TECHNICAL REPORT NO. 555
March 1979

HOW TO SMOOTH CURVES AND SURFACES WITH SPLINES
AND CROSS-VALIDATION

Grace Wahba

This document has been approved for public release and sale; its distribution is unlimited.
DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.
How to Smooth Curves and Surfaces With Splines and Cross-Validation

by

Grace Wahba

ABSTRACT

We briefly review the use of smoothing splines and the method of
generalized cross validation (GCV) for smoothing discrete noisy data from an
unknown but smooth curve. Then we describe the use of "plaque mince" or
Laplace smoothing splines with GCV for smoothing discrete noisy data from an
unknown but smooth surface. A numerical algorithm for this (non-trivial!) computational
problem is described, and an example from a Monte Carlo study is presented to
show how the method works on simulated data. The results are extremely
promising. Some design problems are briefly mentioned. Some conjectures are
made concerning optimality properties of Laplace smoothing splines and
Laplace histosplines.

Note: This report was written for the Proceedings of the 24th Design of
Experiments Conference held at Madison, Wisconsin, May 3-5, 1978,
and sponsored by the U.S. Army Research Office.

Typist: Mary E. Arthur

This work was sponsored in part by the U.S. Army under Contract No. DAMD17-77-C-0807
and in part by the Office of Naval Research under Grant No. N00014-77-C-0675.

How to Smooth Curves and Surfaces With Splines and Cross-Validation

1. Introduction

In the conference talk we considered four problems. The first two had
to do with estimating curves when they are observed discretely and with
error. The model is

\[ y_i = f(x_i) + \varepsilon_i \quad i = 1, 2, \ldots, n \]

where \( f(x) \in [0,1] \) is an unknown curve, only known to be "smooth",
\( 0 \leq \varepsilon_1 \leq \cdots \leq \varepsilon_n \leq \varepsilon \), and \( \varepsilon_i \) are independent zero mean random variables
with a common unknown variance \( \sigma^2 \). The \( (y_i) \) are observed. The first problem
is: How should \( f \) be estimated nonparametrically from \( y = (y_1, \ldots, y_n) \)? The
second (or design) problem is: How should the points \( (x_i) \) be chosen so that
the estimate of \( f \) is as good as possible? The third and fourth problems have
to do with estimating surfaces. The model is

\[ z_i = u(x_i, y_i) + \varepsilon_i \quad i = 1, 2, \ldots, n \]

where \( u(x, y) \), \( (x, y) \in \text{some region in the plane, is only known to be "smooth".} \)
\( (x_i, y_i) \), \( i = 1, 2, \ldots, n \) are in a region in this region, the \( \varepsilon_i \) are zero mean
independent random variables with common unknown variance \( \sigma^2 \), and
\( z = (z_1, \ldots, z_n) \) is observed. The third problem is: How should \( u \) be
estimated nonparametrically from \( z \). The fourth (or design) problem is: How
should the points \( (x_i, y_i) \), \( i = 1, 2, \ldots, n \) be chosen so that the estimate of \( u \)
is as good as possible. We will not discuss the design problems here.
The design work mentioned in the talk has appeared in Athavale and Wahba (1978)
and Wahba (1971, 1974, 1976, 1978c). That work (and the work of others,
mentioned there) represents only some first steps in design problems for
nonparametric curve and surface fitting. There are many open problems.

Very good and relatively complete results for the first (curve estimation) problem are available (Craven and Wahba (1979), C.M., and Golub, Heath and Wahba (1977) (GHW), and translatable code is available from at least three sources Fleisher, (1979), Merz (1979a), and Pahua (1978). We will briefly summarize those results, because they will aid in understanding our discussion of the third problem, that is, surface smoothing. The remainder of this paper will then be devoted to the problem of smoothing of surface data nonparametrically. Some very nice theoretical results are available, and we have turned them into a computer program which delivers very pleasing pictures.

The development of the program is the work of Mr. James Wendelberger, and it and other results will appear in his Ph.D. thesis.

2. Curve Smoothing

For curve smoothing, we recommend that $f$ be estimated by the solution $f_{\mathfrak{P},\nu,s}$ of the minimization problem: Find $f \in H^1 = \{f: f, f', \ldots, f^{(m-1)} \text{ abs. cont.}, f_{[0,1]} \in L^2[0,1]\}$ to minimize

$$
\frac{1}{n} \sum_{i=1}^{n} (f(t_i) - y_i)^2 + \lambda \frac{1}{2} \int (f''(u))^2 du.
$$

(1)

The first term represents infidelity of $f$ to the data, and the second term represents "roughness" of the solution. The parameter $\lambda$ represents the tradeoff between the two. $\nu+2$ represents "psychological" smoothness (we think!) and is frequently used, and gives good results. We briefly discuss the determination of $\nu$ from the data later. The solution $f_{\mathfrak{P},\nu,s}$ is known to be a polynomial spline of degree $2\nu+1$. The parameter $\lambda$ is chosen from the data by the method of generalized cross-validation (GCV). GCV is derived from CV ("ordinary" cross validation). CV goes as follows: Let $f_{\mathfrak{P},\nu,s}$ be the solution of the minimization problem of (1) with the kth data point omitted. The value $\lambda$ will be a good choice if $f_{\mathfrak{P},\nu,s}(t_k)$ comes close, on the average, to $y_k$. We measure this by the "ordinary" cross validation function $V_0(x) = \frac{1}{V_0(x)}$

$$
V_0(x) = \frac{1}{n} \sum_{k=1}^{n} (f_{\nu,s}(t_k) - y_k)^2.
$$

For fixed $x$ the parameter $\lambda$ is chosen by minimizing $V_0(x)$. For technical reasons involving convergence proofs, we replace $V_0(x)$ by the generalized cross validation function

$$
V(x) = \frac{1}{n} \sum_{k=1}^{n} (f_{\nu,s}(t_k) - y_k)^2 s_k(x)
$$

where the $s_k(x)$ are certain weights to reflect unequally spaced data, end effects, etc. Details are given in GHW. It turns out that $V(x)$ is much easier to compute than $V_0(x)$, and $V(x)$ has the representation

$$
V(x) = \frac{1}{(n-1)^2} \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} x_i x_j (f_{\nu,s}(i) - f_{\nu,s}(j))^2}{\sum_{i=1}^{n} \sum_{j=1}^{n} x_i x_j (f_{\nu,s}(i) - f_{\nu,s}(j))^2}
$$

where $A(x)$ is the non matrix which is uniquely determined by

$$
\begin{pmatrix}
\int_{[0,1]} f_{\nu,s}(t_i) f_{\nu,s}(t_j) dt \\
\int_{[0,1]} f_{\nu,s}'(t_i) f_{\nu,s}'(t_j) dt \\
\vdots \\
\int_{[0,1]} f_{\nu,s}''(t_i) f_{\nu,s}''(t_j) dt
\end{pmatrix} = A(x)
$$

Pleasing results have been obtained using smoothing splines with GCV in both Monte Carlo studies and various applications, Benedetti (1977), C.M., GHW, Merz (1978a, 1978b), Stutkie (1977), Utteras (1978), Welch (1979). These results are not surprising in the light of the following theoretical result (DOM, GHW). Let

$$
R(x) = \frac{1}{n} \sum_{k=1}^{n} (f_{\nu,s}(t_k) - f(t_k))^2.
$$

$R(x)$ is the "true mean square error" averaged over the data points. Before data are observed both $V(x)$ and $V(x)$
can be considered functions of the unknown \( f \) and random functions of the 
\((x_i)\). Let \( \lambda^* \) be the minimizer of \( ER(\lambda) \) and \( \lambda \) be the minimizer of \( EV(\lambda) \).
We have under rather general circumstances (see CW, GCV)
\[
\sqrt{\frac{\text{VAR}(\lambda)}{\text{VAR}(\lambda^*)}} \leq 1.
\]
Thus (very loosely), the mean square error with the estimated \( \lambda \) tends to the
minimum mean square error achievable with any \( \lambda \). Let \( \lambda \) be the minimizer of
\( EV(\lambda) \). Numerical results based on Monte Carlo studies with \( m=2 \) reported in CW,
with \( n=50 \) and equally spaced data points, show the achieved inaccuracy
\( \text{VAR}(\lambda)/\text{VAR}(\lambda^*) \) in the range 1.01 to 1.42.

Some numerical experiments to assess the effectiveness of choosing \( m \)
by GCV have been done. (Lucas, 1978). One obtains \( \hat{\lambda}(\lambda) \) for each \( \lambda \)
and minimizes \( \hat{\lambda}(\lambda) \) over \( \lambda \). The results indicate that this procedure does a
good job of picking out the \( m \) and \( \lambda \) which minimize \( \text{VAR}(\lambda) = \text{VAR}(\lambda^*) \), and that
there are classes of \( f \)'s for which it is worthwhile to do this, that is,
\( \text{VAR}(\lambda^*) \) is usefully less than \( \text{VAR}(\lambda) \) for some \( m=2 \). Efficient
\( \lambda \) transportable code is not presently available, however. Depending on \( f \),
reduction in inaccuracy of several percent can be obtained.

3. Surface Smoothing

We now turn to the third problem, that of recovering smooth surfaces.
We recommend that \( m \) be estimated by the solution \( u_{n,m}^\lambda \) of the minimization
problem: Find \( u \in H \) (an appropriate space, to be described) to minimize
\[
\frac{1}{n} \sum_{i=1}^{n} (u(x_i, y_i) - z_i)^2 + \lambda \int \int \left( \frac{\partial^2 u}{\partial x^2} \right)^2 + \left( \frac{\partial^2 u}{\partial y^2} \right)^2 \text{ dxdy}.
\]
\[
\text{(3)}
\]
and that \( \lambda \) (and possibly \( n \)) be estimated by GCV. We now describe how to do
this. For mathematical convenience the limits on the double integral in
\[
\text{(3)}
\]
are taken to be \( -\infty \) and \( +\infty \). \( \lambda \) is taken as \( \lambda = \text{VAR}(\hat{\lambda}) = (u, \psi \cdot D, \psi \cdot D)^{-1} \\text{VAR}(\hat{\lambda}) \).
\[
\text{(4)}
\]
Let \( m \geq 2 \) and \( n \geq \lambda \). The solution \( u_{n,m}^\lambda \) to the problem: Find
\( u \in H \) to minimize
\[
\frac{1}{n} \sum_{i=1}^{n} (u(x_i, y_i) - z_i)^2 + \lambda \int \int \left( \frac{\partial^2 u}{\partial x^2} \right)^2 + \left( \frac{\partial^2 u}{\partial y^2} \right)^2 \text{ dxdy}.
\]
\[
\text{(4)}
\]
is given by
\[
\text{(5)}
\]
where
\[
\text{(6)}
\]
and that \( \lambda \) (and possibly \( n \)) be estimated by GCV. We now describe how to do
this. For mathematical convenience the limits on the double integral in
\[
\text{(3)}
\]
are taken to be \( -\infty \) and \( +\infty \). \( \lambda \) is taken as \( \lambda = \text{VAR}(\hat{\lambda}) = (u, \psi \cdot D, \psi \cdot D)^{-1} \\text{VAR}(\hat{\lambda}) \).
\[
\text{(4)}
\]
Let \( m \geq 2 \) and \( n \geq \lambda \). The solution \( u_{n,m}^\lambda \) to the problem: Find
\( u \in H \) to minimize
\[
\frac{1}{n} \sum_{i=1}^{n} (u(x_i, y_i) - z_i)^2 + \lambda \int \int \left( \frac{\partial^2 u}{\partial x^2} \right)^2 + \left( \frac{\partial^2 u}{\partial y^2} \right)^2 \text{ dxdy}.
\]
\[
\text{(4)}
\]
is given by
\[
\text{(5)}
\]
where
\[
\text{(6)}
\]
and that \( \lambda \) (and possibly \( n \)) be estimated by GCV. We now describe how to do
this. For mathematical convenience the limits on the double integral in
\[
\text{(3)}
\]
shown to be Bayes estimates with a certain (partially improper) prior on \( f \) or \( u \), see Wahba (1978a).

4. An Algorithm for Computation of the Smoothing Surface

We now want to compute \( u_{n,m,\lambda} \) efficiently, and choose \( \lambda \) (and possibly \( m \)) by GCV. Our algorithm below has benefited from the algorithmic work of Palma (1978). However it is different and seems especially well adapted to determining the generalized cross validation function \( V(\lambda) \) for this case. We next derive the equations behind our computational approach.

Let \( R \) be any \( n \times (n-M) \) dimensional matrix of rank \( n-M \) satisfying \( R^T R = G \). Subscripts indicate the dimensions of the subscripted matrix. Since \( \Gamma \Gamma^T = 0 \), we have

\[
\begin{align*}
\Gamma & = R_\gamma \\
\end{align*}
\]

for \( \gamma \) a unique \( n-M \) dimensional vector. Left multiplying (6) by \( R^T \) and substituting (8) into (6) gives

\[
\begin{align*}
R^T (R + I) R \gamma & = R^T R_\gamma \\
\gamma & = (R^T (R + I) R)^{-1} R^T R_\gamma \\
\gamma & = (R^T (R + I) R)^{-1} R^T (R + I) R \gamma \\
\gamma & = (R^T (R + I) R)^{-1} R^T (R + I) R \gamma \\
\end{align*}
\]

The vector \( \delta \) is then given by \( \delta = (\Gamma^T \Gamma)^{-1} \Gamma \zeta \), obtained by left multiplying (6) by \( \Gamma^T \). To estimate \( \lambda \) (equivalently \( \sigma \)) by GCV, we want to choose \( \lambda \) to minimize

\[
V(\lambda) = \frac{1}{2} \| (I - A(\lambda)) \zeta \|^2 \\
\leq \frac{1}{2} \text{Trace}(I - A(\lambda))^2
\]

where \( A(\lambda) \) is the \( n \times n \) matrix determined by

\[
\begin{align*}
& \begin{pmatrix}
u_{n,m,\lambda}(t_1) \\
\vdots \\
u_{n,m,\lambda}(t_n)
\end{pmatrix} = A(\lambda) \zeta \\
& \begin{pmatrix}
u_{n,m,\lambda}(t_1) \\
\vdots \\
u_{n,m,\lambda}(t_n)
\end{pmatrix} = A(\lambda) \zeta
\end{align*}
\]

To talk about good properties of GCV here, we suppose the \( \{t_i\} \) will be in a bounded region of the plane \( \mathbb{R}_2 \) (even though the minimization is over functions in \( \mathbb{R}_2 \)). The basic property (2) of GCV can then be shown to hold as the \( t_i \) become dense in this region - the proof (GWGM) is independent of the nature of the region.

To obtain a convenient representation for \( A(\lambda) \), we see from (5) that

\[
\begin{align*}
\zeta & = \begin{pmatrix}
u_{n,m,\lambda}(t_1) \\
\vdots \\
u_{n,m,\lambda}(t_n)
\end{pmatrix} \\
& = \begin{pmatrix}
u_{n,m,\lambda}(t_1) \\
\vdots \\
u_{n,m,\lambda}(t_n)
\end{pmatrix} = \zeta - K \zeta - T \lambda - T \lambda
\end{align*}
\]

From (6), we have

\[
z - T \lambda = (R + I) \zeta
\]

so that the right hand side of (11) equals \( \sigma \zeta \). Thus,

\[
(I - A(\lambda)) \zeta = \sigma \zeta = \sigma R (R + I) R \zeta
\]

We need to compute \( \sigma \), \( \| (I - A(\lambda)) \zeta \|^2 \), and \( \text{Trace}(I - A(\lambda)) \). Any \( R \times (n-M) \) will have a singular value decomposition

\[
R = U \Sigma V^T
\]

where \( U'U = V'V = I_{n-M} \) and \( D \) is diagonal. Then

\[
R (R + I) R \Sigma V = U V \Sigma V U U' = U (D + 1) U'
\]

Define

\[
R \Sigma V = U' U U'
\]
and define \( r \) and \( d \) by

\[
B = PAT,
\]

where \( P \) and \( T \) are the orthogonal and diagonal matrices in the eigenvalue decomposition of \( B \). Then the right hand side of (14) becomes

\[
U(s^{*}s)^{-1}u_{r} \quad \text{(15)}
\]

\[
= u_{r}^{*}(s^{*}s)u_{r}^{*} \quad \text{(16)}
\]

\[
= U^{T} \begin{bmatrix} 1 & b_{1} \cdots \ldots & 0 \\ 0 & \ddots & \vdots \\ 0 & \cdots & 1 \\ \end{bmatrix} u_{r}^{T} \quad \text{(17)}
\]

where \( b_{1}, \ldots, b_{n,M} \) are the diagonal entries in \( A \) (i.e. the eigenvalues of \( B \)).

Given \( U^{*} \), \( b_{1} \) we compute

\[
U = U^{T} \begin{bmatrix} 1 & b_{1} \cdots \cdots & 0 \\ 0 & \ddots & \vdots \\ 0 & \cdots & 1 \\ \end{bmatrix} u_{r}^{T} \quad \text{(18)}
\]

\[
\left| \left| (1-A(x)) \right| \right|^{2} = e^{2} \left| \left| (1-A(x)) \right| \right|^{2} \quad \text{(19)}
\]

\[
\left( \frac{1}{n} \operatorname{Tr}(A(x)) \right)^{2} = \left( \frac{1}{n} \sum_{i=1}^{n} a_{i} \right)^{2} \quad \text{(20)}
\]

We now discuss the determination of \( U \). It can be seen that \( U \)

is any matrix whose \( n-H \) columns are orthonormal and perpendicular to the \( M \)

columns of \( T \). \( U^{*}(n-H)^{n} T = 0(n-H)^{n} \). We obtained \( U \) as follows. Let

\[
I = T(T^{T}T)^{-1}T = \tilde{U} \tilde{U}^{*} \quad \text{(21)}
\]

where \( \tilde{U} \) is orthogonal and \( \tilde{A} \) is diagonal. Since \( I = (T^{T})^{-1} \)

a projection matrix of rank \( n-H \), \( \tilde{A} \) is a matrix with \( M \) zeros and \( n-H \) ones on the diagonal.

We used EISPACK (Smith et al. [1976]) to perform the eigenvalue decomposition

\( \tilde{U} \tilde{A} \tilde{U}^{*} \) and the \( n-H \) columns of \( \tilde{U} \) are taken as the columns of \( \tilde{U} \) corresponding to the \( n-H \) ones in \( A \). Each such vector is perpendicular to the columns of \( T \),

as can be seen by right multiplying (18) by \( T \). The EISPACK computation of the entries of \( A \) was good to seven figures. Given \( U \), \( B \) is computed and \( r \) and \( d \) are also computed using the eigenvalue decomposition routines in EISPACK.

5. Numerical Results

We present the results of a single Monte Carlo experiment, with \( m = 2 \).

Figure 1 gives a picture of the true function \( w \) that was the subject of the first experiment,

\[
w(x,y) = \frac{1}{2(1.3)^{2}} \left[ e^{- \frac{1}{2(1.3)^{2}}(x-2)^{2} + \frac{1}{2(1.3)^{2}}(y-2)^{2}} \right]
\]

A regular \( 7 \times 7 \) square array of 49 points \( t_{i}, i = 1,2, \ldots, 49 \) was selected, with the middle point being \((0,0)\) and the point spacing being 1.0. Data \( y_{i} \)

were generated as

\[
y_{i} = u(t_{i}) + e_{i}, \quad t_{i} = (x_{i},y_{i}), \quad i = 1,2, \ldots, 49
\]

where the \( e_{i} \) were \( N(0,\sigma^{2}) \) random variables with \( \sigma = .01 \). \( \sigma \) is

about 1/8 of the maximum height of \( w \). Figure 2 presents a picture of the data points, which have been joined by straight lines. Figures 3 and 4 give \( u_{n,M}^{2}, \lambda \) for two values of \( \lambda \). In Figure 3, \( \lambda \) is too large, and in Figure 4, \( \lambda \) is too small. Figure 5 gives \( u_{n,M}^{2}, \lambda \), where \( \lambda \) is the minimizer of \( V(\lambda) \). Figure 6 gives a plot of \( R(\lambda) \) and \( V(\lambda) \) against \( \log \lambda \). It is seen that, in the neighborhood of the minimizer of \( R(\lambda) \), \( V(\lambda) \) roughly follows \( R(\lambda) \). Theoretically, we have \( \min R(\lambda) \leq \min (R(\lambda) + x^{2}) \), for large \( n \), see (4), and this relationship is roughly approximated here. The achieved inefficiency, defined by

\[
R(\lambda)/\min R(\lambda), \text{ where } \lambda \text{ is the minimizer of } V(\lambda), \text{ was 1.54. Note that } \lambda \text{ is the minimizer of } V(\lambda).
\]

If we were fitting a surface which is known to be a linear combination of given functions by regression we would expect the mean square error to be proportional to \( \frac{2}{n} \). Here numerical and theoretical results in the one
dimensional case for reasonably regular arrangements of data points indicate
that \( \text{min } R(a) \geq \text{const.} a^p \), where \( p \) is some power slightly less than one. \( p \)
depends on the rate of decay of eigenvalues of an appropriate reproducing
kernel. See Wahba and Wold (1975), CI, and Wahba (1975b, 1977). If
\( u \in H^p (\mathbb{R}^2) \), \( p = 2m/(2m-1) \). [In preparation].

Mr. Wendelberger's program is running for \( n = 120 \) and quite reasonable
results have been obtained for this case, with randomly chosen points \( (t_j) \).
One cannot increase \( n \) with impunity, however. In the \( n = 49 \) case reported
here the condition number of \( S \), namely \( \max b_j/\min b_j \) was around 200, and in the
irregularly spaced \( n = 120 \) case this condition number was of the order of
\( 4n.10^8 \). (Irregularly spaced points will increase the condition number.)
For large \( n \) and a condition number somewhere around (we guess) \( 10^7 \) or \( 10^8 \),
the computer errors will begin to take over. Thus, in theory, a plot of
\( \log(\text{min } R(a)) \) vs. \( n \) should be approximately linear with slope \( -p \), however, as
roundoff error gets large, this plot will flatten out. Laurenz and colleagues
[1978] have developed a procedure for patching together surfaces of this type
so that groups of points may be handled separately.

The cost of running a program designed just to produce Figure 5 from
data, we estimate to be about \$10.00 at weekend rates at our computing center.
To produce Figure 5 from a second set of data at the same points \( (t_j) \), one
would retain \( U, R \) and \( (b_j) \), which depend on the \( (t_j) \) but not \( z \), and then the
cost would be very small.

6. Miscellaneous Remarks
We hope to implement the \( m = 3 \) and \( m = 4 \) cases, \( m \) can then be selected
from the data by comparing \( V(a) \) for each of the \( m = 2, 3 \) and 4 cases. For
\( m = 2 \), the roughness penalty

\[
\int_{\mathbb{R}^2} \left( \frac{1}{x^2+y^2} \right)^2 \text{dxdy}
\]
is the bending energy of a thin plate. For this reason, Duchon christened the
solutions "plaque fiscale" splines. We have reason to believe that the \( m = 3 \) case
will be appropriate for the smoothing of certain meteorological data. In some
cases the nature of the physical phenomena being smoothed may provide insight
into a choice of \( m \).

We note that the solutions \( u_{n,m,a} \) satisfy

\[
A^m u_{n,m,a}
\]

where \( a \) is the Laplacian operator \( \Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \). The smoothing splines
\( f_{n,m,a} \) satisfy \( f_{n,m,a}(t) = 0 \), \( t \neq t_1, \ldots, t_n \). For this reason, Prof. Iso
Schoenberg has suggested to us that the functions \( u_{n,m,a} \) be called "Laplacian
Smoothing Splines".

We have recently obtained what might be called the Laplacian histosplines,
by analogy with Boemers, Kendall, and Stefanov (1971). These are functions
which minimize the roughness penalty

\[
\int_{\mathbb{R}^2} \left( \frac{1}{x^2+y^2} \right)^2 \text{dxdy}
\]
subject to volume matching conditions of the form

\[
\int_{A_i} u \text{dxdy} = p_i, \quad i = 1, 2, \ldots, n
\]

where the \( A_i \) are \( n \) bounded areas in \( \mathbb{R}^2 \) whose union is \( A \). These functions satisfy

\[
L^m u = \text{constant on each } A_i
\]


Various optimality properties of smoothing splines and histosplines in one
dimension are known. For example, it can be shown from CI and Wahba (1975b) that

\[
E \left[ (f_{n,m,a}(t)-f(t))^2 \right] dt = 0(n^{-2m}/(2m+1)), \quad f \in H^m(0,1)
\]

\[
= 0(n^{-2m}/(4m+1)), \quad f \in W^m(0,1)
\]
and \( f \) satisfies some boundary conditions. It is part of the folklore that these rates cannot be improved upon. Density estimates determined by the minimizer of \( \int f(t)^2 \, dt \) subject to the area-matching conditions
\[
\int_{t_i}^{t_{i+1}} f(t) \, dt = \text{fraction of observations in } [t_i, t_{i+1}]
\]
are known to achieve the best possible convergence rates over \( f \in H^m \) provided the \( t_i \) are chosen properly. See Wahba (1975c, 1976). Stone (1978) has given some results on best possible pointwise convergence rates in \( d \) dimensions.

We conjecture that all the nice convergence properties of polynomial splines can be extended to the Laplacian smoothing splines and Laplacian histogramsm. 

---

**APPENDIX**

**Outline of Proof of Theorem**

Let \( r_1, r_2, \ldots, r_N \) be a subset of \( N \) points selected from \( t_1, \ldots, t_m \) with the property that the \( M \times M \) matrix \( T \) with \( j^{th} \) entry \( s_j(r_i) \) is of full rank. The space \( H = (u: u \in D', \frac{\partial^m u}{\partial x^m} \in L_2, j = 0, 1, \ldots, m-1) \) can be decomposed into the direct sum of two spaces:

\[
H = \mathbb{R}^{m-1} \oplus \mathbb{R}
\]

where \( \mathbb{R}^{m-1} \) is the \( M \) dimensional space of polynomials of total degree \( m-1 \) or less and \( \mathbb{R} = (u: u \in H, u(r_j) = 0, j = 1, 2, \ldots, N) \). It can then be shown that

\[
\langle u, v \rangle = \int_{0}^{1} \int_{0}^{1} u(x, y) v(x, y) \, dx \, dy
\]

defines an inner product on \( \mathbb{R} \). If an inner product is defined on \( \mathbb{R}^{m-1} \) by

\[
\langle u, v \rangle_{m-1} = \sum_{j=1}^{m-1} u(r_j) v(r_j),
\]

then \( \mathbb{R}^{m-1} \) and \( \mathbb{R} \) are orthogonal subspaces. \( \mathbb{R} \) (and \( \mathbb{R}^{m-1} \) and hence \( H \)) are reproducing kernel spaces.

If the reproducing kernel \( K(s, t) \) for \( \mathbb{R} \) can be found, then the solution \( u_{n,m,k} \) to the minimization problem of (4) will have a representation

\[
u_{n,m,k}(t) = \sum_{j=1}^{N} c_j K(t, t_j) + \sum_{j=1}^{m} d_j s_j(t). \tag{A.1}
\]

(See, e.g. Kimeldorf and Wahba (1971)). \( u_{n,m,k} \) will, of course, be independent of the choice of \( r_1, \ldots, r_N \). The reproducing kernel \( K \) has been found by Meinguet (1978, 1979) and is given by
\[ K(s, t) = E_m(s, t) - \sum_{m=1}^{M} \sum_{j=1}^{N} (r_j^m) E_m(r_j^m) \]
\[ - \sum_{m=1}^{M} \sum_{j=1}^{N} p_j(s) E_m(r_j^m) \]
\[ + \sum_{m=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{N} p_j(s) p_k(t) E_m(r_j^m r_k^m) \cdot \]
\[ (A.2) \]

where \((p_j^m)\) span \(\mathbb{V}_m\) and are chosen so that \(p_j(r_j^m) = 1\), \(s = v, r_0 = \gamma v\).

Substituting (A.2) into (A.1), it is seen that a representation of the form (5) for \(u, v, \lambda\) holds.

To show that \(K\) is the reproducing kernel for \(\mathbb{V}\), it is necessary to show that

1) \(K(s, \cdot) \in \mathbb{V}\), each \(s\),

\[ \langle u, K(s, \cdot) \rangle_{\mathbb{V}} = K(s, t) \cdot \]
\[ (A.3) \]

where

\[ \langle u, x \rangle_{\mathbb{V}} = \frac{1}{\lambda^2} \int_{\mathbb{V}} \frac{\partial^m u}{\partial x^m} \frac{\partial^m v}{\partial y^m} \, dy \cdot \]
\[ (A.4) \]

Define

\[ H_s(t) = E_m(s, t) - \sum_{j=1}^{N} p_j(s) E_m(r_j^m) \cdot \]

Then

\[ K(s, t) = H_s(t) - \sum_{j=1}^{N} p_j(s) H_j(r_j^m) \cdot \]
\[ (A.5) \]

The hard part is to show that \(H_s \in \mathbb{V}\). (Note that \(E_m \not\in \mathbb{V}\).) Meinguet shows that \(H_s \in \mathbb{V}\), for each \(s\), and we omit the proof. It then follows that

\[ K(s, \cdot) \in \mathbb{V}, \quad \text{and, since} \quad \sum_{j=1}^{N} p_j(s) H_j(r_j^m) \quad \text{is the polynomial interpolating to} \quad H_s \at r_1, \ldots, r_M, \quad K(s, r_j^m) = 0, \quad \forall \, v = 0, 1, \ldots, M, \quad \text{and so} \quad K(s, \cdot) \in \mathbb{V}. \]

To establish (A.3), first note that

\[ \frac{\partial^m}{\partial x^m} \frac{\partial^m}{\partial y^m} K(s, \cdot) = \frac{\partial^m}{\partial x^m} \frac{\partial^m}{\partial y^m} H(s, \cdot). \]
\[ (A.6) \]

Consider the Green's formula

\[ (\partial^m u)(s, t) \quad \text{is} \quad u(s) \quad \text{for} \quad u \in \mathbb{V} \cdot \]

where \(\lambda = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \cdot \) This formula holds provided, \(\forall \, v \in \mathbb{V} \cap L_2 \) and \(u \in \mathbb{V} \).

If \(u = 0\), then the potential formula

\[ \int (\partial^m u)(s, t) E_m(s, t) \, dt = u(s) \quad \text{holds (see Schwartz (1966)) and in particular} \]

\[ \int \langle a u, H_s \rangle \, dt = u(s) = \sum_{j=1}^{N} p_j(s) u(r_j^m) \cdot \]
\[ (A.8) \]

Meinguet argues that, in fact (A.7) and (A.8) hold for \(u = H_s, v = H_j\), giving

\[ (\partial^m u)(s, t) \quad \text{is} \quad u(s) \quad \text{for} \quad u \in \mathbb{V} \cdot \]

\[ \text{then} \quad H_s (s) \quad \text{is} \quad \sum_{j=1}^{N} p_j(s) H_j(r_j^m) \quad \text{in} \quad K(s, t) \cdot \]

which, combined with (A.6), gives (A.3).

Equation (7) can be obtained as follows: Considering \(K(t, t_j)\) as a function of \(t\),

\[ K(t, t_j) = E_m(t, t_j) - \sum_{j=1}^{N} p_j(t) E_m(t, r_j^m) \]
\[ + \text{a polynomial of degree } m-1 \text{ or less.} \]
\[ (A.9) \]

Now, if \(\alpha\) is any element of \(\mathbb{V}_m\), we have

\[ \alpha(t) = \sum_{j=1}^{N} p_j(t) \alpha(r_j^m) = 0 \cdot \]
\[ (A.10) \]

Letting \(e_1, e_2, \ldots, e_N\) be the coefficients of \(E_m(\cdot, t_j), E_m(\cdot, r_2), \ldots, E_m(\cdot, t_M)\), in (A.9), it can be verified from (A.10) that
\[ \sum_{j=1}^{n} k_{j}(t_c) = 0, \quad j = 1, 2, \ldots, n, \]

which results directly in the conditions (7) on the coefficient vector \( c \) in (5), namely, \( Tc = 0 \). Equation (6) is obtained as follows: One substitutes (A.1) into (4), and then uses (A.3) to evaluate the expression (4) to be minimized. By repeatedly using \( Tc = 0 \), one obtains that \( c \) and \( d \) are chosen subject to \( Tc = 0 \), to minimize

\[ \|z - Kc - Td\|^2 + \lambda c^T Kc. \]

Differentiating this expression with respect to \( c \) and setting the result equal to zero, and using \( Tc = 0 \), gives (6).

REFERENCES


Duchon, Jean (1975a). Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces. R.A.I.R.O. Analyse numerique, 10, 12, pp.5-12.


HOW TO SMOOTH CURVES AND SURFACES WITH SPLINES AND CROSS-VALIDATION

Grace Nahba

Department of Statistics
University of Wisconsin
Madison, Wisconsin 53706

U.S. Army Research Off.
P.O. Box 12211
800 N. Quincy Street
Res. Triangle Pk., N.C. 27709
Arlington, VA 22217

Distribution of this document is unlimited

Abstract: We briefly review the use of smoothing splines and the method of generalized cross validation (GCV) for smoothing discrete noisy data from an unknown but smooth curve. Then we describe the use of "plaque mince" or Laplacian smoothing splines with GCV for smoothing discrete noisy data from an unknown but smooth surface. A numerical algorithm for this (non-trivial) computational problem is described, and an example from a Monte Carlo study is presented to show how the method works on simulated data. The results are extremely promising. Some design problems are briefly mentioned. Some conjectures are made concerning optimality properties of Laplacian smoothing splines and Laplacian histosplines.

Keywords: splines, smoothing, estimation of smooth surfaces, multidimensional splines, Laplacian splines