omega)\big]+\lambda\omega^4X(\omega)=0 \tag{7.7.3}$$

† $X(\omega)$ and its complex conjugate $X^*(\omega)$ are treated as independent variables in Eqs. (7.7.2) and (7.7.3).

‡ The boundary conditions for this variational problem are that $X(\omega)\to 0$ for $\omega\to\pm\infty$.

which leads to the transfer function $H(\omega)=X(\omega)/Y(\omega)$ between the input $Y(\omega)$ and the output $X(\omega)$:

$$H(\omega)=\frac{1}{1+\lambda\omega^4}\qquad \text{(equivalent smoothing filter)} \tag{7.7.4}$$

Its impulse response (i.e., the inverse Fourier transform) is

$$h(t)=\frac{a}{2}\big(\sin a|t|+\cos at\big)e^{-a|t|}\,,\quad -\infty<t<\infty \tag{7.7.5}$$

where $a=(4\lambda)^{-1/4}$. The impulse response $h(t)$ is double-sided, and therefore, it cannot be used in real-time applications. However, it is evident that the filter is a lowpass filter with a (6-dB) cutoff frequency of $\omega_0=\lambda^{-1/4}$. Fig. 7.7.1 depicts $h(t)$ and $H(\omega)$ for three values of the smoothing parameter, $\lambda=1$, $\lambda=1/5$, and $\lambda=5$.

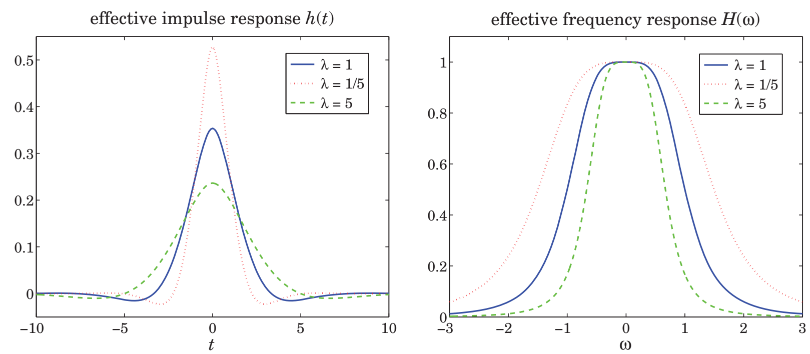


Fig. 7.7.1 Effective impulse and frequency responses in spline smoothing.

One can also work in the time-domain with similar results. The Euler-Lagrange equation (7.2.11) leads to,

$$x(t)-y(t)+\lambda\ddddot{x}(t)=0\quad\Rightarrow\quad x(t)+\lambda\ddddot{x}(t)=y(t) \tag{7.7.6}$$

Fourier transforming both sides we obtain $(1+\lambda\omega^4)X(\omega)=Y(\omega)$, which leads to Eq. (7.7.4) by solving for $H(\omega)=X(\omega)/Y(\omega)$.

A similar approach will be used in the Whittaker-Henderson discrete-time case discussed in Sec. 8.1 The resulting filter is often referred to in the business and finance literature as the Hodrick-Prescott filter.

Variants of the Whittaker-Henderson approach were first introduced in 1880 by Thiele [405,406] and in 1899 by Bohlmann [407]. Bohlmann considered and solved both the discrete- and continuous-time versions of the performance index,

$$\begin{aligned}\mathcal{J}&=\sum_{n=0}^{N-1}(y_n-x_n)^2+\lambda\sum_{n=1}^{N-1}(x_n-x_{n-1})^2=\min\\ \mathcal{J}&=\int_{t_a}^{t_b}|y(t)-x(t)|^2dt+\lambda\int_{t_a}^{t_b}|\dot{x}(t)|^2dt=\min\end{aligned} \tag{7.7.7}$$

In the continuous-time case, the Euler-Lagrange equation, transfer function, and impulse response of the resulting smoothing filter are:

$$x(t)-\lambda\ddot{x}(t)=y(t)\quad\Rightarrow\quad H(\omega)=\frac{1}{1+\lambda\omega^2}\,,\quad h(t)=\frac{1}{2\sqrt{\lambda}}e^{-|t|/\sqrt{\lambda}} \tag{7.7.8}$$

Thiele considered the unequally-spaced knot case and the weighted performance index:

$$\mathcal{J}=\sum_{n=0}^{N-1}\frac{1}{\sigma_n^2}\big[y_n-x(t_n)\big]^2+\sum_{n=1}^{N-1}\frac{1}{w_n^2}\big[x(t_n)-x(t_{n-1})\big]^2=\min \tag{7.7.9}$$

It is remarkable that Thiele formulated this problem as a state-space model—to use modern parlance—and solved it recursively using essentially the Kalman filter and associated smoother. Moreover, he showed how to estimate the unknown model parameters using the EM algorithm. We will be discussing these ideas later on.

7.8 Stochastic Model

Like the exponential smoothing case, spline smoothing can be given a stochastic state-space model interpretation [397–404]. The spline function solution $x(t)$ of Eq. (7.3.1) can be regarded as an optimum linear estimate of the underlying stochastic process based on the N observations $\{y_0,y_1,\dots,y_{N-1}\}$ subject to some additional assumptions on the initial conditions [399].

The state-space model allows the use of Kalman filtering techniques resulting in efficient computational algorithms, which like the Reinsch algorithm are also $O(N)$. But in addition, the state-space model allows the estimation of the smoothing parameter.

The basis of such a stochastic model (for the cubic spline case) is the stochastic differential equation:

$$\ddot{x}(t)=w(t) \tag{7.8.1}$$

where $w(t)$ is a zero-mean white-noise process of variance σ_w^2, that is, its autocorrelation function is $E\big[w(t)w(\tau)\big]=\sigma_w^2\delta(t-\tau)$.

In the observation model $y(t)=x(t)+v(t)$, we may assume that $v(t)$ is uncorrelated with $w(t)$ and is white noise with variance σ_v^2. It turns out that the smoothing parameter can be identified as the ratio $\lambda=\sigma_v^2/\sigma_w^2$. The N actual observed values are $y_n=x(t_n)+v(t_n)$. Integrating Eq. (7.8.1) over the interval $[t_n,t]$, we obtain,

$$\begin{aligned}\dot{x}(t)&=\dot{x}(t_n)+\int_{t_n}^{t}w(\tau)\,d\tau\\ x(t)&=x(t_n)+(t-t_n)\dot{x}(t_n)+\int_{t_n}^{t}(t-\tau)w(\tau)\,d\tau\end{aligned} \tag{7.8.2}$$

The process $\dot{x}(t)$ is integrated white noise, or a Wiener or Brownian process. The process $x(t)$ is an integrated Wiener process. We may write these in vector form by defining the state and noise vectors,

$$\mathbf{x}_t=\begin{bmatrix}x(t)\\ \dot{x}(t)\end{bmatrix},\quad \mathbf{x}_n=\begin{bmatrix}x(t_n)\\ \dot{x}(t_n)\end{bmatrix},\quad \mathbf{w}_t=\int_{t_n}^{t}\begin{bmatrix}t-\tau\\ 1\end{bmatrix}w(\tau)\,d\tau \tag{7.8.3}$$

and the state transition matrix,

$$A(t,t_n) = \begin{bmatrix} 1 & t-t_n \\ 0 & 1 \end{bmatrix} \tag{7.8.4}$$

Then, Eq. (7.8.2) can be written compactly as

$$\mathbf{x}_t = A(t,t_n)\mathbf{x}_n + \mathbf{w}_t \tag{7.8.5}$$

The covariance matrix of the noise component $\mathbf{w}_t$ is:

$$\begin{aligned} E[\mathbf{w}_t\mathbf{w}_t^T] &= \int_{t_n}^{t}\int_{t_n}^{t}\begin{bmatrix} t-\tau \\ 1\end{bmatrix}[t-\tau',\,1]\,E[w(\tau)w(\tau')]\,d\tau d\tau' \\ &= \int_{t_n}^{t}\int_{t_n}^{t}\begin{bmatrix} t-\tau \\ 1\end{bmatrix}[t-\tau',\,1]\,\sigma_w^2\,\delta(\tau-\tau')\,d\tau d\tau' \\ &= \sigma_w^2\begin{bmatrix} \frac{1}{3}(t-t_n)^3 & \frac{1}{2}(t-t_n)^2 \\ \frac{1}{2}(t-t_n)^2 & (t-t_n)\end{bmatrix} \end{aligned} \tag{7.8.6}$$

At $t = t_{n+1}$, we obtain the state equation,

$$\mathbf{x}_{n+1} = A(t_{n+1},t_n)\mathbf{x}_n + \mathbf{w}_{n+1}\,, \quad \mathbf{w}_{n+1} = \int_{t_n}^{t_{n+1}}\begin{bmatrix} t_{n+1}-\tau \\ 1\end{bmatrix} w(\tau)\,d\tau \tag{7.8.7}$$

where, using $h_n = t_{n+1} - t_n$,

$$A(t_{n+1},t_n) = \begin{bmatrix} 1 & h_n \\ 0 & 1\end{bmatrix}, \quad E[\mathbf{w}_{n+1}\mathbf{w}_{n+1}^T] = \sigma_w^2\begin{bmatrix} \frac{1}{3}h_n^3 & \frac{1}{2}h_n^2 \\ \frac{1}{2}h_n^2 & h_n\end{bmatrix} \tag{7.8.8}$$

In terms of the spline coefficients, we have $a_n = x(t_n)$ and $b_n = \dot{x}(t_n)$ at $t = t_n$, and similarly at $t = t_{n+1}$. Following [28], we would like to show the following estimation result. Given the state-vectors $\mathbf{x}_n, \mathbf{x}_{n+1}$ at the two end points of the interval $[t_n, t_{n+1}]$, the spline function $x(t)$ of (7.3.1), and its derivative $\dot{x}(t)$, can be regarded as the mean-square estimates of the state-vector $\mathbf{x}_t$ based on $\mathbf{x}_n, \mathbf{x}_{n+1}$, that is, assuming gaussian noises, given by the conditional mean,

$$\hat{\mathbf{x}}_t = E[\mathbf{x}_t|\mathbf{x}_n, \mathbf{x}_{n+1}] \tag{7.8.9}$$

If we orthogonalize $\mathbf{x}_{n+1}$ with respect to $\mathbf{x}_n$, that is, replacing it by the innovations vector $\boldsymbol{\varepsilon}_{n+1} = \mathbf{x}_{n+1} - E[\mathbf{x}_{n+1}|\mathbf{x}_n]$, then we may use the regression lemma from Chap. 1 to write (7.8.9) in the form:

$$\hat{\mathbf{x}}_t = E[\mathbf{x}_t|\mathbf{x}_n, \mathbf{x}_{n+1}] = E[\mathbf{x}_t|\mathbf{x}_n] + \Sigma_{x_t\varepsilon_{n+1}}\Sigma_{\varepsilon_{n+1}\varepsilon_{n+1}}^{-1}\boldsymbol{\varepsilon}_{n+1} \tag{7.8.10}$$

We have from Eq. (7.8.5) and (7.8.7),

$$E[\mathbf{x}_t|\mathbf{x}_n] = A(t,t_n)\mathbf{x}_n\,, \quad E[\mathbf{x}_{n+1}|\mathbf{x}_n] = A(t_{n+1},t_n)\mathbf{x}_n \tag{7.8.11}$$

the latter implying,

$$\boldsymbol{\varepsilon}_{n+1} = \mathbf{x}_{n+1} - E[\mathbf{x}_{n+1}|\mathbf{x}_n] = \mathbf{x}_{n+1} - A(t_{n+1},t_n)\mathbf{x}_n = \mathbf{w}_{n+1} \tag{7.8.12}$$

and therefore, we have for the covariance matrices:

$$\Sigma_{x_t\varepsilon_{n+1}} = E[\mathbf{x}_t\boldsymbol{\varepsilon}_{n+1}^T] = E[\mathbf{x}_t\mathbf{w}_{n+1}^T] = E[\mathbf{w}_t\mathbf{w}_{n+1}^T]\,, \quad \Sigma_{\varepsilon_{n+1}\varepsilon_{n+1}} = E[\mathbf{w}_{n+1}\mathbf{w}_{n+1}^T]$$

The latter has already been calculated in (7.8.7). For the former, we split the integration range of $\mathbf{w}_{n+1}$ as follows,

$$\mathbf{w}_{n+1} = \int_{t_n}^{t_{n+1}}\begin{bmatrix} t_{n+1}-\tau \\ 1\end{bmatrix} w(\tau)\,d\tau = \left(\int_{t_n}^{t} + \int_{t}^{t_{n+1}}\right)\begin{bmatrix} t_{n+1}-\tau \\ 1\end{bmatrix} w(\tau)\,d\tau$$

and note that only the first term is correlated with $\mathbf{w}_t$, thus, resulting in

$$\begin{aligned} \Sigma_{x_t\varepsilon_{n+1}} = E[\mathbf{w}_t\mathbf{w}_{n+1}^T] &= \int_{t_n}^{t}\int_{t_n}^{t}\begin{bmatrix} t-\tau \\ 1\end{bmatrix}[t_{n+1}-\tau',\,1]\,E[w(\tau)w(\tau')]\,d\tau d\tau' \\ &= \sigma_w^2\int_{t_n}^{t}\begin{bmatrix} t-\tau \\ 1\end{bmatrix}[t_{n+1}-\tau,1]\,d\tau \\ &= \begin{bmatrix} \frac{1}{6}(t-t_n)^2(t_n+2h_n-t) & \frac{1}{2}(t-t_n)^2 \\ \frac{1}{2}(t-t_n)(t_n+2h_n-t) & (t-t_n)\end{bmatrix} \end{aligned} \tag{7.8.13}$$

We may now calculate the estimation matrix $H_{n+1} = \Sigma_{x_t\varepsilon_{n+1}}\Sigma_{\varepsilon_{n+1}\varepsilon_{n+1}}^{-1}$,

$$H_{n+1} = \begin{bmatrix} \frac{1}{h_n^3}(t-t_n)^2(2t_n+3h_n-2t) & \frac{1}{h_n^2}(t-t_n)^2(t-t_n-h_n) \\ \frac{6}{h_n^3}(t-t_n)(t_n+h_n-t) & \frac{1}{h_n^2}(t-t_n)(3t-3t_n-2h_n)\end{bmatrix} \tag{7.8.14}$$

It follows that the estimate $\hat{\mathbf{x}}_t$ is

$$\hat{\mathbf{x}}_t = E[\mathbf{x}_t|\mathbf{x}_n] + H_{n+1}\boldsymbol{\varepsilon}_{n+1} = A(t,t_n)\mathbf{x}_n + H_{n+1}\big(\mathbf{x}_{n+1} - A(t_{n+1},t_n)\mathbf{x}_n\big) \tag{7.8.15}$$

Setting

$$\hat{\mathbf{x}}_t = \begin{bmatrix} \hat{x}(t) \\ \hat{\dot{x}}(t)\end{bmatrix}, \quad \mathbf{x}_n = \begin{bmatrix} a_n \\ b_n\end{bmatrix}, \quad \mathbf{x}_{n+1} = \begin{bmatrix} a_{n+1} \\ b_{n+1}\end{bmatrix}$$

we obtain

$$\begin{aligned} \hat{x}(t) &= a_n + b_n(t-t_n) + \frac{1}{h_n^3}(t-t_n)^2(2t_n+3h_n-2t)(a_{n+1}-a_n-b_nh_n) \\ &\quad + \frac{1}{h_n^2}(t-t_n)^2(t-t_n-h_n)(b_{n+1}-b_n) \\ \hat{\dot{x}}(t) &= b_n + \frac{6}{h_n^3}(t-t_n)(t_n+h_n-t)(a_{n+1}-a_n-b_nh_n) \\ &\quad + \frac{1}{h_n^2}(t-t_n)(3t-3t_n-2h_n)(b_{n+1}-b_n) \end{aligned}$$

Using the continuity relationships (7.3.6),

$$a_{n+1} = a_n + b_n h_n + \frac{1}{2}c_n h_n^2 + \frac{1}{6}d_n h_n^3$$
$$b_{n+1} = b_n + c_n h_n + \frac{1}{2}d_n h_n^2$$

it follows that the expressions for $\hat{x}(t)$ and $\hat{\dot{x}}(t)$ reduce to those of Eq. (7.3.1),

$$\hat{x}(t) = a_n + b_n(t-t_n) + \frac{1}{2}c_n(t-t_n)^2 + \frac{1}{6}d_n(t-t_n)^3$$
$$\hat{\dot{x}}(t) = b_n + c_n(t-t_n) + \frac{1}{2}d_n(t-t_n)^2$$

the second being of course the derivative of the first. The asymptotic filter (7.7.4) may also be given a stochastic interpretation in the sense that it can be regarded as the optimum double-sided (i.e., unrealizable) Wiener filter of estimating $x(t)$ from $y(t)$ of the signal model,

$$y(t) = x(t) + v(t)\,, \quad \ddot{x}(t) = w(t) \tag{7.8.16}$$

We will see in Chap. 11 that for stationary signals $x(t), y(t)$, with power spectral densities $S_{xy}(\omega)$ and $S_{yy}(\omega)$, the optimum double-sided Wiener filter has frequency response:

$$H(\omega) = \frac{S_{xy}(\omega)}{S_{yy}(\omega)} \tag{7.8.17}$$

Because $x(t)$ is an integrated Wiener process, it is not stationary, and therefore, $S_{xy}(\omega)$ and $S_{yy}(\omega)$ do not exist. However, it has been shown [643–649] that for certain types of nonstationary signals, which have the property that they become stationary under a suitable filtering transformation, Eq. (7.8.17) remains valid in the following modified form:

$$H(\omega) = \frac{S_{\bar{x}\bar{y}}(\omega)}{S_{\bar{y}\bar{y}}(\omega)} \tag{7.8.18}$$

where $\bar{x}(t), \bar{y}(t)$ are the stationary filtered versions of $x(t), y(t)$. For the model of Eq. (7.8.16), the necessary filtering operation is double differentiation, $\bar{x}(t) = \ddot{x}(t) = w(t)$, which can be expressed in the frequency domain as $\bar{X}(\omega) = D(\omega)X(\omega)$, with $D(\omega) = (j\omega)^2 = -\omega^2$. For the observation signal, we have similarly $\bar{y}(t) = \ddot{y} = \ddot{x} + \ddot{v}$. Since $w(t), v(t)$ are uncorrelated, we find

$$S_{\bar{x}\bar{y}}(\omega) = S_{ww}(\omega) = \sigma_w^2$$
$$S_{\bar{y}\bar{y}}(\omega) = S_{ww}(\omega) + S_{vv}(\omega)|D(\omega)|^2 = \sigma_w^2 + \sigma_v^2\,\omega^4$$

which leads to

$$H(\omega) = \frac{S_{\bar{x}\bar{y}}(\omega)}{S_{\bar{y}\bar{y}}(\omega)} = \frac{\sigma_w^2}{\sigma_w^2 + \sigma_v^2\,\omega^4} = \frac{1}{1+\lambda\omega^4}\,, \quad \lambda = \frac{\sigma_v^2}{\sigma_w^2} \tag{7.8.19}$$

This can be written in the form,

$$H(\omega) = \frac{\sigma_w^2/\omega^4}{\sigma_w^2/\omega^4 + \sigma_v^2} \tag{7.8.20}$$

which is what we would get from Eq. (7.8.17) had we pretended that the spectral densities did exist. Indeed it would follow in such a case from Eq. (7.8.16) that $S_{xy}(\omega) = \sigma_w^2/\omega^4$ and $S_{yy}(\omega) = \sigma_w^2/\omega^4 + \sigma_v^2$.

7.9 Computational Aspects

Eqs. (7.3.22) and (7.3.24) describing the complete spline solution have been implemented by the MATLAB function `splsm`,

```
P = splsm(t,y,lambda,w);     % spline smoothing
```

where `t,y` are the knot times $[t_0, t_1, \dots, t_{N-1}]$ and data $[y_0, y_1, \dots, y_{N-1}]$ (entered as row or column vectors), `lambda` is the smoothing parameter λ, and `w` the vector of weights $[w_0, w_1, \dots, w_{N-1}]$, which default to unity values. The output `P` is an $N\times 4$ matrix whose n-th row are the polynomial coefficients $[a_n, b_n, c_n, d_n]$. Thus, the vector **a** is the first column of `P`. Internally, the matrices T, Q are computed as sparse banded matrices with the help of the function `splmat`,

```
[T,Q] = splmat(h);     % spline sparse matrices T,Q
```

where `h` is the vector of knot spacings $[h_0, h_1, \dots, h_{N-1}]$, which is simply computed by the `diff` operation on the knot times `t`, that is, `h=diff(t)`. The smoothing spline may be evaluated at any value of t in the range $t_a \le t \le t_b$ using Eq. (7.3.1). The function `splval` performs the evaluation of $x(t)$ at any vector of t's,

```
ys = splval(P,t,ts);     % spline evaluation at a vector of grid points ts
```

where `ys` is the vector of values $x(t_s)$, and `P,t` are the spline coefficients and knot times. The GCV criterion (7.5.5) (with the $1/N$ factor removed) may be calculated for any vector of λ values by the function `splgcv`:

```
gcv = splgcv(t,y,lambda,w);     % GCV evaluation at a vector of λ's
```

The optimum λ may be selected by finding the minimum of the GCV over the computed range. Alternatively, the optimum λ may be computed by the related function `splambda`, which performs a golden-mean search over a given interval of λ's,

```
[lopt,gcvopt] = splambda(t,y,la,lb,Nit,w);     % determine optimum λ
```

The starting interval is $[\lambda_a, \lambda_b]$ and `Nit` denotes the number of golden-mean iterations (typically, 10–20). The function `splsm2` is a "robustified" version of `splsm`,

```
[P,ta] = splsm2(t,y,la,w,Nit);     % robust spline smoothing
```

The function starts with the original triplet `[t,y,w]` and uses the LOESS method of repeatedly modifying the weights (with a total of `Nit` iterations), with the outliers being given smaller weights. Because of the modification and zeroing of some of the weights, the output matrix `P` will have dimension $N_a\times 4$ with $N_a \le N$. The function also outputs the corresponding knot times ta (also N_a-dimensional) that survive the down-weighting process.

All of the above functions assume that the observations y_n are unique at the knot times t_n. If there are repeated observations, then the weighted observations and their weights given by Eq. (7.6.2) must be the inputs to the above functions. They may be determined with the function splav, which is similar in spirit to the function avobs, except that it computes weighted averages instead of plain averages:

```
[ta,ya,wa] = splav(t,y,w);          % weighted averages of repeated observations
```

where the outputs [ta,ya,wa] are the resulting unique knot times, observations, and weights.

Example 7.9.1: *Motorcycle data.* The usage of these functions is illustrated by the motorcycle data that we considered earlier in local polynomial modeling. The upper-left graph of Fig. 7.9.1 shows a plot of the GCV calculated with the function splgcv. The optimum value was found to be $\lambda_{opt} = 15.25$ by the function splambda and placed on the graph.

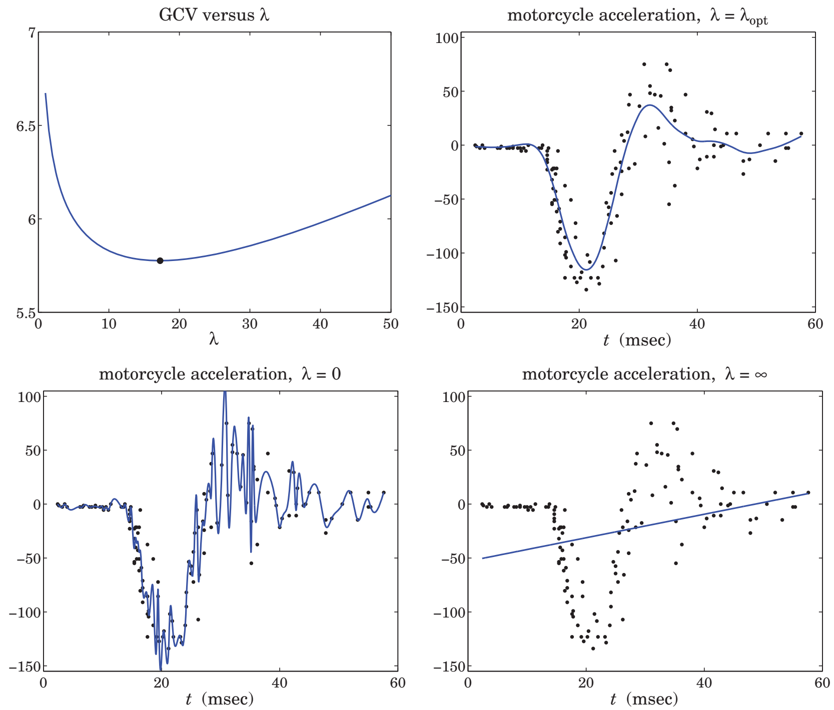


Fig. 7.9.1 Spline smoothing of motorcycle data.

The MATLAB code used for this graph was:


```
Y = loadfile('mcyc.dat');                      % load data

tobs = Y(:,1);  yobs = Y(:,2);                 % extract knot times and observations

[t,y,w] = splav(tobs,yobs);                    % average repeated observations

la1=1; la2=50; Nit=30;                         % search interval and no. of iterations
[lopt,gcvopt] = splambda(t,y,la1,la2,Nit,w);   % determine optimum λ

la = linspace(la1,la2,100);                    % evaluate GCV over λ1 ≤ λ ≤ λ2
gcv = splgcv(t,y,la,w);

figure; plot(la,gcv, lopt,gcvopt, '.');        % plot GCV versus λ
```

The upper-right graph shows the smoothing spline corresponding to $\lambda = \lambda_{opt}$, and evaluated at a uniform grid of time points. The lower two graphs depict the special cases of $\lambda = 0$ corresponding to spline interpolation, and $\lambda = \infty$ corresponding to a linear fit. The following MATLAB code generates these graphs:

```
P = splsm(t,y,lopt,w);                         % spline coefficients

ts = locgrid(t,1001);                          % grid time points
ys = splval(P,t,ts);                           % evaluate spline at grid

figure; plot(tobs,yobs,'.', ts,ys,'-');        % upper-right graph

la = 0; P = splsm(t,y,la,w);                   % spline coefficients for λ = 0
ys = splval(P,t,ts);                           % evaluate spline at grid

figure; plot(t,y,'.', ts,ys,'-');              % lower-left graph

la = inf; P = splsm(t,y,la,w);                 % spline coefficients for λ = ∞
ys = splval(P,t,ts);

figure; plot(t,y,'.', ts,ys,'-');              % lower-right graph
```

Because the motorcycle data have repeated observations, the actual observations were replaced by their averaged values prior to evaluating the splines. The solid-line curves represent the evaluated splines, and the dots are the original data points, except for the lower-left graph in which the dots represent the averaged observations with the spline curve interpolating through them instead of the original points. □

.

Example 7.9.2: *Robust spline smoothing.* Fig. 7.9.2 shows an example of robust spline smoothing. It is the same example considered earlier in Figs. 4.5.3 and 5.6.1.

The optimum value of λ was determined by the function splambda to be $\lambda = 3.6562$, but neighboring value would be just as good. The left graph shows the case of ordinary spline smoothing with no robustness iterations, and the right graph, using $N_{it} = 10$ iterations. The MATLAB code generating the right graph was as follows,

```
t = (0:50)'; u = t/max(t);
x0 = (1-cos(2*pi*u))/2;                        % noise-free signal

seed=2005; randn('state',seed);
```

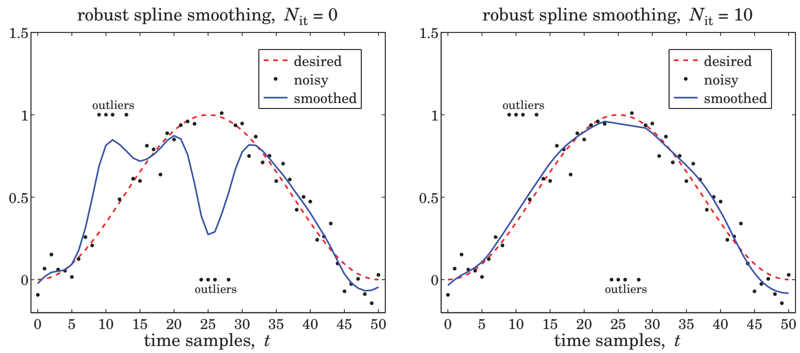


Fig. 7.9.2 Robust spline smoothing.

```
y = x0 + 0.1 * randn(size(x0));          % noisy observations

m = [-1 0 1 3];                           % outlier indices relative to n0, n1
n0=25; y(n0+m+1)=0.0;
n1=10; y(n1+m+1)=1.0;

la = splambda(t,y,1,10,30);               % optimum λ = 3.6562

w = ones(size(t)); Nit = 10;              % initial weights
[P,ta] = splsm2(t,y,la,w,Nit);            % robust spline smoothing
ya = P(:,1);                              % smoothed values

plot(t,x0,'--', t,y,'.', ta,ya,'-');      % right graph
```

The optimum λ was searched for in the interval $1 \le \lambda \le 10$ calling `splambda` with 30 golden-mean iterations. The resulting knot vector t_a has length 42, while the original vector t had length 51. The missing knot times correspond to the positions of the outliers.

Example 7.9.3: *NIST ultrasonic data.* We apply spline smoothing to a nonlinear least-squares benchmark example from the NIST Statistical Reference Dataset Archives. The data file `Chwirut1.dat` is available online from the NIST web sites:

```
http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml
http://www.itl.nist.gov/div898/strd/nls/data/chwirut1.shtml
```

The data are from a NIST study involving ultrasonic calibration and represent ultrasonic response versus metal distance. There are multiple observations for each distance. In fact, there are 214 observation pairs (x_n, y_n), but only 22 unique x_ns. The data have been fit by NIST using a nonlinear least squares method to a function of the form:

$$y = \frac{\exp(-b_1 x)}{b_2 + b_3 x} \qquad \text{(NIST fit)}$$

with the following fitted parameter values:

$$b_1 = 0.19027818370\,, \quad b_2 = 0.0061314004477\,, \quad b_3 = 0.0010530908399$$

The right graph in Fig. 7.9.3 compares the smoothing spline curve (solid line) with the above NIST fit (dotted line). Except for the rightmost end of the curves, the agreement is very close and the curves are almost indistinguishable.

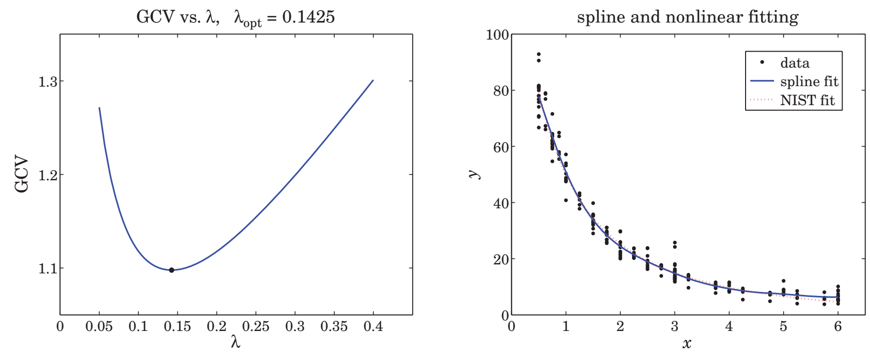


Fig. 7.9.3 Spline smoothing vs. Nonlinear fitting.

The function `splav` is called first to determine the unique x_ns and the corresponding averaged observations and their multiplicities. These are then used in `splambda` to determine the optimum GCV smoothing parameter, $\lambda_{opt} = 0.1425$, which is used by the function `splsm` to perform the spline smoothing fit. The GCV is evaluated and plotted at a range of λs to illustrate its minimum. Finally, the spline is evaluated at dense grid of x-abscissas for plotting. The following code segment illustrates the computations:

```
Y = loadfile('Chwirut1.dat');                    % data file in OSP toolbox
x = Y(:,2); y = Y(:,1);                          % read (x,y) observation pairs

[x,i] = sort(x); y = y(i);                       % sort x's in increasing order

b1 = 1.9027818370E-01;  b2 = 6.1314004477E-03;  b3 = 1.0530908399E-02;

yf = exp(-b1*x)./(b2+b3*x);                      % NIST fit

[xa,ya,wa] = splav(x,y);          % unique x's, averaged observations, and multiplicities

la1=0.01; la2=1; Nit=20;                         % search interval
[lopt,gopt] = splambda(xa,ya,la1,la2,Nit,wa);    % determine optimum λ

la = linspace(0.05,0.4,51);                      % range of λ's
gcv = splgcv(xa,ya,la,wa);                       % evaluate GCV

figure; plot(la,gcv, lopt,gopt,'.');             % left graph

P = splsm(xa,ya,lopt,wa);                        % spline smoothing coefficients
xs = locgrid(xa,200);                            % evaluation grid of abscissas
ys = splval(P,xa,xs);                            % evaluate spline at xs

figure; plot(x,y,'.', xs,ys,'-', x,yf,':');      % right graph
```

7.10 Problems

7.1 Show that the matrices Q, S defined in Eqs. (7.3.13) and (7.3.30) satisfy the orthogonality property $Q^T S = 0$. Assuming that the weighting diagonal matrix W has positive diagonal entries, argue that the $N\times N$ matrix $A = [W^{-1/2}Q, W^{1/2}S]$ is non-singular. Using this fact, prove the projection matrix property (7.3.31). [*Hint:* work with $A(A^TA)^{-1}A^T$.]

7.2 Show that the Euler-Lagrange equation for the variational problem (7.6.1) is:

$$\dddot{x}(t) = \lambda^{-1}\sum_{n=0}^{N-1}\sum_{i=1}^{m_n} w_{ni}\big(y_{ni} - x(t_n)\big)\delta(t - t_n)$$

and show that it is equivalent to

$$\ddddot{x}(t) = \lambda^{-1}\sum_{n=0}^{N-1}\bar{w}_n\big(\bar{y}_n - x(t_n)\big)\delta(t - t_n)$$

where $\bar{w}_n, \bar{y}_n$ were defined in (7.6.2). This is an alternative way to establish the equivalence of the variational problems (7.6.1) and (7.6.3).

7.3 First prove the following Fourier transform pair:

$$\exp(-b|t|) \longleftrightarrow \frac{2b}{b^2 + \omega^2}$$

where b is any complex number with $\mathrm{Re}(b) > 0$. Then, use it to prove that Eqs. (7.7.4) and (7.7.5) are a Fourier transform pair. Show the same for the pair in Eq. (7.7.8).

8 Whittaker-Henderson Smoothing

8.1 Whittaker-Henderson Smoothing Methods

Whittaker-Henderson smoothing is a discrete-time version of spline smoothing for equally spaced data. Some of the original papers by Bohlmann, Whittaker, Henderson and others,† and their applications to trend extraction in the actuarial sciences, physical sciences, and business and finance, are given in [405–438], including Hodrick-Prescott filters in finance [439–467], and recent realizations in terms of the ℓ_1 norm [468–478], as well as extensions to seasonal data [622–625,636,638]. The performance index was defined in Eq. (7.1.2),

$$\boxed{\mathcal{J} = \sum_{n=0}^{N-1} w_n\big|y_n - x_n\big|^2 + \lambda\sum_{n=s}^{N-1}\big|\nabla^s x_n\big|^2 = \min} \tag{8.1.1}$$

where $\nabla^s x_n$ represents the backward-difference operator $\nabla x_n = x_n - x_{n-1}$ applied s times. We encountered this operation in Sec. 4.2 on minimum-R_s Henderson filters. The corresponding difference filter and its impulse response are

$$\begin{aligned} D_s(z) &= (1 - z^{-1})^s \\ d_s(k) &= (-1)^k\binom{s}{k}, \quad 0 \le k \le s \end{aligned} \tag{8.1.2}$$

For example, we have for $s = 1, 2, 3$,

$$\mathbf{d}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad \mathbf{d}_2 = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}, \quad \mathbf{d}_3 = \begin{bmatrix} 1 \\ -3 \\ 3 \\ -1 \end{bmatrix}$$

Because $D_s(z)$ is a highpass filter, the performance index attempts, in its second term, to minimize the spectral energy of x_n at the high frequency end, while attempting to interpolate the noisy observations with the first term. The result is a lowpass,

† Bohlmann considered the case $s = 1$, Whittaker and Henderson, $s = 3$, and Hodrick-Prescott, $s = 2$.