Chapter 10

Interpolation, Trend Removal, and Data Smoothing

Lancaster and Salkauskas' book *Curve and Surface Fitting: An Introduction* is a good reference on this topic, here designated L&S. It is recommended reading from cover to cover. These rough notes are based on Chapters 1-6, which deal with curve fitting. The latter part of the book is devoted to surface fitting.

The basic problem we wish to consider in this chapter is finding a function that describes the continuum believed to lie behind a set of discrete observations. That is given a set of observations $(x_i, y_i), i = 1, \ldots, n$ how should we interpolate the data, find some underlying trend, or smooth the observations? The exact procedure to be followed depends on the ultimate goal, and the following considerations may be of importance in determining what strategies are appropriate to the problem at hand.

1. Criteria for Curve and Surface Fitting

(1) Do we want differentiability of the fitted curve and the smoothness that this implies? Should we choose curves that fall in C^r, the class of function that are continuous up to the rth derivative? If we use piecewise linear functions (or linear splines) the derivatives will be discontinuous at each knot point. Infinite differentiability could be achieved with a spline of degree N+1. But can the eye really distinguish among the class of curves C^r with $r \geq 2$?

(2) How confident are we in the data? Should we interpolate between the measured points, or is there experimental error that should be smoothed out of the fitted curve? Should we use least squares (LS) fitting to something like a smoothing spline or to piecewise linear or piecewise cubic functions. Or should we use filtering techniques to remove high frequency contributions? The latter topic will be dealt with in part II of this class.

(3) Global versus Local Techniques: Should data at a given point influence the nature of the fitted curve at distant points? If not how far should the influence be allowed to propagate? This depends on knowledge of physics, and density of data points. Classical polynomials and most splines are global, but they have differing degrees of attenuation as one moves away from data points. Under local schemes, such as piecewise linear and cubic polynomials one can add extra data and have the effect confined to a small neighborhood. Trends are global - they may be polynomial, exponential, oscillatory,.... and are often estimated using a

global least squares fit to the appropriate curve or surface. We have already discussed LS fitting at some length.

(4) Computational effort may increase rapidly with the degree of the fitted curve. Local techniques may offer significant computational advantages over global methods.

(5) Convergence: this is a mathematical issue. Suppose f is a theoretical function and in fitting n points we get the function g_n. As we add more data do g_n, $g_{n+1}, \ldots$ converge to f? How fast? How close? The relevance of convergence can often be unclear. There may be tradeoffs between rate of convergence and computational effort.

(6) Visual Criteria: Are there clear visual criteria for acceptability of a curve? These may be completely intuitive, and for geophysicists tend to involve a mixture of differentiability, smoothing, and frequency analysis, plus a (perhaps subconscious) consideration of bounds on curvature and the distance of the curve from a piecewise linear interpolant to all the data points. Quantifying these intuitions is important in searching for an appropriate and reproducible curve fitting technique.

2. Fitting Polynomials

Polynomials play a supremely important role in curve fitting to provide a continuous representation for data. The main reason for this is that they are easy to compute, although as we shall see ease of computation does not necessarily lead to a desirable outcome.

Let us begin with the idea of wanting to find a function underlying a set of observations: such a function can be used to fill the gaps between data. We need (1) a concept of fit to the observations, and (2) to decide what kind of functions should be used in the fit. It is possible that a unique function of the kind specified in (2) may exist that achieves the kind of fit specified in (1).

For example, suppose we decide that the fit should be exact, and we want the underlying function to be a polynomial of degree N or less, that is a member of the class we will call P_N. The following theorem indicates that in one regard at least this is a happy choice.

Theorem: If $x_0, x_1, x_2, \ldots, x_N$ form a set of $N + 1$ distinct numbers paired with $f_0, f_1, f_2, \ldots, f_N$, then there exists a unique polynomial in P_N, say $p(x)$ such that $p(x_i) = f_i$ for $0, 1, 2, \ldots, N$.

This polynomial will have degree N or less. Note that the Nth degree polynomial requires finding $N + 1$

coefficients. As a trivial example suppose that we have just 2 points x_0, f_0 and x_1, f_1. Clearly there is a unique straight line passing through each point $p(x) = a_0 + a_1 x$, and the variation in the curve between the points doesn't seem unreasonable, although we know nothing at all about how well it represents the unmeasured parts of the continuous function $f(x)$. However, L&S show in their Figure 2.1 (using $f(x) = \frac{1}{1+x^2}$), that the polynomial fit often does not satisfy one's intuition about what the underlying function should be.

Figure 10.1 goes here.

It is obvious that finding the values for the polynomial coefficients involves the solution of the following set of equations:

$$\begin{aligned} a_0 + a_1 x_0 + a_2 x_0^2 + \ldots a_N x_0^N &= f_0 \\ a_0 + a_1 x_1 + a_2 x_1^2 + \ldots a_N x_1^N &= f_1 \\ &\vdots \\ a_0 + a_1 x_N + a_2 x_N^2 + \ldots a_N x_N^N &= f_N \end{aligned} \tag{1}$$

or in the matrix notation we adopted earlier we have

$$V\vec{a} = \vec{f} \tag{2}$$

We know both V and $\vec{f}$ and need to find $\vec{a}$ which we can do (in principle) using

$$\vec{a} = V^{-1}\vec{f}, \tag{3}$$

noting that V^{-1} exists provided that the x_i are distinct. In this context V is often know as the Vandermonde matrix, and is notoriously ill-conditioned. Alternative methods involve using Lagrange's method or Hermite interpolation which effectively use different bases (or cardinal functions) to represent the polynomial. **Cardinal functions** $L_i(x)$ explicitly give the resulting interpolant $p(x)$ in the form of a weighted sum of the functional values f_i

$$p(x) = \sum_{i=0}^{N} f_i L_i(x).$$

Hermite interpolation is a generalization of the problem we have considered so far and allows the specification of the slopes f_i' in addition to the functional values f_i at positions x_i.

The polynomial interpolation problem is easily extended to include the idea of LS fitting of polynomials to a set of data, a topic we have considered in some detail already. LS fitting was introduced in this context independently by Gauss, Lagrange and others, and in the notation used here we would minimize

$$S(p) = \sum_{i=0}^{n} [p(x_i) - f_i]^2. \tag{4}$$

Recall from Chapter 7 that this gives

$$V^T V \vec{a} = V^T \vec{f}. \tag{5}$$

When the number of parameter equals the number of data this gives an exact fit and the polynomial interpolant. Hence

$$\vec{a} = (V^T V)^{-1} V^T \vec{f} \tag{6}$$

In using LS fitting we introduce the idea of statistical variations in the data, and admit that interpolation could in some sense produce excessive oscillations in the resulting function. LS fitting then can be thought of as a kind of smoothing process used to isolate the reliable part of a signal from measurement error. Although in practice the fitting procedure may be the same, smoothing may be distinguished from trend removal, in that the latter may simply be a mechanism for separating the signal into distinct parts, perhaps corresponding to different physical processes. The trend is the part that varies more slowly with the independent variable, and the rest is the residual or deviation from the trend.

3. Moving Least Squares Fits (LOESS or LWSS)

An important variant on LS fitting is the idea of a moving least squares fit - variously described as LOESS (LOWESS) or LWSS see (*e.g.*, Cleveland, 1979). This method finds wide application in surface interpolation with scattered data (see *e.g.*, L&S, Chapter 10), but here we introduce it in the context of curve fitting as introduced by L&S, Chap 2.9-2.11).

The basic idea is just to modify S in equation (4) above, so that there exist positive weights w_i that depend on x with each deviation and minimize

$$S_x(p) = \sum_{i=0}^{N} w_i(x)[p(x_i) - f_i]^2 \tag{7}$$

Suppose now that $w_i(x)$ is positive and relatively large close to x_i, and relatively small for more distant x_i, with a monotonic decrease as $|x - x_i|$ increases, and that now

$$p(x) = \sum_{i=0}^{m} a_i x^i. \tag{8}$$

Now we could solve the normal equations in the usual way with m = 1 or 2 for example, but we would need to include the weights, and to do so for each x, and the resulting coefficients would depend on x, so this is not in general a polynomial solution. We designate the resulting function $g(x)$.

The above is known as a **moving least squares fit** (MLS), and requires that one specifies a distribution of weights for each x. For example one can use

$$w_i(x) = exp\{-\frac{(x - x_i)^2}{50}\}, \tag{9}$$

where the weights have the same underlying function, but the center of symmetry moves. The matrix description for Moving LS is

$$V^T W(x) V\vec{a} = V^T W(x)\vec{f} \tag{10}$$

The effect of W is to draw the curve closer towards the data point than in ordinary LS. Note that the MLS needs at least (m + 1) data points in the region of support for $w(x)$, otherwise the normal equations will be singular. That is we need m + 1 non-vanishing weights in (10).

A limiting form of this is the interpolating MLS method, where the weight function then forms a basis for interpolation. By a suitable choice of weights, one can interpolate some points and not others – using $w_i(x) \to \infty$ for interpolation. The weight functions used for interpolation are something like inverse even powers of $|x|$, for example, $w(x) = \frac{1}{x^2}$ or $\frac{1}{x^4}$, so that for each datum at position x_i one has for example

$$w_i(x) = \frac{1}{(x - x_i)^2} \tag{11}$$

or if very rapid attenuation with distance is desired

$$w(x) = \frac{exp(-x^2)}{x^2}. \tag{12}$$

The choice of w depends on the density of sampling, and whether there is a need for interpolation. If $w(x) = x^{-k}$, with $k > 0$ then the smoothness of g depends critically on k. We can also arrange for $w(x) = 0$ at sufficiently large x, using something like

$$w(x) = \begin{cases} ax^{-k}\{1 - |x|/d\}^2 & |x| \le d \\ 0 & |x| > d \end{cases} \tag{13}$$

One advantage of truncating $w(x)$ is that it produces a **local** interpolation scheme. The value of $g(x)$ is only determined by data near x – that is within the region of support $(x - d, x + d)$. Values outside this region do not affect $g(x)$.

Here is a very simple example. Suppose $m = 0$. The normal equations give

$$a_0(x) = \frac{\sum_{i=0}^{N} w_i(x) f_i}{\sum_i w_i(x_i)} \tag{14}$$

In a simple special case we set $w(x_i) = 1$ for $i = 0, 1, \ldots, N$ yielding the average of all the data

$$a_0 = \frac{\sum f_i}{N + 1} \tag{15}$$

yielding a straight line parallel to the x-axis. We could instead use a moving average which would be described by

$$w_i(x) = \begin{cases} 1 & x \in [x - \alpha, x + \alpha] \\ 0 & otherwise \end{cases} \tag{16}$$

We can compare these with the kernel function estimates for density estimation discussed in Chapter 9 for which we had (equation 9.4)

$$\hat{\phi}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\big(\frac{x - x_i}{h}\big),$$

while in the averaging scheme here we have a weighting of the observations by the kernel function,

$$\hat{p}(x) = \frac{\sum_{i=1}^{n} K(x - x_i) f_i}{\sum_{i=1}^{n} K(x - x_i)}.$$

This give most weight to observations close to x and least to those far from x. K is usually an even function specified up to an unknown smoothing parameter h selected by data-based methods. The moving average just corresponds to a boxcar kernel. For more on these issues see review article by Hastie and Loader entitled *Local regression: automatic kernel carpentry* (1993, Statistical Science, 8, 120–143; there's a link to the JSTOR electronic version on the class web site).

Interpolation, Splines, and Smoothing Splines

invertibel

4. Piecewise Polynomial Functions

As the number of data to be interpolated increases an increasingly flexible curve is needed for interpolation, and this can be achieved by increasing the degree of a polynomial $P(x)$ but only at the risk of introducing severe oscillations into the resulting function. An alternative is to keep the polynomial degree low, and make an interpolant by connecting pieces of polynomial functions. The best known approach to this is that of a cubic spline which we will get to shortly. First we will consider the simpler, piecewise linear function known as a linear spline.

With linear splines we demand continuity of the function, but not of its derivative, which can be discontinuous at so-called knots of the spline.

Figure 10.2

Let K be defined as a set of knots $\{k_0, k_1, \ldots, k_N\}$. These are real numbers with $k_0 < k_1, \ldots < k_N$. We define a function l for all real numbers x by

$$l(x) = a_0|x - k_0| + a_1|x - k_1| + \ldots + a_N|x - k_N|$$

with a_0, a_1, …, a_N fixed real numbers. $l(x)$ is a continuous piecewise linear function (a linear spline) with knot sequence K. Usually we want to do interpolation on a finite interval $[a, b]$ so we choose $k_0 = a$ and $k_N = b$ so that all knots lie in the interval $[a, b]$. If we choose the knots to coincide with x_i then there is a unique spline l (in the vector space $L_N(K)$ that satisfies $l(x_j) = f_j, j = 0, 1, \ldots, N$. If we don't put the knots at the x_i then the spline is not unique and there may be no interpolant.

If we want to find the coefficients that represent this linear spline then we write the following set of equations:

$$\begin{aligned} a_0|x_0 - x_0| + a_1|x_0 - x_1| + \ldots a_N|x_0 - x_N| &= f_0 \\ &\vdots \\ a_0|x_N - x_0| + a_1|x_N - x_1| + \ldots a_N|x_N - x_N| &= f_N \end{aligned} \tag{17}$$

yielding a matrix V with zeroes on the diagonal,

$$V\vec{a} = \vec{f}$$

This matrix is invertible, but it is more convenient to use a *cardinal basis*, that is a set of functions l_i from $L_N(K)$ with each l_i solving the interpolation problem

$$l_i(x_k) = \delta_{ik} \qquad k = 0, \ldots, N \text{ and } i = 0, \ldots, N$$

The l_i are pyramid or tent functions:

Figure 10.3

$$\begin{aligned} l_0(x) &= \frac{x - x_1}{x_0 - x_1} & a \le x \le x_1 \\ &= 0 & x_1 \le x \le b \\ l_j(x) &= 0 & a \le x \le x_{j-1} \\ &= \frac{x - x_{j-1}}{x_j - x_{j-1}} & x_{j-1} \le x \le x_j \\ &= \frac{x - x_{j+1}}{x_j - x_{j+1}} & x_j \le x \le x_{j+1} \\ &= 0 & x_{j+1} \le x \le b \end{aligned}$$

The interpolant is then

$$l(x) = l_0(x)f_0 + \ldots l_N(x)f_N = \vec{l}(x)^T \vec{f}$$

and in fact we can represent any function in $L_N(K)$ in the form

$$l(x) = l_0(x)y_o + l_1(x)y_1, + \ldots l_N(x)y_n = \vec{l}(x)^T \vec{y}$$

with the y_i the parameters controlling the functional form of $l(x)$.

5. LS Approximation by Linear Splines

We can use these tent functions in LS fitting to give an approximate function rather than an interpolant. Then we would work in a vector space $L_M(K)$ with smaller dimension $1 \le M \le N$, with knot sequence K such that $k_0 = a$, $k_M = b$ and $k_0 < k_1 < k_2 \ldots < k_M$. The interior knots are then unrelated to sample points. Of course $M = 1$ corresponds to linear LS fit. If $M = 2$ there is one interior knot between a and b and the residual sum of squares to be minimized is

$$S(l) = \sum_{j=0}^{N} \{ \sum_{i=0}^{M} y_i l_i(x_j) - f_j \}^2$$

With this scheme one must be careful to locate the knots so that there are actually data in support of each l_i, that is in regions where $l_i(x) \neq 0$. As usual one finds the LS solution from $\frac{\partial S}{\partial y_k} = 0$ for $k = 0, \ldots, M$. With this linear spline, $l'(x)$ is piecewise constant and we can write

$$l'(x) = A + \sum_{i=1}^{N-1} a_i J(x - k_i)$$

with

$$\begin{aligned} J(x) &= -1 \qquad x < 0 \\ &= +1 \qquad x \geq 0 \end{aligned}$$

6. Interpolation with Piecewise Cubic Functions

From Lancaster & Salkauskas (1986, p.78). There exists a unique function S in $C_N(K)$ that has prescribed values f_i and prescribed slopes $m_i,\ i = 0, 1, \ldots, N$ at the knots k_i. $S(x)$ can be written in the form

$$S(x) = \frac{Ax^3}{3} + \frac{Bx^2}{2} + Cx + D + \sum_{i=1}^{N-1} a_i \frac{1}{6}|x - k_i|^3 + \sum_{i=1}^{N-1} b_i \frac{1}{2}|x - k_i|^2 J(x - k_i) \tag{18}$$

where

$$J(x) = \begin{cases} -1, & x < 0 \\ 1, & x \geq 0 \end{cases} \tag{19}$$

If there are no discontinuities in $S"(x)$ then in (18) all the $b_i = 0$ and $S(x)$ is a cubic spline. That is a cubic spline on the set of knots K has the form

$$S(x) = \frac{Ax^3}{3} + \frac{Bx^2}{2} + Cx + D + \sum_{i=1}^{N-1} a_i \frac{1}{6}|x - k_i|^3 \tag{20}$$

If K is a set of $N + 1$ distinct knots on $[a, b]$, then the family $S_N(K)$ specified by (20) is a vector space of dimensions $N + 3$ and is a subspace of $C^2[a, b]$. Unique interpolation at $N + 1$ knots in $[a, b]$ is not possible with cubic splines. There are two remaining degrees of freedom.

7. Piecewise cubic interpolation with one continuous derivative

Suppose now that we have two distinct kinds of data at each knot point, namely f_i the value of the function and m_i, the slope, then we need $2N + 2$ cardinal functions $\Phi_i, \Psi_i,\ i = 0, 1, \ldots, N$ in $C_N(K)$ such that

$$\Phi_i(x_j) = \delta_{ij} \qquad \Phi_i'(x_j) = 0 \qquad \Psi_i(x_j) = 0 \qquad \Psi_i'(x_j) = \delta_{ij} \tag{21}$$

Then the interpolant is

$$S(x) = \sum_{i=0}^{N} \Phi_i(x) f_i + \sum_{i=0}^{N} \Psi_i(x) m_i \tag{22}$$

The cardinal functions Φ are uniquely determined by the values and slopes at the ends of an interval. For $i = 1, 2, \ldots, N-1$ then $\Phi_i(x) \equiv 0$ when $x < x_{i-1}$ or $x > x_{i+1}$. Also $\Phi_0(x) \equiv 0$ when $x > x_1$ and $\Phi_N(x) \equiv 0$ when $x < x_{N-1}$. The cardinal functions have the property of small support. For $i = 1, \ldots, N-1$ and $h_i = x_i - x_{i-1}$

$$\Phi_i(x) = \begin{cases} 0 & x < x_{i-1} \\ -(2/h_i^3)(x - x_{i-1})^2(x - x_i - \frac{1}{2}h_i), & x_{i-1} \le x < x_i \\ (2/h_{i+1}^3)(x - x_{i+1})^2(x - x_i + \frac{1}{2}h_{i+1}), & x_i \le x < x_{i+1} \\ 0 & x \ge x_{i-1} \end{cases} \tag{23}$$

For $i = 0$ and $i = N$ only the last or first two parts of the definition apply. Similarly one can construct the Ψ_i. For $i = 1, \ldots, N-1$ and $h_i = x_i - x_{i-1}$

$$\Psi_i(x) = \begin{cases} 0 & x < x_{i-1} \\ (h_i^2)^{-1}(x - x_{i-1})^2(x - x_i), & x_{i-1} \le x < x_i \\ (h_{i+1}^2)^{-1}(x - x_{i+1})^2(x - x_i), & x_i \le x < x_{i+1} \\ 0 & x \ge x_{i-1} \end{cases} \tag{24}$$

Often only the f_i are available in which case we can manipulate the values of the slopes and adjust the interpolating curve to taste: One could determine slopes from the ordinates in linear or non-linear ways. When the slopes m_i are found in a linear way, we can write

$$\begin{pmatrix} m_0 \\ \vdots \\ m_N \end{pmatrix} = M \begin{pmatrix} f_0 \\ \vdots \\ f_N \end{pmatrix} \tag{25}$$

With this we can write

$$S(x) = [\phi_0(x), \ldots, \phi_N(x)] \begin{pmatrix} f_0 \\ \vdots \\ f_N \end{pmatrix} \tag{26}$$

where we now have

$$\phi_i(x) = \Phi_i(x) + [\Psi_0(x), \ldots, \Psi_N(x)]\, M_i \tag{27}$$

and M_i is the ith column of M.

Two choices for evaluating M are discussed by L& S: The linear one uses the slope at x_i of the unique parabola that interpolates at x_{i-1}, x_i, x_{i+1} (Bessel's method); a non- linear choice given by Akima (1970) is

$$m_i = \frac{|S_{i+2} - S_{i+1}|S_i + |S_i - S_{i-1}|S_{i+1}}{|S_{i+2} - S_{i+1}| + |S_i - S_{i-1}|}$$

with S_k the slope of the line segment joining points (x_{k-1}, f_{k-1}) and (x_k, f_k) for $k = i-1,\ i,\ i+1,\ i+2$.

8. Cubic Splines

With a given knot sequence K, $a = k_0 < k_1 <, \ldots, < k_N = b$ a cubic spline, S(x), is a cubic polynomial in each subinterval $[k_i, k_{i+1}]$, with the property that the function constructed from the segments has two continuous derivatives on $[a, b]$.

Let us repeat that $S(x)$ can be written

$$S(x) = \alpha x^3 + \beta x^2 + \gamma x + \delta + \sum_{i=1}^{N-1} a_i |x - k_i|^3 \tag{28}$$

for some constants $\alpha, \beta, \ldots$. Direct estimation of these coefficients is not a good idea.

8:1 Cardinal Basis for Cubic Splines

We can go back to the cardinal basis representation of the piecewise cubic function in equation (22) which has only one continuous derivative

$$S(x) = \sum_{i=0}^{N} \Phi_i(x) f_i + \sum_{i=0}^{N} \Psi_i(x) m_i, \tag{29}$$

and impose the condition that $S"(x_k^-) = S"(x_k^+)$. These conditions supply $N - 1$ constraints for the $N + 1$ slopes of the cubic splines $(m_0, \ldots, m_N)$. Two degrees of freedom remain, and one must supply two more equations involving m_0 and m_N to provide a unique solution. These can be written as

$$2m_0 + \mu_0 m_1 = c_0, \qquad \lambda_N m_{N-1} + 2m_N = c_N \tag{30}$$

where $\mu_0,\ \lambda_N,\ c_0$, and c_N are parameters to be chosen appropriately. We introduce the notation

$$\begin{aligned} \lambda_k &= \frac{h_{k+1}}{h_k + h_{k+1}} \qquad\qquad \mu_k = 1 - \lambda_k, \\ c_k &= 3\lambda_k[\frac{f_k - f_{k-1}}{h_k}] + 3\mu_k[\frac{f_{k+1} - f_k}{h_{k+1}}], \qquad\qquad k = 1, \ldots, N-1. \end{aligned} \tag{31}$$

With this in place the constraint equations for the slopes are

$$\begin{pmatrix} 2 & \mu_0 & 0 & \ldots & 0 & 0 & 0 \\ \lambda_1 & 2 & \mu_1 & \ldots & 0 & 0 & 0 \\ 0 & \lambda_2 & 2 & \ldots & 0 & 0 & 0 \\ & & & & & & \\ 0 & 0 & 0 & \ldots & 2 & \mu_{N-2} & 0 \\ 0 & 0 & 0 & \ldots & \lambda_{N-1} & 2 & \mu_{N-1} \\ 0 & 0 & 0 & \ldots & 0 & \lambda_N & 2 \end{pmatrix} \begin{pmatrix} m_0 \\ m_1 \\ m_2 \\ \vdots \\ m_{N-2} \\ m_{N-1} \\ m_N \end{pmatrix} = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{N-2} \\ c_{N-1} \\ c_N \end{pmatrix} \tag{32}$$

Equations (31) give some control in specifying the various end conditions on the spline. *e.g.*, to specify the numerical value of the slope of $S(x)$ at the left end one would set

$$\mu_0 = 0 \qquad\qquad c_0 = 2S'(x_0) \tag{33}$$

to get $m_0 = S'(x_0)$. Similarly one could set $\lambda_N = 0$ at the right end.

Controls on the second derivative $S"(x_0)$ and $S"(x_N)$ can be exercised by setting $\mu_0 = \lambda_N = 1$ and

$$\begin{aligned} 2m_0 + m_1 &= 3[\frac{(f_1 - f_0)}{h_1}] - \frac{h_1}{2}S"(x_0) \\ m_{N-1} + 2m_N &= 3[\frac{(f_N - f_{N-1})}{h_N}] - \frac{h_N}{2}S"(x_N) \end{aligned} \tag{34}$$

A natural cubic spline is a cubic spline with the additional properties $S"(x_0) = S"(x_N) = 0$. There is a unique cubic spline that interpolates, *i.e.*$S(x_j) = f_j$ for $j = 0, 1, \ldots, N$ together with one of the following conditions:

(1) prescribed slopes m_0, m_n at x_0, x_n respectively;

(2) prescribed second derivatives $S"(x_0)$ and $S"(x_N)$;

(3) end conditions of the form in Equation (14) with $\lambda_N < 4$ and $\mu_0 < 4$

(4) periodic end conditions

Another alternative is to impose the requirement that the third derivative be continuous at x_1 or x_{N-1}. This is also known as the *not a knot* condition and the constraints produce the following parameters for (31), (32).

$$\begin{aligned} \mu_0 &= \frac{2(h_1 + h_2)}{h_2} \\ c_0 &= \frac{2(3h_1 + 2h_2)(f_1 - f_0)}{(h_1 + h_2)h_1} + \frac{2h_1^2(f_2 - f_1)}{h_2(h_1 + h_2)} \end{aligned} \tag{35}$$

for x_1, and

$$\begin{aligned} \lambda_N &= \frac{2(h_{N-1} + h_N)}{h_{N-1}} \\ c_N &= \frac{2h_N^2(f_{N-1} - f_{N-2})}{h_{N-1}(h_{N-1} + h_N)} + \frac{2(2h_{N-1} + 3h_N)(f_N - f_{N-1})}{(h_{N-1} + h_N)h_N} \end{aligned} \tag{36}$$

for x_{N-1}.

If we write the RHS of (32) as $\mathbf{c} = C\mathbf{f}$ with C determined from (30) and (31), then we can see that the matrix M of (25) is just

$$M = A^{-1}C$$

with A the coefficient matrix in (32). Although A is tridiagonal, the product $A^{-1}C$ is full. Thus the cardinal splines are not of small support. This poses some inconvenience in computing. B-Splines are non-cardinal basis functions with minimal support.

8:2 B-spline Basis

Cubic splines are twice differentiable, and this means that a basis needs a minimal support of 4 consecutive intervals. If K is a knot sequence satisfying $k_0, k_1 <, \ldots, < k_m$, then for $j = 2, 3, \ldots, M-2$ there exists a choice of non-zero ordinates f_{j-1}, f_j, f_{j+1}such that the natural cubic spline with knot sequence K that satisfies

$$S(k_i) = \begin{cases} f_i, & i = j-1, j, j+1 \\ 0, & \text{otherwise} \end{cases} \tag{37}$$

vanishes outside the interval (k_{j-2}, k_{j+2}), and has zero slope at k_{j-2}, k_{j+2}.

For knots that are equally spaced a distance Δ apart, the cubic B-spline function has the following form

$$B_i(x) = \begin{cases} \frac{1}{6}(\frac{x-x_i}{\Delta} - 2)^3 & x \in [x_i, x_{i+1}], \\ \frac{2}{3} - \frac{1}{2}(\frac{x-x_i}{\Delta})^3 - (\frac{x-x_i}{\Delta})^2 & x \in [x_{i+1}, x_{i+2}], \\ \frac{2}{3} + \frac{1}{2}(\frac{x-x_i}{\Delta})^3 - (\frac{x-x_i}{\Delta})^2 & x \in [x_{i+2}, x_{i+3}], \\ \frac{1}{6}(2 - \frac{x-x_i}{\Delta})^3 & x \in [x_{i+3}, x_{i+4}], \\ 0 & \text{otherwise.} \end{cases} \tag{38}$$

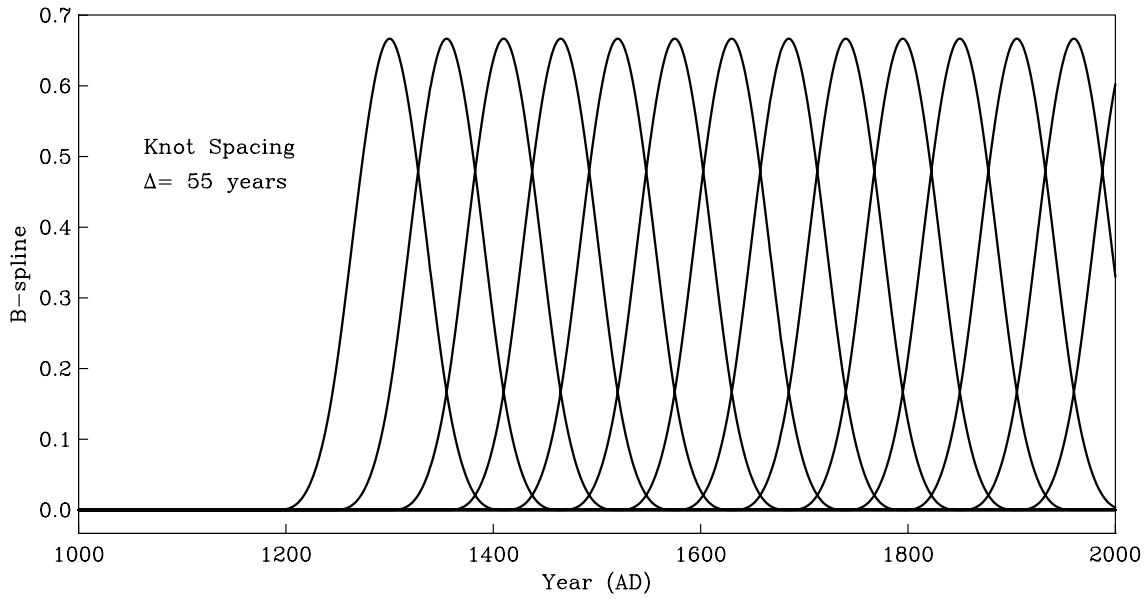

Figure 10.4: Example of a B-spline temporal basis with a knot spacing of 55 years

9. Smoothing Splines and Resolution

As with the linear splines one could do a LS fit to a subspace of cubic splines with the functional form for example of

$$p(t) = \sum_{k=1}^{M} \alpha_k B_k(t)$$

where the B_k are cubic B-spline basis functions. But a more interesting function might be one that finds the best fit function with minimum curvature. To do this the objective function minimized in finding $p(t)$ from data $(t_i, y_i), i = 1, \ldots, n$ on time interval (t_s, t_e) is

$$\sum_{i=1}^{n} (y_i - p(t_i))^2 + \tau \int_{t_s}^{t_e} [p''(t)]^2 dt \tag{39}$$

B-Splines form a basis with local support, but the curve-fitting procedure has an equivalent representation as a variable kernel smoother convolved with the data - valid except near ends of interval. This is presumably the cardinal basis. Silverman (1984) says we can write

$$\hat{p}(s) = n^{-1} \sum_{i=1}^{n} G(t_i, s) y_i \tag{40}$$

with

$$G(s,t) = \frac{1}{f(t)h(t)} \kappa \left(\frac{s-t}{h(t)} \right) \tag{41}$$

$h(t)$ is the local bandwidth of the equivalent kernel, and $f(t)$ is the local density of knot points.

$$\kappa(u) = \frac{1}{2} \exp \left(-\frac{|u|}{\sqrt{2}} \right) \sin \left(\frac{|u|}{\sqrt{2}} + \pi/4 \right) \tag{42}$$

$h(t)$ can be discovered once the fit has been performed and τ is known.

$$h(t) = \left[\frac{\tau}{n f(t)} \right]^{\frac{1}{4}} \tag{43}$$

τ controls the tradeoff between smoothness in $p(t)$ and fitting the data. Note that the distance from the central peak to the first zero of $\kappa \approx 3.32h$.

What does the equivalent kernel look like near a boundary? For any point $t \in [t_s, t_e]$ we define

$$\delta = \min(t - t_s, t_e - t)$$

as the distance from t to the nearest boundary. We want to find the equivalent kernel when δ is small. Define r and α, both depending on t and λ as

$$r\cos\alpha = 1 - 2\sin(\frac{2^{\frac{1}{2}}\delta}{h}) \qquad\qquad r\sin\alpha = 1 - 2\cos(\frac{2^{\frac{1}{2}}\delta}{h}) \tag{44}$$

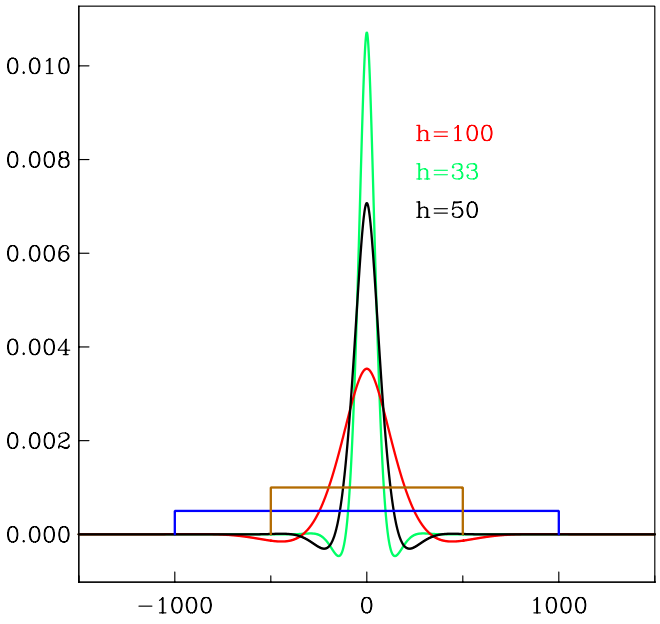

Figure 10.5: Examples of the equivalent kernel smoother for various B-spline temporal bases with bandwidth h ranging from 33 to 100 years. Also shown for comparison are two boxcar smoothers

and a kernel $\kappa^*(u)$ also implicitly dependent on t and λ by

$$\kappa^*(u) = -2^{-\frac{3}{2}} \exp\left(-\frac{|u|}{\sqrt{2}}\right) \sin\left(\frac{|u|}{\sqrt{2}} - \alpha\right). \tag{45}$$

Now let t^* be the reflection of t in the nearest boundary, that is

$$t^* = \begin{cases} t - 2\delta & \text{if } t < \frac{1}{2}(a+b) \\ t + 2\delta & \text{if } t > \frac{1}{2}(a+b) \end{cases} \tag{46}$$

Then for s and t in $[t_s, t_e]$

$$G(s,t) \equiv \frac{1}{f(t)h(t)} \left\{ \kappa\left(\frac{s-t}{h(t)}\right) + \kappa^*\left(\frac{s-t^*}{h(t)}\right) \right\}. \tag{47}$$

In the limiting case where $\delta = 0$ this yields for $x \geq 0$

$$\kappa(x) + \kappa^*(x) = 2^{\frac{1}{2}} \exp\left(-\frac{x}{\sqrt{2}}\right) \cos\left(\frac{x}{\sqrt{2}}\right). \tag{48}$$

The equivalent kernel representation provides a means of evaluating h after the fit is performed for a smoothing spline. When the fitting procedure makes use of a depleted cubic spline basis ($M < N$) and an approximation to a smoothing spline the calculation is a bit more complicated (see Constable & Parker, 1988, for details). In CP88 it is argued that the approximation is reasonable provided the knot spacing, Δ, is chosen so that $\Delta < 1.65h$. For the Δ of 55 years used in Figure 10.3 we can accordingly expect reliable results with this depleted basis if $h > 33$ years. Users may prefer to think in terms of the distance from

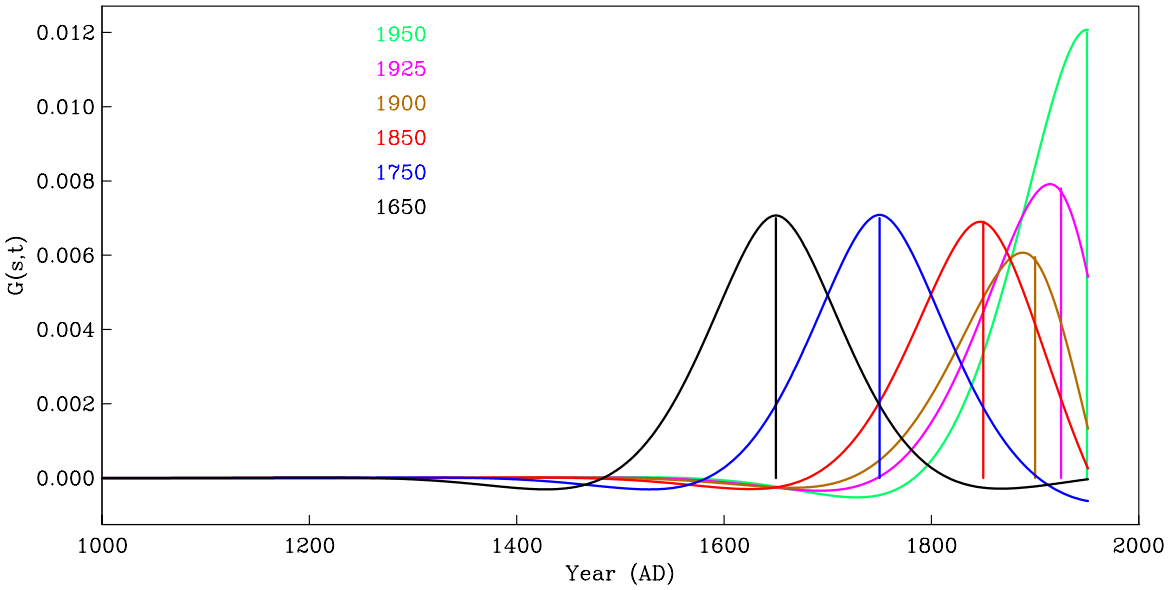

Figure 10.5: Equivalent kernel representation near a boundary with $h = 50$.

the central peak to the first zero crossing for κ, here designated h_z. The minimum is h_z then $3.32h$, around 100 years. Loosely speaking we see that by the most optimistic measure the kernel should average data in a window about 2 centuries wide; otherwise the approximation to a smoothing spline will be poor. The exact calculation of h for real data series may be complicated for a number of reasons: different weights are applied to individual data in the modeling; the temporal distribution of the observations is uneven; and the functional predictions may be non-linearly related to the model parameters.

Cleveland, W. S., 1979. Robust locally weighted regression and smoothing scatterplots. *Journal of American Statistical Association*, **74**, 829–836.

Constable, C.G., & R.L. Parker, (1988). Smoothing, splines and smoothing splines: their application in geomagnetism. *J. Comp. Physics*, **78**, 493–508.

Hastie & Loader, 1993. Local regression: Automatic kernel carpentry. *Statistical Science*, **8**, 120–143.

Lancaster, P., & M. Salkauskas, 1986. *Curve and Surface Fitting*. Academic Press. San Diego, California

Silverman, B.W., 1984. Spline smoothing: the equivalent variable kernel method. *The Annals of Statistics*, **12**, 898–916.