CONF-9006124--2

SAND--89-2244C

DE90 009319

Spline Function Smoothing and Differentiation of Noisy Data on a Rectangular Grid

Clark R. Dohrmann
Sandia National Laboratories
Albuquerque, NM 87185

Henry R. Busby
The Ohio State University
Columbus, OH 43210

Received by OSTI

APR 1 6 1990

Abstract

A method is presented for smoothing and differentiating noisy data given on a rectangular grid. The method makes use of a one-dimensional smoothing algorithm to construct the solution to an associated two-dimensional problem. Smoothing parameter selection is automated using a technique that does not require prior knowledge of the amount of noise in the data. Numerical examples are provided demonstrating the application of the method.

Introduction

The need for functional representations of data given on a two-dimensional domain arises in a variety of fields. Depending upon whether the data is noisy or not, an approximating function is constructed to pass either near to (smoothing) or through (interpolation) the data points. The function can then be used to provide estimates for the data and its spatial derivatives over the entire domain. Two example applications are the contouring of geologic surfaces and the differentiation of Moiré fringe pattern data to obtain strain fields.

Polynomial spline functions have been widely used for both interpolating and smoothing empirical data. One of the major advantages of spline functions is their ability to pass smoothly through a large set of data points, contrary to what often occurs when a single, higher order polynomial function is used to interpolate data. Continuity of lower order derivatives also makes splines well-suited for numerical differentiation of data.

In a previous paper, dynamic programming was applied to the problem of smoothing one-dimensional data with splines, including selection of the smoothing parameter [1]. The method applies to the general case of unequally weighted and unequally spaced data. This paper is an extension of the above method for application to data given on a two-dimensional rectangular grid. In the interest of brevity, attention is restricted to bicubic spline approximation of data given on a uniform rectangular grid. Generalization of the method to higher order splines, nonuniform grids, and unequal weighting of the data is easily programmed, but requires a more lengthy explanation.

The two-dimensional smoothing algorithm is presented after a brief review of spline functions, one-dimensional smoothing, and smoothing parameter selection. Examples of smoothing and differentiating simulated data with bicubic splines are provided. We conclude with a brief summary of the merits and shortcomings of the present approach.

Background

An example of a cubic spline function having "knots" at $x_1, \ldots, x_5$ is shown in Figure 1. Between each adjacent pair of knots the spline is a separate cubic polynomial in the independent variable x. The cubic spline and its first and second derivatives are continuous throughout the interval $[x_1, x_5]$. The third derivative is piecewise constant with discontinuities occurring at the knots.

Consider the case of data given at $x_1, \ldots, x_n$, where $x_{i+1} = x_i + h$ $(i = 1, \ldots, n-1)$. For the knot sequence $\{x_i\}_1^n$, a cubic spline, $f(x)$, can be represented as a combination of $n+2$ basis functions

$$f(x) = \sum_{i=1}^{n+2} c_i s_i(x) \tag{1}$$

where

$$\begin{aligned}
s_1(x) &= 1 \\
s_2(x) &= (x - x_1) \\
s_3(x) &= (x - x_1)^2/2 \\
s_{i+3}(x) &= \begin{cases} 0 & \text{for } x < x_i \\ (x - x_i)^3/6 & \text{for } x_i \le x < x_{i+1} \\ h^3/6 + h^2(x - x_{i+1})/2 + h(x - x_{i+1})^2/2 & \text{for } x \ge x_{i+1} \end{cases}
\end{aligned} \tag{2}$$

For the one-dimensional problem, the object is to determine the coefficients, c_i, in Eq. 1 that will result in a smoothly varying approximation to the data. Stated more precisely,

Given: a set of data z_i taken at x_i $(i = 1, \ldots, n)$ and a smoothing parameter, μ (> 0).

Find: the cubic spline minimizing the functional

$$\phi_1(f) = \sum_{i=1}^{n} (f(x_i) - z_i)^2 + \mu \int_{x_1}^{x_n} \left(\frac{d^3 f}{dx^3} \right)^2 dx \tag{3}$$

A method for finding the f that minimizes ϕ_1 is given in the Appendix.

The value of the smoothing parameter can greatly affect the solution. In the limiting case of μ approaching zero, the spline function, f, passes through each data point. For extremely large values of μ, the third derivative of f is forced to zero throughout the interval $[x_1, x_n]$, resulting in a least squares quadratic fit.

The goal in selecting the smoothing parameter, μ, is to obtain a solution that passes near the data points while remaining reasonably smooth. Using values of μ that are too large often causes oversmoothing, resulting in a loss of the low frequency content of the data. Choosing μ too small also gives unsatisfactory results, especially when derivatives are estimated.

One technique that has been used successfully to choose the smoothing parameter in one-dimensional problems is generalized cross validation [2]. This technique is particularly useful for situations where there is not prior knowledge of the amount of noise in the data. Another technique, described later, is used when estimates of the levels of noise in the data are known. Regardless of whether the level of noise in the data is known or not, selection of the smoothing parameter requires the trace of the so-called influence matrix [2] and an iterative process of minimization.

The motivation for the current work was an interest in determining whether or not the methods developed for smoothing one-dimensional data could be extended to data given on a two-dimensional domain. A very helpful reference in our study was a paper by Hu and Schumaker [3]. In their paper the authors considered smoothing data given on a rectangular grid with tensor-product B-splines as basis functions. They did not, however, address the issue of selecting the smoothing parameter for cases where there is no prior knowledge of noise levels. Making use of the results in Reference 3, it was deduced that a one-dimensional smoothing algorithm could be used in the solution of an associated two-dimensional problem for data given on a rectangular grid. Furthermore, it also became clear that generalized cross validation could be used to select the smoothing parameter.

One-Dimensional Smoothing

As was mentioned previously, a technique has been developed for solving the one-dimensional smoothing problem based on a dynamic programming approach. The method consists of a backward and a forward "sweep" through the data and is outlined in the Appendix. Conceptually, the smoothing process can be viewed as a transformation of the data, z_i, into the constants c_i appearing in Eq. 1. This transformation is illustrated pictorially in Figure 2.

The constants c_i, as well as the values of the cubic spline and its derivatives at the knots, are calculated in a recursive fashion. Between two adjacent pair of knots $(x_i \le x < x_{i+1})$ the spline is given by the equation

$$f(x) = f(x_i) + \left.\frac{df}{dx}\right|_{x_i} (x - x_i) + \left.\frac{d^2 f}{dx^2}\right|_{x_i} (x - x_i)^2/2 + c_{i+3}(x - x_i)^3/6 \qquad (4)$$

Equation 4 is preferred over Eq. 1 for evaluating the spline between knots because it requires fewer calculations.

As a model of errors inherent in experimental measurements, consider the expression

$$z_i = g(x_i) + \epsilon_i \qquad i = 1, \ldots, n \tag{5}$$

where g is a smooth curve representing the measured quantity. The scalars ϵ_i are assumed to be random errors satisfying

$$E[\epsilon_i] = 0 \tag{6}$$

$$E[\epsilon_i \epsilon_j] = 0 \quad \text{for} \quad i \neq j \tag{7}$$

$$E[\epsilon_i \epsilon_j] = \sigma^2 \quad \text{for} \quad i = j \tag{8}$$

where $E[\cdot]$ denotes mathematical expectation.

Reference 2 suggests two methods for selecting the smoothing parameter depending upon whether or not the amount of noise in the data is known beforehand. If an estimate of σ^2 is given, the smoothing parameter is taken as the value of μ that minimizes the function

$$\hat{R}(\mu) = (1/n)\sum_{i=1}^{n}(f(x_i) - z_i)^2 + (2\sigma^2/n)Tr\mathbf{A} - \sigma^2 \tag{9}$$

$\hat{R}$ is an unbiased estimate of the true mean square error [2]. When σ^2 is not known, the smoothing parameter is taken as the minimizer of the generalized cross validation function

$$V(\mu) = \frac{(1/n)\sum_{i=1}^{n}(f(x_i) - z_i)^2}{(1 - (1/n)Tr\mathbf{A})^2} \tag{10}$$

In Eqs. 9 and 10, $\mathbf{A}$ is the influence matrix and Tr denotes the trace. Determining the value of μ that minimizes $\hat{R}$ or V requires an iterative approach. The influence matrix is a function of the smoothing parameter, μ, and linearly transforms the data into the values of the spline at the knots

$$\begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix} \tag{11}$$

An efficient procedure for calculating $Tr(\mathbf{A}) = \sum_{i=1}^{n} a_{ii}$ is provided in the Appendix.

Two-Dimensional Smoothing

For the two-dimensional problem, the data, z_{ij}, is assumed to be given on a uniform rectangular grid (x_i, y_j); $i = 1, \ldots, n$, $j = 1, \ldots, \tilde{n}$. The grid spacing in the x-direction is given by $x_{i+1} = x_i + h$ while the spacing in the y-direction is $y_{j+1} = y_j + \tilde{h}$. Basis functions for the y-direction are defined similarly to those for the x-direction.

$$\begin{aligned}
\tilde{s}_1(y) &= 1 \\
\tilde{s}_2(y) &= (y - y_1) \\
\tilde{s}_3(y) &= (y - y_1)^2/2 \\
\tilde{s}_{j+3}(y) &= \begin{cases} 0 & \text{for } y < y_j \\ (y - y_j)^3/6 & \text{for } y_j \leq y < y_{j+1} \\ \tilde{h}^3/6 + \tilde{h}^2(y - y_{j+1})/2 + \tilde{h}(y - y_{j+1})^2/2 & \text{for } y \geq y_{j+1} \end{cases}
\end{aligned} \tag{12}$$

With reference to Eqs. 1 and 12, a bicubic spline function, $f(x, y)$, can be expressed as a combination of products of basis functions in the x and y-directions.

$$f(x,y) = \sum_{i=1}^{n+2} \sum_{j=1}^{\tilde{n}+2} c_{ij} s_i(x) \tilde{s}_j(y) \tag{13}$$

The bicubic spline and its derivatives $\partial^{(k+l)} f / (\partial x^k \partial y^l)$ $(k, l = 0, 1, 2)$ are continuous over the entire rectangular grid because of the continuity of the basis functions and their derivatives.

The two dimensional smoothing problem requires finding the bicubic spline minimizing the functional, ϕ_2, defined by

$$\begin{aligned}
\phi_2(f) &= \sum_{i=1}^{n} \sum_{j=1}^{\tilde{n}} (f(x_i, y_j) - z_{ij})^2 + \mu \sum_{j=1}^{\tilde{n}} \int_{x_1}^{x_n} \left(\frac{\partial^3 f}{\partial x^3} \right)^2 dx + \mu \sum_{i=1}^{n} \int_{y_1}^{y_{\tilde{n}}} \left(\frac{\partial^3 f}{\partial y^3} \right)^2 dy \\
&\quad + \mu^2 \int_{x_1}^{x_n} \int_{y_1}^{y_{\tilde{n}}} \left(\frac{\partial^6 f}{\partial x^3 \partial y^3} \right)^2 dx\, dy
\end{aligned} \tag{14}$$

Because of the special structure of Eq. 14, the solution to the one-dimensional problem can be directly applied to the two-dimensional case. This extension was deduced by examining the solution scheme given in Reference 3. The coefficients, c_{ij}, in Eq. 13 which result in the minimization of ϕ_2 can be obtained through a two-step process. This process is outlined below and illustrated in Figure 3.

1. Pick a value for the smoothing parameter μ (see Eq. 14).
2. Smooth the $\tilde{n}$ rows of data one at a time using the one-dimensional solution scheme (see Appendix). In essence, this amounts to transforming each row of data, z_{ij} $(i = 1, \ldots, n)$, into the constants $\hat{c}_{ij}$ $(i = 1, \ldots, n+2)$. The transformation referred to is the one depicted in Figure 2.

3. Using the results of Step 2, smooth each column of $\hat{c}_{ij}$ using the one-dimensional solution scheme with $h = \tilde{h}$. This amounts to transforming each column of $\hat{c}_{ij}$ $(j = 1, \ldots, \tilde{n})$, into the constants c_{ij} $(j = 1, \ldots, \tilde{n} + 2)$.

Steps 2 and 3 above may be reversed without affecting the solution. That is, the columns of data can be smoothed first and then the rows. The number of calculations required in Steps 2 and 3 is proportional to $n\tilde{n}$. The two-dimensional counterparts of Eqs. (9) and (10) used in smoothing parameter selection are given by

$$\hat{R}(\mu) = \frac{1}{n\tilde{n}} \sum_{i=1}^{n} \sum_{j=1}^{\tilde{n}} (f(x_i, y_j) - z_{ij})^2 + \frac{2\sigma^2}{n\tilde{n}} Tr\mathbf{A}_2 - \sigma^2 \tag{15}$$

$$V(\mu) = \frac{\frac{1}{n\tilde{n}} \sum_{i=1}^{n} \sum_{j=1}^{\tilde{n}} (f(x_i, y_j) - z_{ij})^2}{(1 - \frac{1}{n\tilde{n}} Tr\mathbf{A}_2)^2} \tag{16}$$

where $\mathbf{A}_2$ is the influence matrix for the two-dimensional problem. Letting $\mathbf{A}$ and $\tilde{\mathbf{A}}$ denote the influence matrices for smoothing in the x and y-directions, respectively, it can be shown that the trace of the influence matrix for the two-dimensional problem is given by the product of the traces of $\mathbf{A}$ and $\tilde{\mathbf{A}}$.

$$Tr(\mathbf{A}_2) = Tr(\mathbf{A}) Tr(\tilde{\mathbf{A}}) \tag{17}$$

Equation (17) is essential for smoothing parameter selection because both Eqs. (15) and (16) require $Tr(\mathbf{A}_2)$.

Numerical Examples

We now present some numerical examples of smoothing and differentiating data given on a rectangular grid. In all of the examples, data was simulated at the grid points by adding white noise, ϵ_{ij}, to a known function, $g(x, y)$, of the independent variables x and y.

$$z_{ij} = g(x_i, y_j) + \epsilon_{ij} \tag{18}$$

$$E[\epsilon_{ij}] = 0 \tag{19}$$

$$E[\epsilon_{ij}\epsilon_{kl}] = 0 \quad \text{for } i \neq k \text{ or } j \neq l \tag{20}$$

$$E[\epsilon_{ij}\epsilon_{kl}] = \sigma^2 \quad \text{for } i = k \text{ and } j = l \tag{21}$$

The true mean square error, $R(\mu)$, is defined to be the average squared error between the bicubic spline, $f(x,y)$, and the noise-free data, $g(x,y)$.

$$R(\mu) = \frac{1}{n\tilde{n}} \sum_{i=1}^{n} \sum_{j=1}^{\tilde{n}} (f(x_i, y_j) - g(x_i, y_j))^2 \tag{22}$$

For smoothing data, μ should be chosen so that R is minimized. The true mean square error can be calculated directly for simulated data; however, this is not possible for most applications because the noise-free data is not known. For the purposes of this study, the inefficiencies $I_{\hat{R}}$ and I_V are defined as a measure of how well the smoothing parameter is chosen.

$$I_{\hat{R}} = \frac{R(\mu_{\hat{R}})}{R(\mu_R)} \tag{23}$$

$$I_V = \frac{R(\mu_V)}{R(\mu_R)} \tag{24}$$

where $\mu_{\hat{R}}$, μ_V, and μ_R are the minimizers of $\hat{R}$, V, and R, respectively (see Eqs. 15, 16, and 22). Inefficiencies close to unity indicate near optimal selection of the smoothing parameter.

The first example considers the set of data

$$x_i = 0.2(i-1) \qquad i = 1, \ldots, 51 \tag{25}$$

$$y_j = 0.2(j-1) \qquad j = 1, \ldots, 51 \tag{26}$$

$$g(x,y) = \sin\left(\frac{\pi x}{10}\right) \sin\left(\frac{\pi y}{10}\right) \tag{27}$$

White noise with σ=0.01 and σ=0.05 was added to $g(x,y)$ per Eq. 18 to simulate real data. The inefficiencies $I_{\hat{R}}$ and I_V are listed in Table 1 for five different sets of noise added to the data. The inefficiencies all have values near unity, indicating that the smoothing parameter was selected in a near optimal manner. The most encouraging result was that reasonable values for μ were chosen even when it was assumed that σ was not known.

The functions $f(x,y)$, $g(x,y)$, and the noisy data along the row defined by y=3 are plotted in Figure 4 for one of the data sets in which σ=0.05. The smoothing effect of the bicubic spline is evident from the figure.

The second example is taken from Reference 4.

$$x_i = 0.3(i-6) \qquad i = 1, \ldots, 11 \tag{28}$$

$$y_j = 0.3(j-6) \qquad j = 1, \ldots, 11 \tag{29}$$

$$g(x,y) = \exp(-x^2 - y^2) \tag{30}$$

Again, white noise with σ=0.01 and σ=0.05 was added to $g(x, y)$ to simulate real data. Figure 5 illustrates the variation of the functions R, $\hat{R}$, and V with the smoothing parameter, μ, for one of the data sets where σ=0.01. Notice that the minimizing values of μ for all three functions are near each other. The inefficiencies $I_{\hat{R}}$ and I_V for this example are listed in Table 2.

Two comments should be made regarding the effects of the number of grid lines, n and $\tilde{n}$, on the results. Comparing the values of the true mean square errors in Tables 1 and 2, it is clear that the spline approximation of the data in example 1 is better than that for example 2. When the number of grid lines in example 1 was reduced to $n = \tilde{n} = 11$, values for the true mean square error were comparable to those for example 2. In general, it has been observed that by increasing the number grid lines, and hence data points, better approximations to the data are possible. A second observation is that increasing the refinement of the grid causes the generalized cross validation function to become less sensitive to changes in the smoothing parameter, resulting in a less clearly defined minimum.

We conclude this section with some examples of numerical differentiation of the data given in the first example. Because the function used to generate the data was known, it was possible to make comparisons between the known derivatives and those estimated with the bicubic spline. In the examples, σ=0.01 and the smoothing parameter was chosen by minimizing the generalized cross validation function.

Comparisons between exact and estimated derivatives are shown in Figures 6 through 8 along the row defined by y=3. Figures 6 and 7 show comparisons for the first derivatives $\partial f/\partial x$ and $\partial f/\partial y$ and indicate excellent agreement. A comparison of the second derivative $\partial^2 f/(\partial x \partial y)$ is given in Figure 8 and shows good overall agreement. Higher order derivatives are typically more difficult to estimate than lower order ones and the errors are usually most pronounced at the endpoints.

Conclusions

A previously developed smoothing algorithm for one-dimensional problems is extended to handle situations where data is given on a rectangular grid. The method involves smoothing the data in the two different grid directions using the one-dimensional algorithm to obtain a bicubic spline approximation of the data. In addition to smoothing, the bicubic spline also provides estimates for the first two derivatives of the data.

Selection of the smoothing parameter, μ, is automated by finding the minimizer of one of two functions. When the amount of noise in the data is unknown, the value of μ minimizing the generalized cross validation function is used. Minimization of another function, Eq. 15, is used for selecting μ if there is prior knowledge of the noise levels. Both techniques are shown to select near optimal values for the smoothing parameter in the examples. Good estimates for the derivatives of simulated data are also obtained.

Because of the special structure of the rectangular grid and the smoothing functional, the two-dimensional problem lends itself to an efficient method of solution. The number of computations required for an n by $\tilde{n}$ grid of data is proportional to $n\tilde{n}$. Although efficient, the method is restricted to situations where data is given on a structured rectangular grid.

References

1. Dohrmann, C.R. and Busby, H.R., "Algorithms for Smoothing Noisy Data With Spline Functions and Smoothing Parameter Selection," VI International Congress on Experimental Mechanics, June 5-10, 1988, Portland, Oregon.

2. Craven, P. and Wahba, G., "Smoothing Noisy Data With Spline Functions," Numer. Math., Vol. 31, 1979, pp. 377-403.

3. Chiu, L.H. and Schumaker, L.L., "Complete Spline Smoothing," Numer. Math., Vol. 49, 1986, pp. 1-10.

4. Dierckx, P., "A Fast Algorithm for Smoothing Data on a Rectangular Grid While Using Spline Functions," SIAM J. Numer. Anal., Vol. 19, No. 6, 1982, pp. 1286-1304.

Appendix

This section summarizes the steps required for smoothing uniformly spaced data and for calculating the trace of the influence matrix. A more detailed description of the method can be found in Reference 1.

Smoothing is accomplished by a backward and forward sweep through the data. The steps are summarized below:

1. Begin the backward sweep by calculating the matrix $\mathbf{R}_n$ and vector $\mathbf{s}_n$ from Eqs. (A1-A3).

2. Calculate the matrices $\mathbf{R}_{n-1}, \ldots, \mathbf{R}_1$ and the vectors $\mathbf{s}_{n-1}, \ldots, \mathbf{s}_1$ using the recursive equations in (A5), storing the scalars d_k and $\mathbf{p}^T\mathbf{s}_k$ and the vectors $\mathbf{h}_k^T\mathrm{M}$ for $k = n, \ldots, 2$.

3. Begin the forward sweep by calculating the starting vector $\mathbf{f}_1$ from Eq. (A6).

4. Sweep forward calculating the scalars $g_1, \ldots, g_{n-1}$ and vectors $\mathbf{f}_2, \ldots, \mathbf{f}_n$ using the equations in (A7).

5. The coefficients c_i appearing in Eq. 1 are given by the equations in (A8).

The following equations are used in the smoothing procedure ($\overline{\mu} = h^{-5}\mu$ is the normalized value of the smoothing parameter and **I** is the identity matrix).

$$\mathbf{f}_i = \begin{pmatrix} f(x_i) \\ h\dfrac{df(x_i)}{dx} \\ h^2\dfrac{d^2f(x_i)}{dx^2} \end{pmatrix} \qquad \mathbf{z}_i = \begin{pmatrix} z_i \\ 0 \\ 0 \end{pmatrix} \tag{A1}$$

$$\mathbf{M} = \begin{pmatrix} 1 & 1 & 1/2 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{L} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad \mathbf{p} = \begin{pmatrix} 1/6 \\ 1/2 \\ 1 \end{pmatrix} \tag{A2}$$

$$\mathbf{R}_n = \mathbf{L} \qquad \mathbf{s}_n = -2\mathbf{L}\mathbf{z}_n \tag{A3}$$

$$d_{i+1} = 1/(2\overline{\mu} + 2\mathbf{p}^T\mathbf{R}_{i+1}\mathbf{p}) \qquad \mathbf{h}_{i+1} = 2\mathbf{R}_{i+1}\mathbf{p} \tag{A4}$$

$$\mathbf{R}_i = \mathbf{L} + \mathbf{M}^T(\mathbf{R}_{i+1} - (d_{i+1}/2)\mathbf{h}_{i+1}\mathbf{h}_{i+1}^T)\mathbf{M} \qquad \mathbf{s}_i = -2\mathbf{L}\mathbf{z}_i + \mathbf{M}^T(\mathbf{I} - d_{i+1}\mathbf{h}_{i+1}\mathbf{p}^T)\mathbf{s}_{i+1} \tag{A5}$$

$$\mathbf{f}_1 = -(1/2)\mathbf{R}_1^{-1}\mathbf{s}_1 \tag{A6}$$

$$g_i = -d_{i+1}[\mathbf{p}^T\mathbf{s}_{i+1} + \mathbf{h}_{i+1}^T\mathbf{M}\mathbf{f}_i] \qquad \mathbf{f}_{i+1} = \mathbf{M}\mathbf{f}_i + \mathbf{p}g_i \tag{A7}$$

$$c_1 = \mathbf{f}_1(1) \qquad c_2 = \mathbf{f}_1(2)/h \qquad c_3 = \mathbf{f}_1(3)/h^2 \qquad c_{i+3} = g_i/h^3 \quad \text{for } i = 1, \ldots, n-1 \tag{A8}$$

When interpolating data (μ=0), it is recommended that $\overline{\mu}$ be given a value of 1 in the calculation of d_n and d_2 to avoid numerical problems. The quantities $\mathbf{R}_i$, $\mathbf{h}_i$, and d_i need only be calculated for one row, or column, of the grid because they do not depend upon the data.

Calculating the trace of the influence matrix, **A**, is also accomplished in a recursive manner:

1. Calculate the matrices $\mathbf{X}_1$ through $\mathbf{X}_n$ using Eqs. (A9) and (A10).

2. Calculate the trace of the influence matrix, **A**, from Eq. (A11).

$$\hat{\mathbf{M}}_i = \mathbf{M} - d_i \mathbf{p} \mathbf{h}_i^T \mathbf{M} \qquad \mathbf{E}_i = -d_i \mathbf{p} \mathbf{p}^T \tag{A9}$$

$$\mathbf{X}_1 = \mathbf{R}_1^{-1} \qquad \mathbf{X}_i = \hat{\mathbf{M}}_i \mathbf{X}_{i-1} \hat{\mathbf{M}}_i^T - 2\mathbf{E}_i \tag{A10}$$

$$Tr(\mathbf{A}) = \sum_{i=1}^{n} \mathbf{X}_i(1,1) \tag{A11}$$

Table 1. Inefficiencies associated with the two different methods for selecting the smoothing parameter in Example 1.

Noise Set	σ	$R(\mu_R)$	$R(\mu_{\hat{R}})$	$R(\mu_V)$	$I_{\hat{R}}$	I_V	Noise Set	σ	$R(\mu_R)$	$R(\mu_{\hat{R}})$	$R(\mu_V)$	$I_{\hat{R}}$	I_V
1	0.01	2.47e-6	2.50e-6	2.50e-6	1.01	1.01	6	0.05	3.35e-5	3.54e-5	3.59e-5	1.06	1.07
2	0.01	2.41e-6	2.42e-6	2.42e-6	1.00	1.00	7	0.05	3.38e-5	3.44e-5	3.45e-5	1.02	1.02
3	0.01	2.66e-6	2.67e-6	2.66e-6	1.00	1.00	8	0.05	3.71e-5	4.12e-5	4.12e-5	1.11	1.11
4	0.01	2.55e-6	2.63e-6	2.69e-6	1.03	1.05	9	0.05	4.52e-5	4.52e-5	4.52e-5	1.00	1.00
5	0.01	2.46e-6	2.46e-6	2.47e-6	1.00	1.00	10	0.05	3.46e-5	3.48e-5	3.47e-5	1.01	1.00

Table 2. Inefficiencies associated with the two different methods for selecting the smoothing parameter in Example 2.

Noise Set	σ	$R(\mu_R)$	$R(\mu_{\hat{R}})$	$R(\mu_V)$	$I_{\hat{R}}$	I_V	Noise Set	σ	$R(\mu_R)$	$R(\mu_{\hat{R}})$	$R(\mu_V)$	$I_{\hat{R}}$	I_V
11	0.01	2.94e-5	3.04e-5	3.36e-5	1.03	1.14	16	0.05	6.96e-4	7.04e-4	6.98e-4	1.01	1.00
12	0.01	2.39e-5	2.42e-5	2.49e-5	1.01	1.04	17	0.05	6.93e-4	6.93e-4	7.05e-4	1.00	1.02
13	0.01	3.28e-5	3.31e-5	3.28e-5	1.01	1.00	18	0.05	6.03e-4	6.18e-4	6.06e-4	1.02	1.01
14	0.01	2.12e-5	2.15e-5	2.15e-5	1.01	1.01	19	0.05	6.84e-4	7.13e-4	7.19e-4	1.04	1.05
15	0.01	4.39e-5	4.40e-5	4.39e-5	1.00	1.00	20	0.05	4.87e-4	4.87e-4	4.88e-4	1.00	1.00

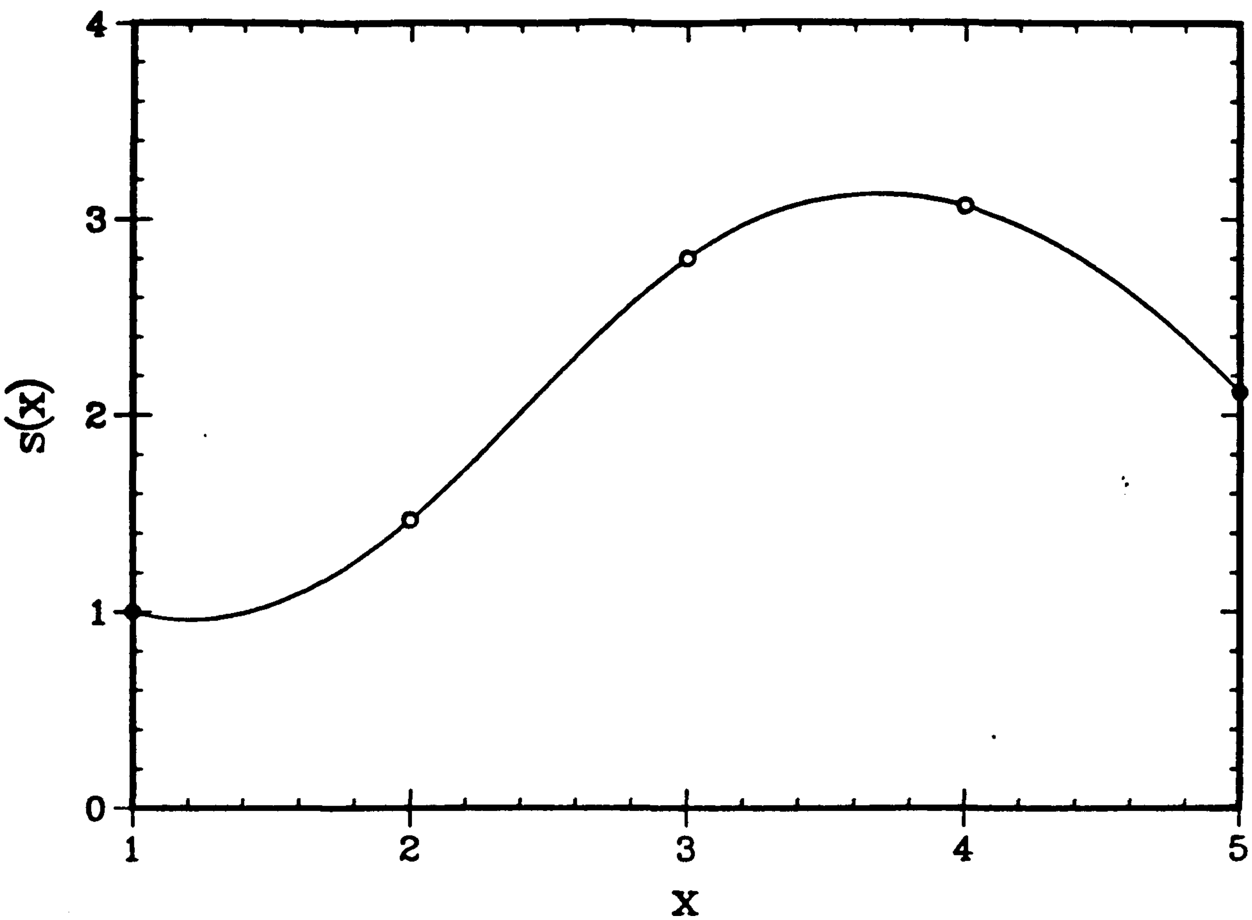

Figure 1. Example of a cubic spline function defined for the knots $x_i = i$ $(i = 1, \ldots 5)$. The plot was generated using Eq. 1 with $n = 5$, $c_1 = 1$, $c_2 = -0.4$, $c_3 = 2$, $c_4 = -0.8$, $c_5 = -2.8$, $c_6 = 0.4$ and $c_7 = 0.3$.

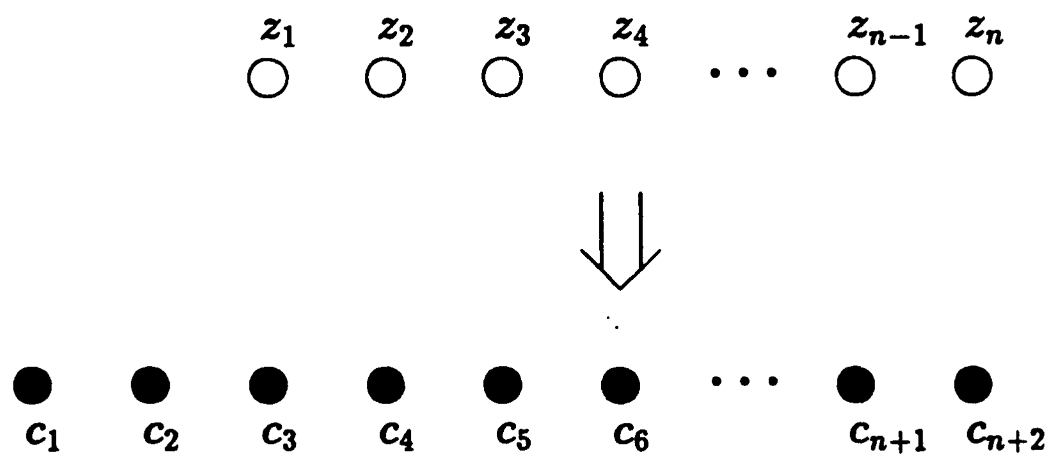

Figure 2. Illustration of one-dimensional smoothing. The data z_i is transformed into the coefficients c_i (see Eqs. 1 and 3).

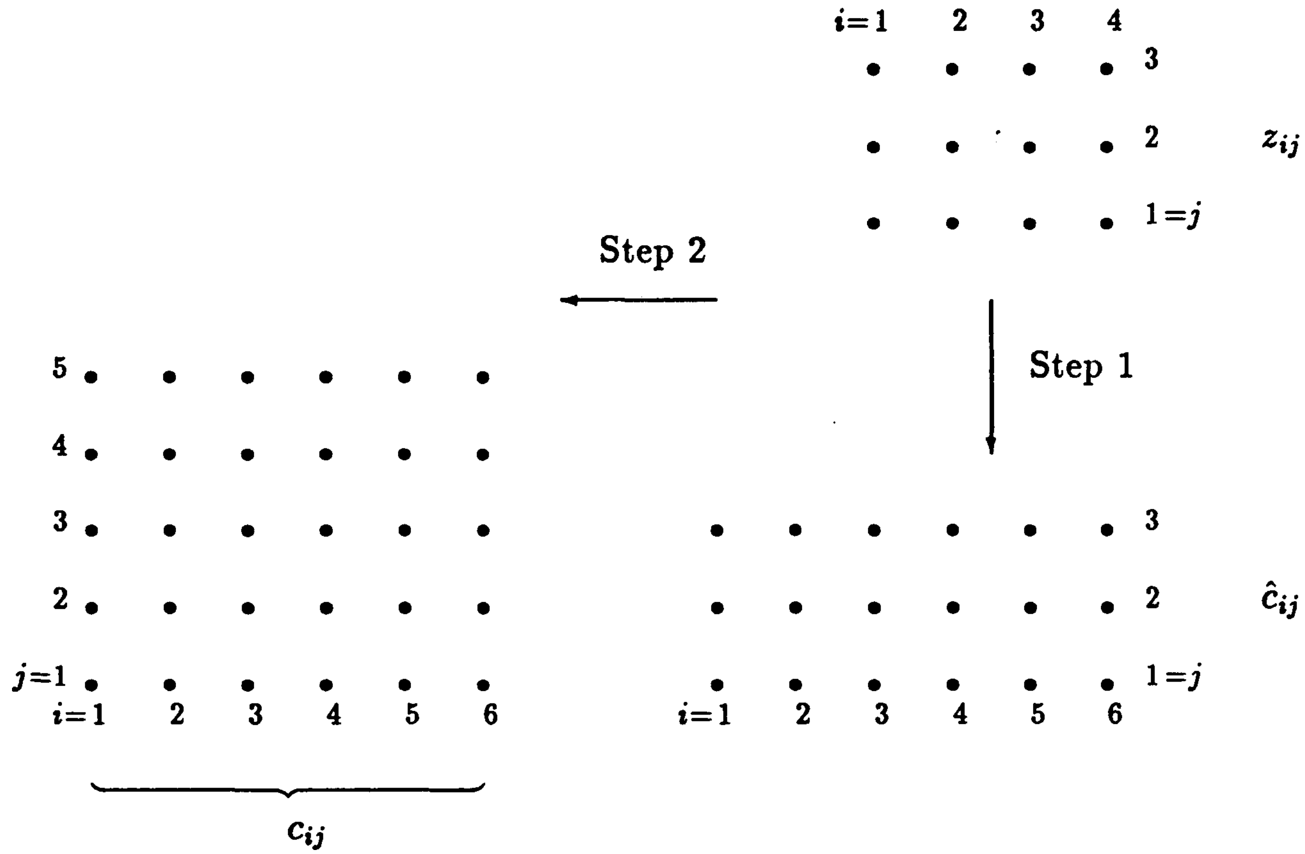

Figure 3. Illustration of the two-dimensional smoothing process for n=4 and $\tilde{n}$=3. The first step transforms each row of data z_{ij} into the constants $\hat{c}_{ij}$ $(i = 1, \ldots, n+2,\ j = 1, \ldots, \tilde{n})$. The second step transforms each column of $\hat{c}_{ij}$ into the constants c_{ij} $(i = 1, \ldots, n+2,\ j = 1, \ldots, \tilde{n}+2)$. Both transformations are based upon the one used for the one-dimensional problem (see Fig. 2). The net result is a transformation of the data, z_{ij}, into the coefficients, c_{ij}, of the bicubic spline function (see Eq. 13).

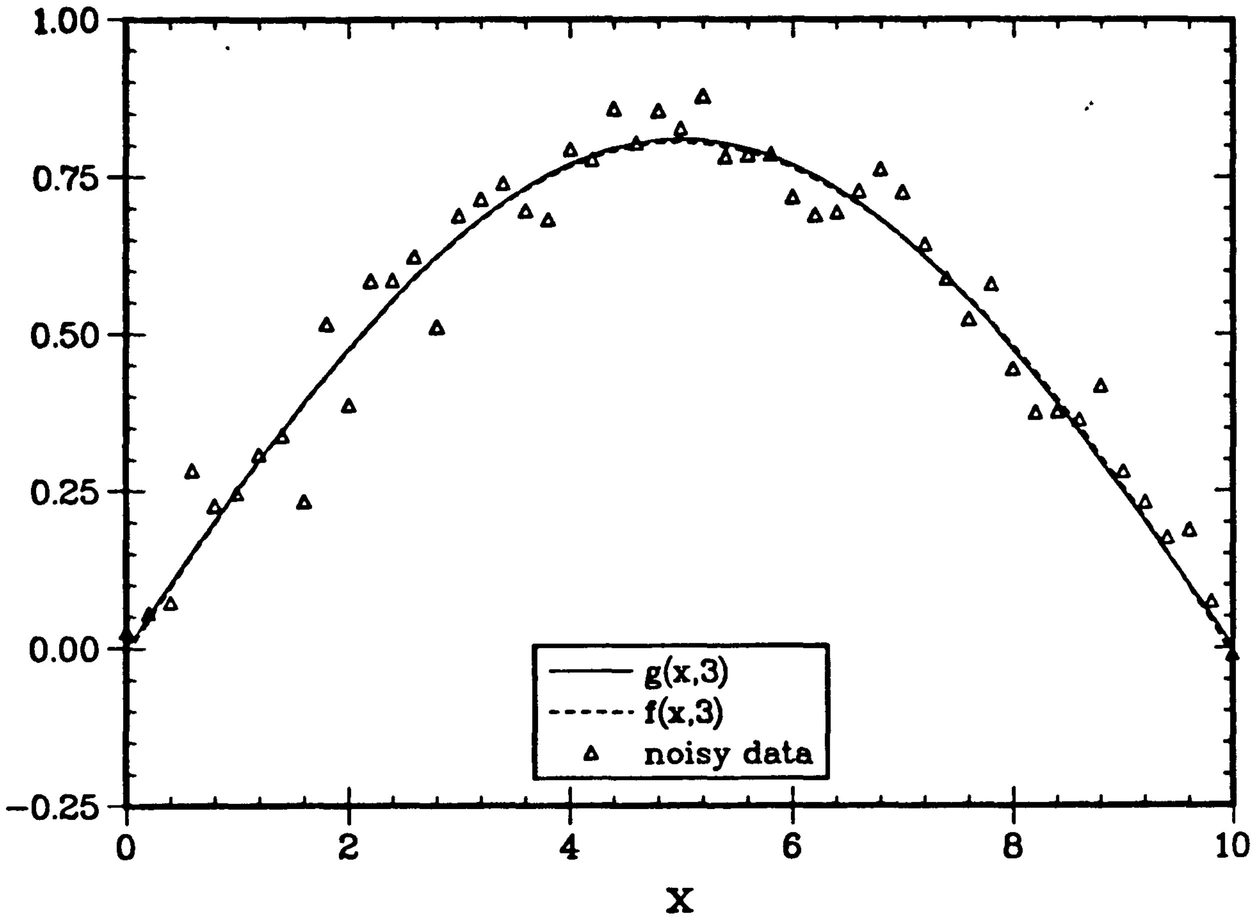

Figure 4. Plot of the noise-free function, g, the bicubic spline, f, and the noisy data along a row defined by $y = 3$ for example 1 ($\sigma = 0.05$).

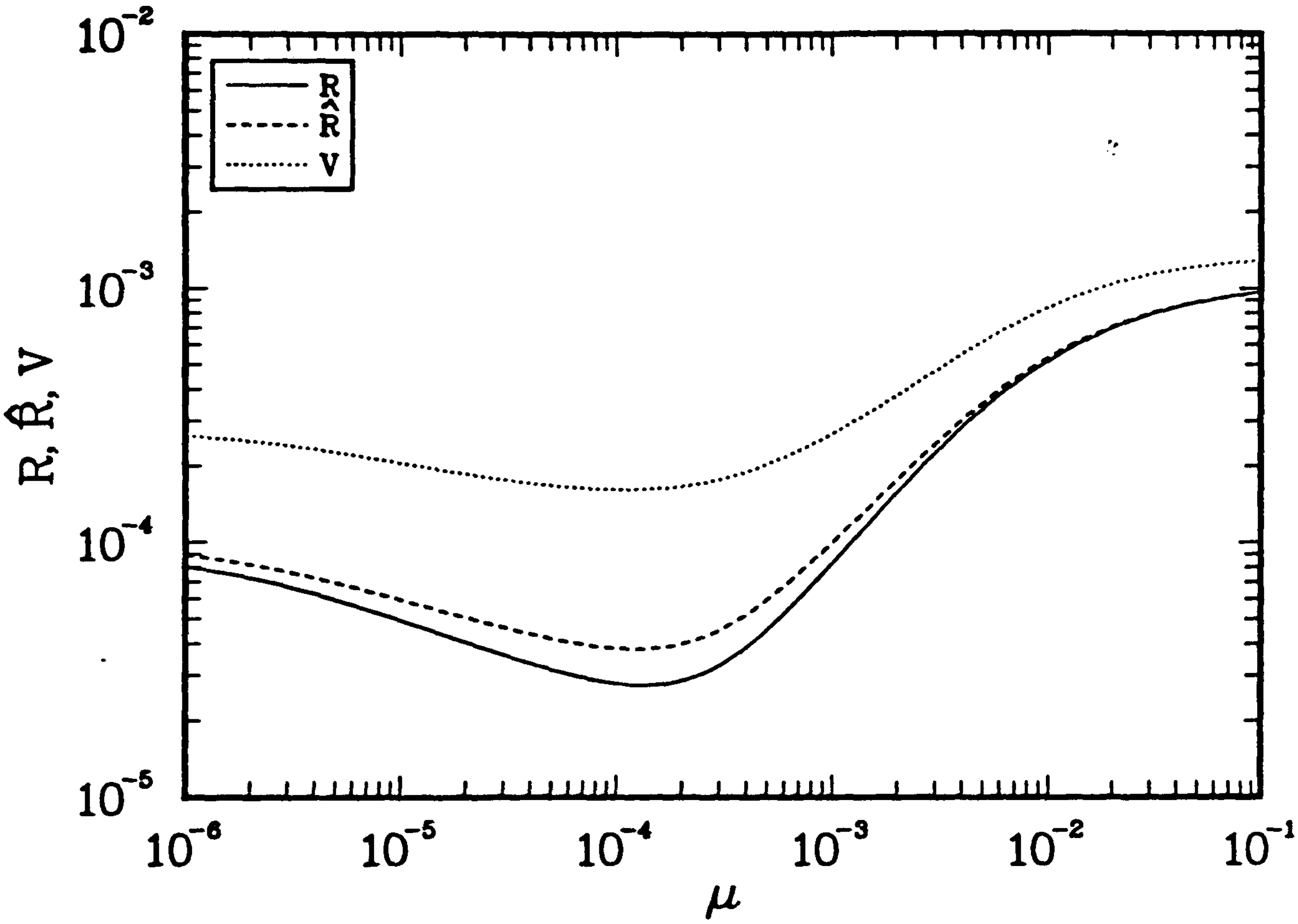

Figure 5. Variation of the functions R, $\hat{R}$, and V with the smoothing parameter, μ, for example 2 ($\sigma = 0.01$).

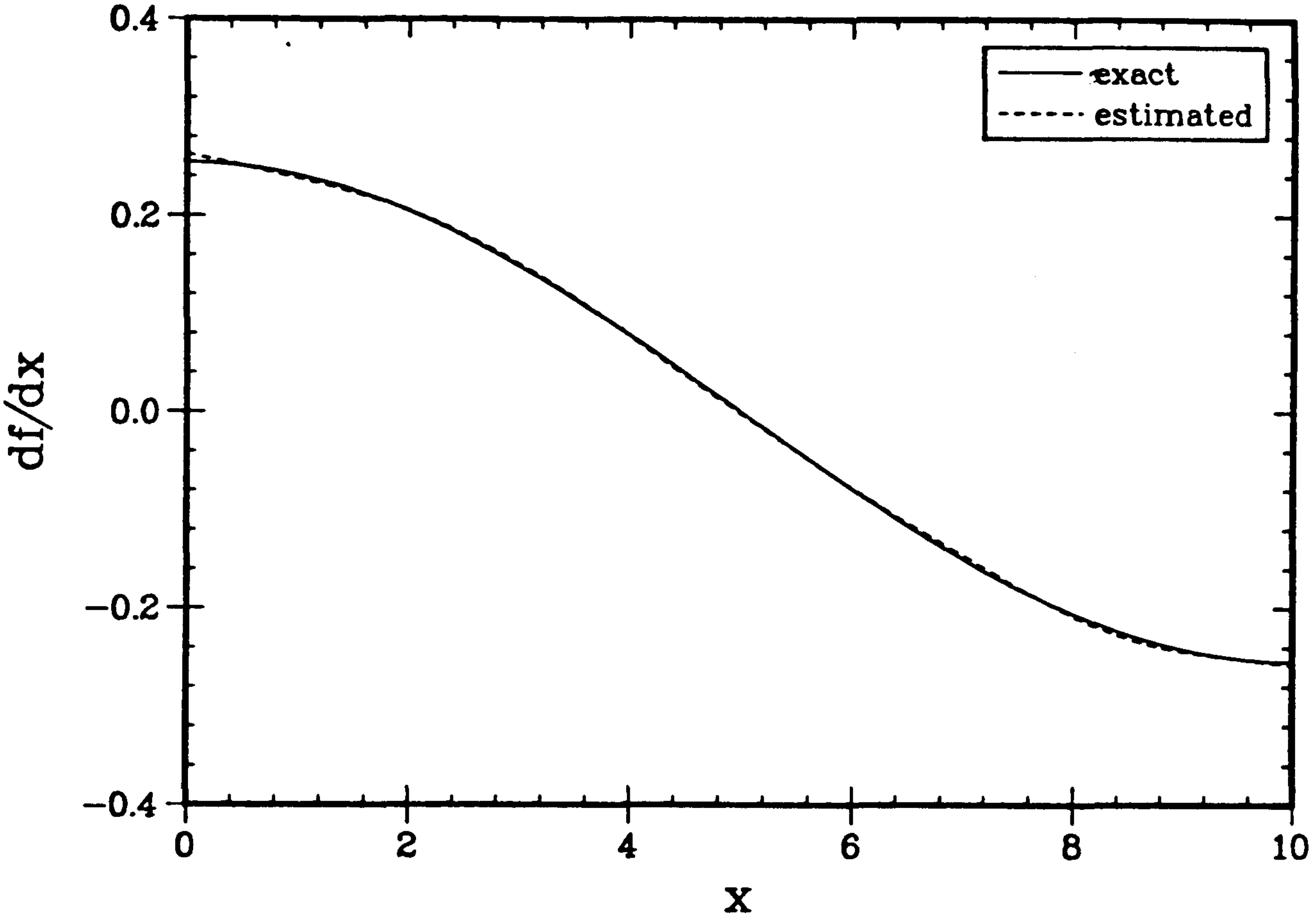

Figure 6. Comparison of the estimated and exact derivative $\partial f/\partial x$ along the row defined by y=3 for example 1 ($\sigma = 0.01$).

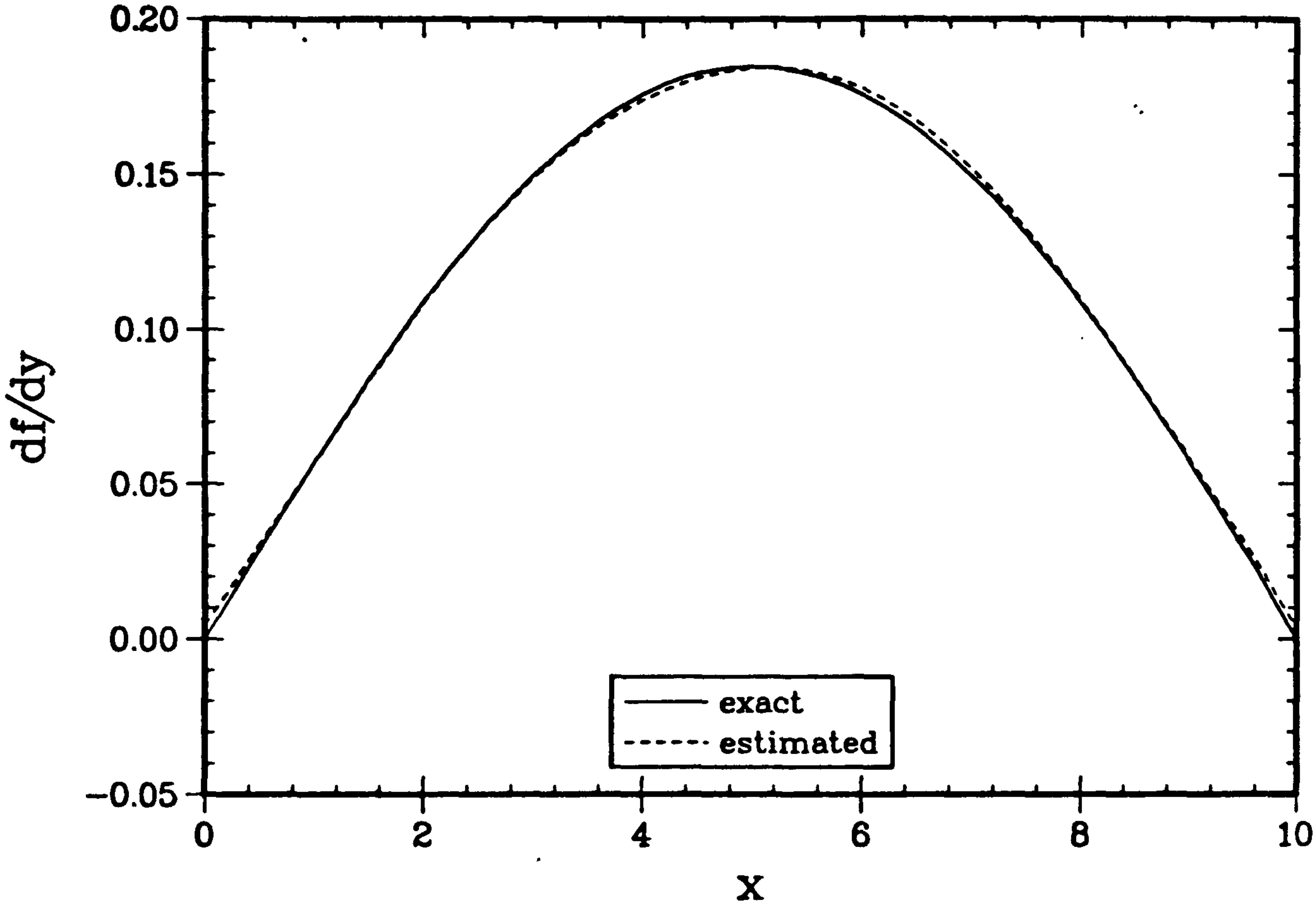

Figure 7. Comparison of the estimated and exact derivative $\partial f/\partial y$ along the row defined by y=3 for example 1 ($\sigma = 0.01$).

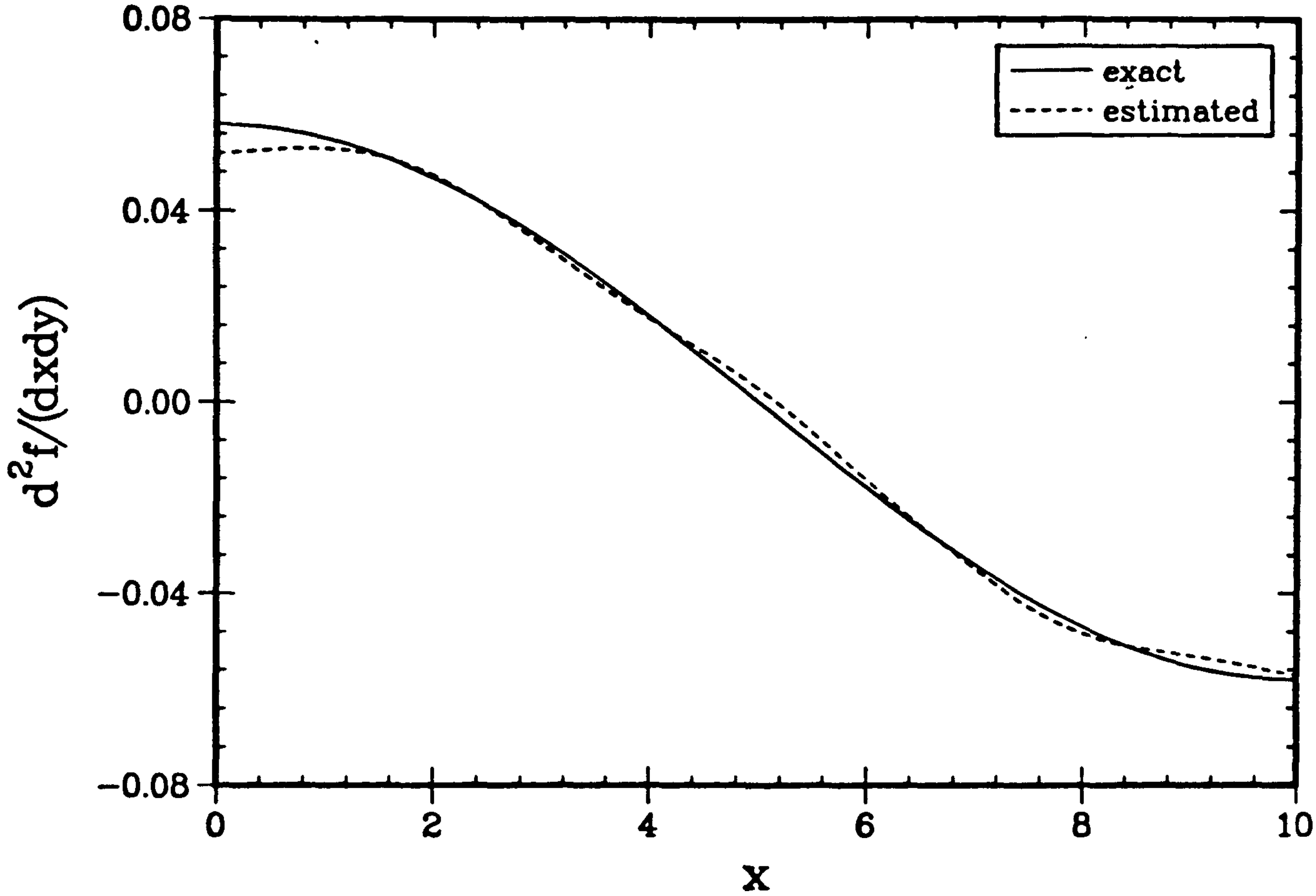

Figure 8. Comparison of the estimated and exact derivative $\partial^2 f/(\partial x \partial y)$ along the row defined by y=3 for example 1 ($\sigma = 0.01$).