The Π-Method for Estimating Multivariate Functions from Noisy Data

By

Leo Breiman*
University of California
Berkeley, California

Technical Report No. 231
December 1989
(revised July 1990)

*Partially supported by NSF Grant DMS-8718362

Department of Statistics
University of California
Berkeley, California

The Π-Method for Estimating Multivariate Functions from Noisy Data

Leo Breiman

University of California

Berkeley, California

ABSTRACT

Using noisy data, the Π-method for estimating an underlying smooth function of M variables, $(x_1, \ldots, x_M)$, is based on approximating it by a sum of products of the form $\Pi_m \phi_m(x_m)$. The problem is then reduced to estimating the univariate functions in the products. A convergent algorithm is described. The method keeps tight control on the degrees of freedom used in the fit. Many examples are given. The quality of fit given by the Π-method is excellent. Usually, only a small number of products are enough to fit even fairly complicated functions. The coding into products of univariate functions allows a relatively understandable interpretation of the multivariate fit.

Key words: nonparametric regression, regression splines, knot deletion, function estimation.

The Π-Method for Estimating Multivariate Functions from Noisy Data

1.0. Introduction.

Given data $\mathbf{x}_n = (x_{1n}, \ldots, x_{Mn})$, $n = 1, \ldots, N$ and values

$$y_n = f(\mathbf{x}_n) + \varepsilon_n$$

where $f(\cdot)$ is unknown but assumed "smooth" in $E^{(M)}$ (Euclidean M-space) and the $\{\varepsilon_n\}$ are "mean zero noise", estimating $f(\cdot)$ is a difficult problem.

In the one dimensional case, a number of satisfactory methods are available. These include smoothing splines, kernel estimates, moving linear smoothers, and dks curve fitting (see Breiman and Peters (1988), Breiman (1989)).

The situation in two or more dimensions is less satisfactory. Work is in progress on interaction splines (see Gu, et. al (1988)). Tensor products of splines have been proposed (see Schumaker (1976), (1984)). There is a new and promising approach by Friedman (1988). One early and ingenious method is projection pursuit (see Friedman and Stuetzle (1981)). The comments on projection pursuit in Friedman (1988) give some reasons that explain why it has not been more commonly used in applied work.

The difficulty in accurately estimating complex multivariate functions using sparse noisy data is summarized by the Bellman phrase "the curse of dimensionality". In one dimension, 8 spline functions are enough to approximate most reasonably smooth

functions. In two dimensions, if we use a tensor spline product with 8 functions per coordinate, then 64 coefficients have to be estimated. The complexity of multivariate surfaces can grow exponentially with dimension but usually our data does not.

Noisy sparse data does not lend itself to precise estimation. Properly employed, this is the statisticians defense against the ''curse of dimensionality''. For instance, the most compelling justification for linear regression is that the data set is too small or too noisy to resolve any nonlinearities. Once this is no longer true, the justification for a linear fit vanishes.

The method we investigate consists of approximating a function $f(\mathbf{x})$, $\mathbf{x} \in E^{(M)}$, by a sum of products:

$$\sum_{j=1}^{J} \prod_{m=1}^{M} \phi_{j,m}(x_m).$$

With noisy data, estimates require only a few products. Multivariate estimation is reduced to the estimation of the univariate functions $\{\phi_{j,m}(x_m)\}$. Furthermore, these univariate functions can be estimated by a simple iterative scheme.

This reduction to univariate function estimation disarms 'the curse of dimensionality''. For example, in high noise situations, one product is often an adequate fit to the surface. Then, the data is only supporting the estimation of a few univariate functions. This approach is particularly appealing when the x-variables are qualitatively different. If x_1 is yield in tons of potatoes per acre, and x_2 is annual rainfall, then efforts to treat

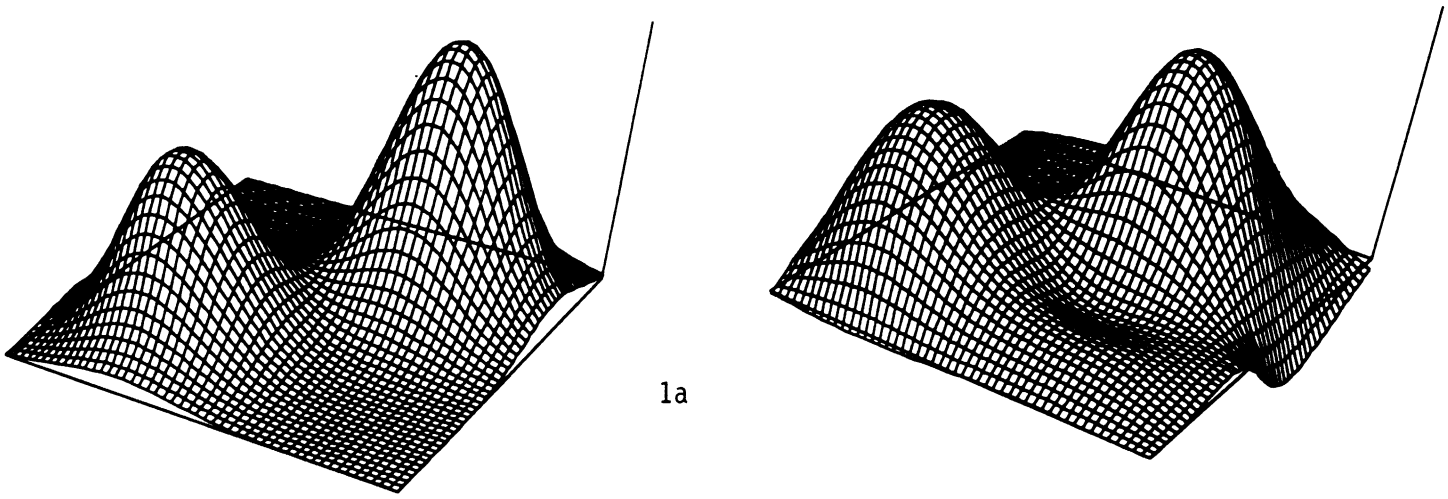
them as geometrically equivalent when normalized by some arbitrary measure of spread arouses a disquieting skepticism. The Π-method treats each variable in its own intrinsic domain.

To give an item of what the Π-method can produce, examine figure 1a-f. The left hand illustration in each figure is the underlying function, the right side illustration is the fit produced by the Π-method. In each case (x_1, x_2) was uniformly sampled 100 times on a square and normal noise added. The functions are in increasing order of complexity, and signal/noise ratio (standard deviation of the function divided by the standard deviation of the noise). The successive s/n ratios in 1a-1f are 1.0, 2.0, 2.0, 2.5, 3.0, 4.0. The equations of the functions and the sampled square are given in Appendix II.

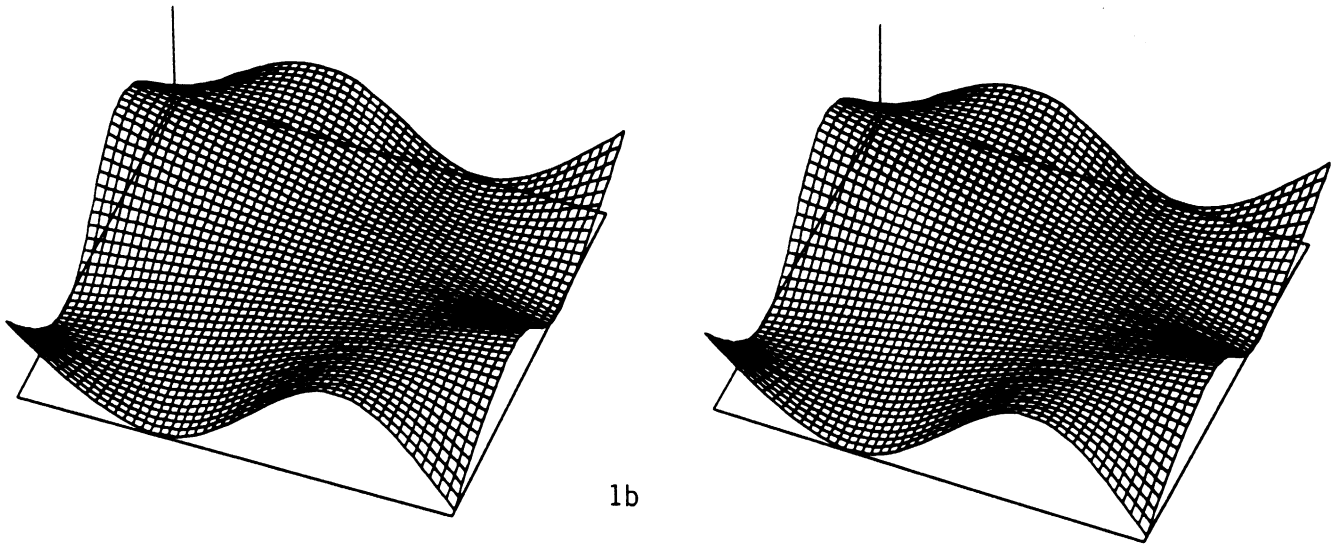
One hundred samples in two dimensions is about 10 samples per dimension. Still, as seen from the figures, the Π-method is able to faithfully reproduce the underlying function even in low signal/noise situations. In figure 1f we are reaching the limits of resolution. The underlying function is complex and 100 data points, even with high signal/noise are not enough.

Figure 2a-d is an example fitted both in Gu, et. al (1988) and Friedman (1988). Here (x_1, x_2) are sampled 300 times and s/n $\cong$ 3.0. Figure 2a is the original function, 2b is a reproduction of the interaction spline fit (Gu, et. al (1988)), 2c is the fit of

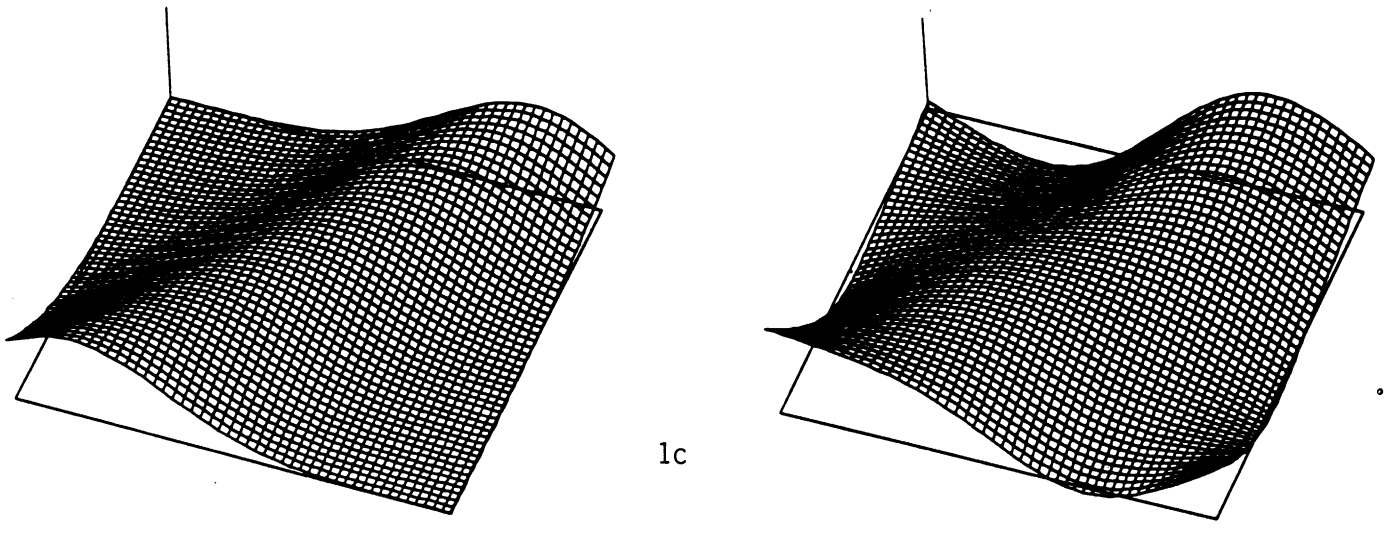
Figure 1



1a

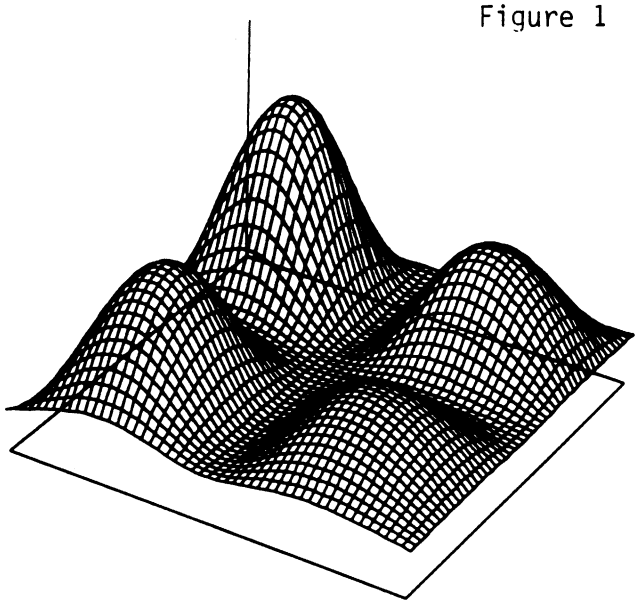


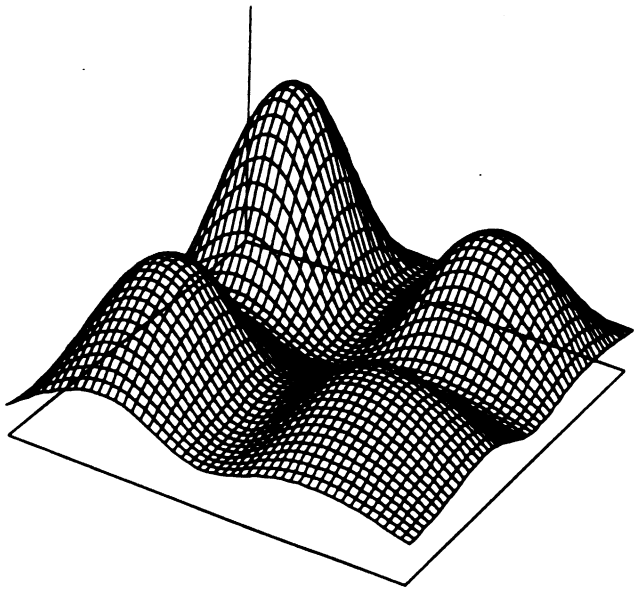
1b



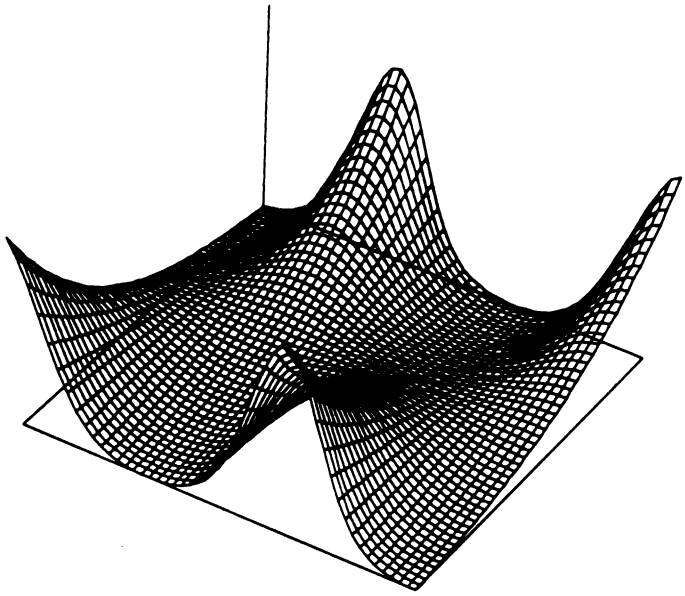
1c

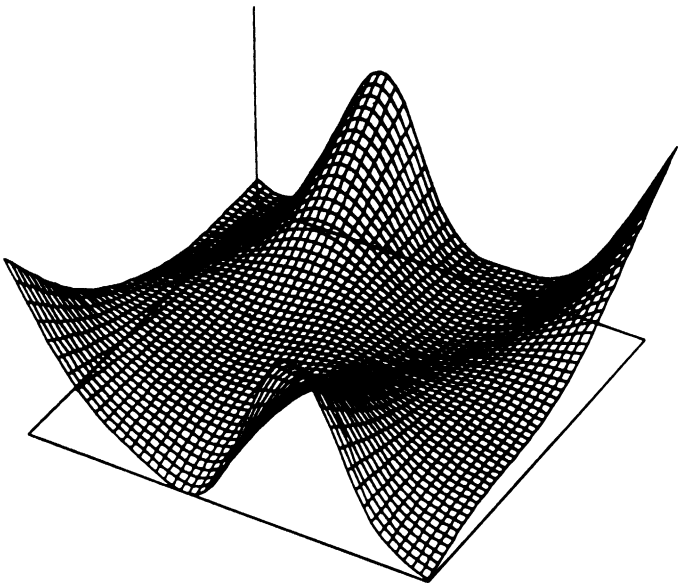
Figure 1 Continued



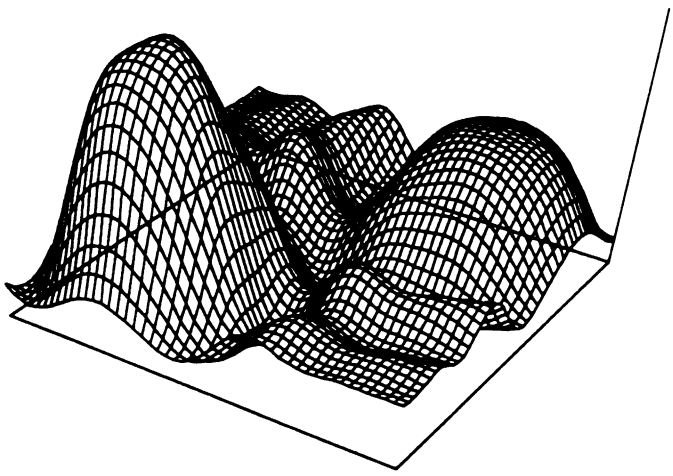


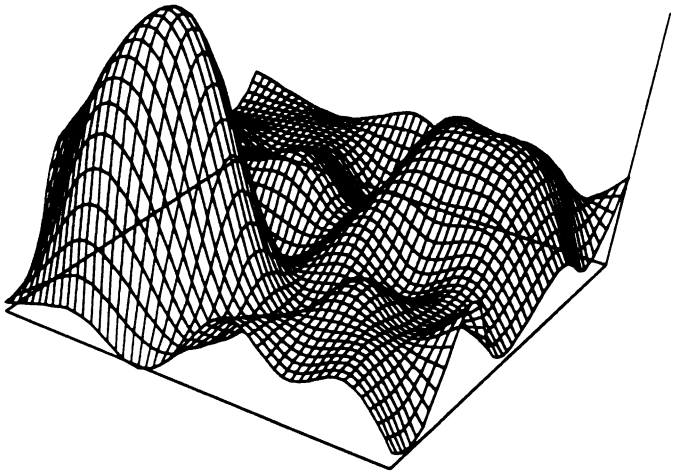
1d





1e





1f

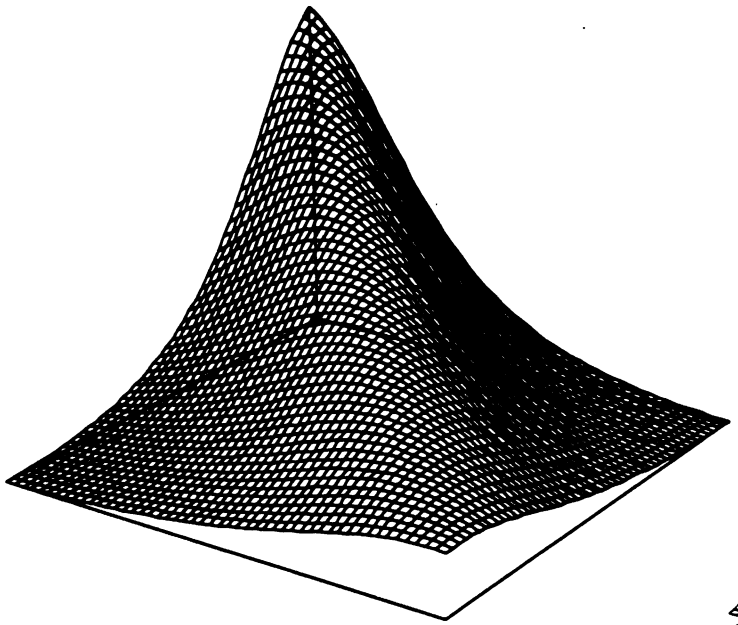
Friedman's (1988) program and 2d is the Π-method fit. The Π-method gives an MSE about 20% lower than the published value for the interaction spline fit. Friedman notes that his MSE is about the same as the interaction spline fit. But the random numbers are not the same, so the comparison is not definitive.

Noisy data, even in two dimensions where it can be visualized, may bear an imperfect relation to the underlying smooth surface. To illustrate this, figure 3 contains bar graphs of the noisy data for two of the previous examples. We leave it to the reader to sort out which is which.

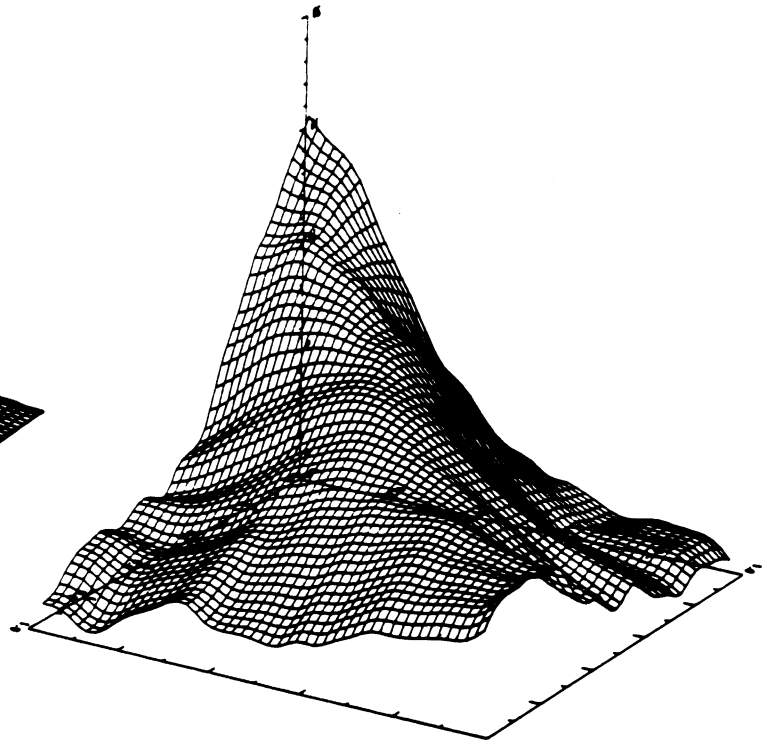
Another problem in estimating multivariate functions is how to understand the result. A 2-dimensional function can be plotted in 3-dimensional space and visually inspected. But understanding the shape of a function of 3 or more variables is not easy. The Π-method gives an efficient way of coding the information in a multivariate function estimate. For instance, if the estimate of a function of 3 variables is a single product $\prod_1^3 \phi_m(x_m)$, then all of the information about the estimate is contained in the 3 bivariate graphs of $\phi_m(x_m)$ v.s. x_m.

In describing the Π-method, we give the theoretical rationale in Section 2. It derives from a numerical analysis problem of which an important special case was solved in the early 1900's. The implementation is discussed in Section 3. The theoretical Π-method uses an iterative scheme to get the product functions. The data

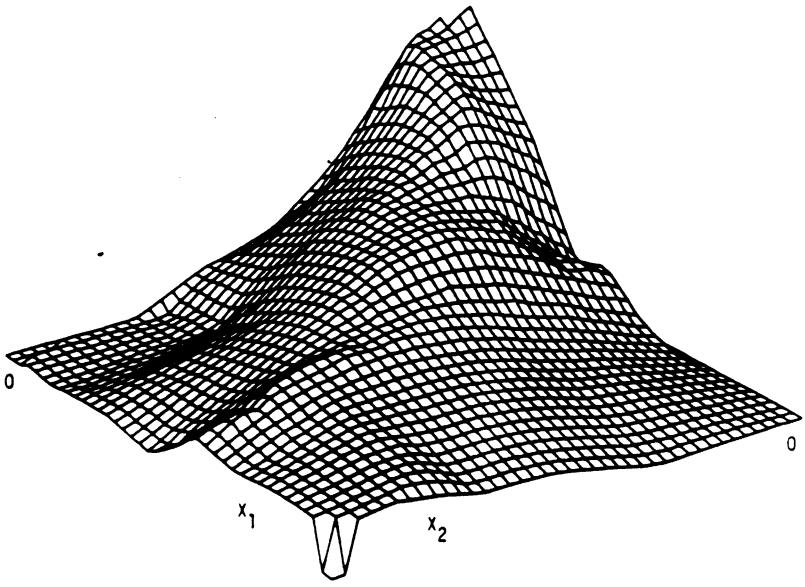
Figure 2



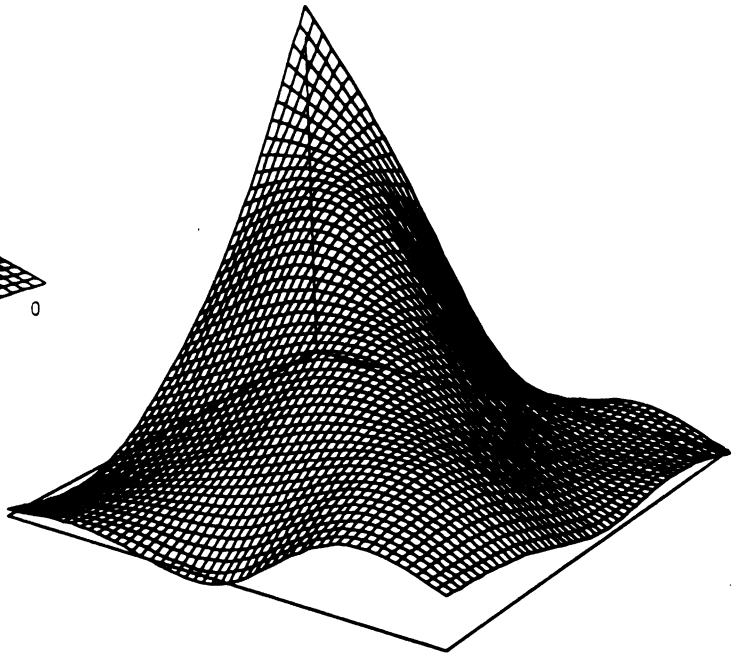
2a Original Function



2b Interaction Spline Fit



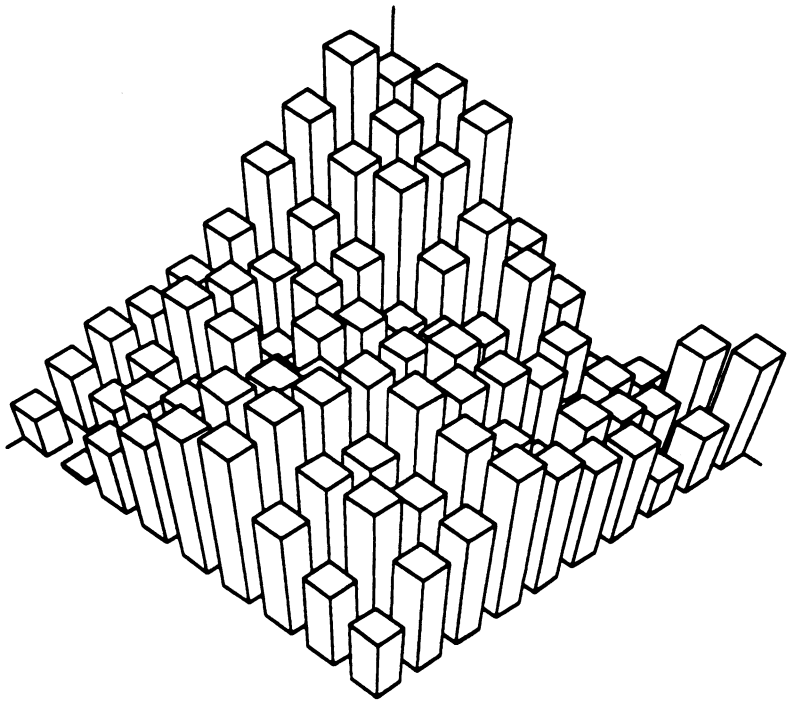
2c Friedman's "MARS" Fit

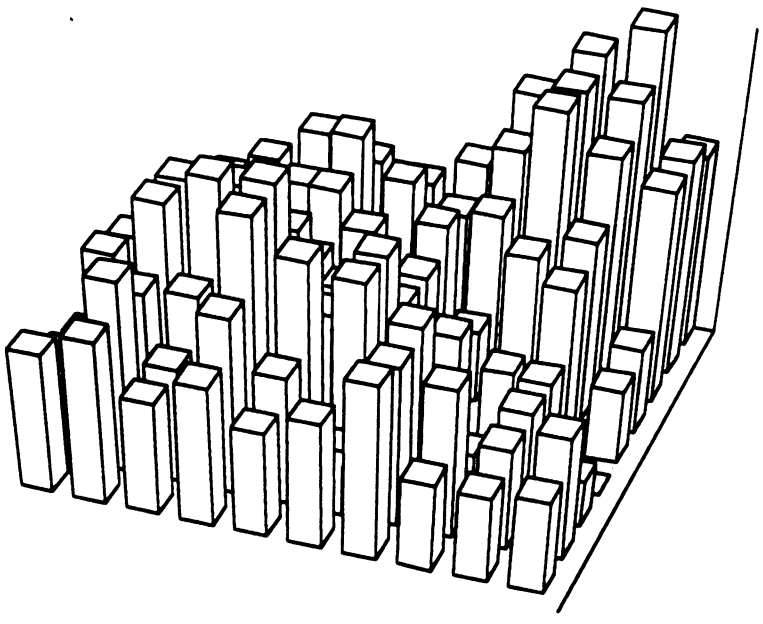


2d II - Method Fit

Figure 3

Noisy Data





implementation selected gives a convergent iteration scheme and also tightly controls the degrees of freedom used in the fit.

Section 4 gives further examples on both real and simulated data in 2 and 3 dimensions. Section 5 looks at the issue of centering the response variable and Section 6 gives a short summary. Appendix I describes our method for finding initial values for the iterative scheme.

2.0. **The Π-Method.**

2.1. **General Case.**

The idea of the Π-method is this: given random variables y, $\mathbf{x} = (x_1, \ldots, x_M)$ denote by Π_j products of the form

$$\Pi_j = \prod_{m=1}^{M} \phi_{jm}(x_m).$$

What we are aiming at is an approximation

$$E(y|\mathbf{x}) \sim \Pi_1 + \Pi_2 + \cdots + \Pi_J.$$

In particular, we want to find $\Pi_1, \ldots, \Pi_J$ to minimize

$$E[y - \sum_{1}^{J} \Pi_j]^2. \tag{2.1}$$

That this is an effective approximation method is partially conveyed by the following: write y as

$$y = E(y|\mathbf{x}) + z$$

where z is the noise component.

Proposition 2.2. *If* $Ey^2 < \infty$, *then for any* $\varepsilon > 0$, *there is a sum of products*

$$\Pi_1 + \cdots + \Pi_J$$

such that

$$E(y - \Pi_1 - \cdots - \Pi_J)^2 \leq Ez^2 + \varepsilon.$$

Proof. This follows from the fact that sums of products are dense in the class of squared-integrable functions on $E^{(M)}$.

What makes the Π-method workable is that, like the ACE algorithm (Breiman and Friedman (1985)), solutions can be gotten by iterated sequences of one-dimensional conditional expectations. To minimize $E[y - \Pi]^2$, make an initial guess $\Pi = \Pi_1^M \phi_m^{(0)}(x_m)$. Hold $\phi_2^{(0)}, \ldots, \phi_M^{(0)}$ constant and ask for that function $\phi_1(x_1)$ which minimizes

$$E[y - \phi_1(x_1)\Pi_2^M \phi_m^{(0)}(x_m)]^2.$$

The solution is clear;

$$\phi_1(x_1) = \frac{E[y\Pi_2^M \phi_m^{(0)}(x_m) \mid x_1]}{E[(\Pi_2^M \phi_m^{(0)}(x_m))^2 \mid x_1]}.$$

Call this $\phi_1^{(1)}(x_1)$. Now hold $\phi_1^{(1)}, \phi_3^{(0)}, \ldots, \phi_M^{(0)}$ constant and minimize over $\phi_2(x_2)$, etc. At each step $E(y - \Pi)^2$ is decreasing so convergence to a limit is provable by standard arguments under weak conditions. At the limit, the stationary equations hold:

$$\phi_m(x_m) = \frac{E[y\Pi_{m' \neq m} \phi_{m'}(x_{m'}) \mid x_m]}{E[(\Pi_{m' \neq m} \phi_{m'}(x_{m'}))^2 \mid x_m]}. \tag{2.3}$$

This can be generalized to the problem of minimizing $E(y - \Pi_1 - \cdots - \Pi_J)^2$. Let initial functions for Π_j, $j = 1, \ldots, J$ be $\phi_{jm}^{(0)}(x_m)$. Hold $\phi_{jm}^{(0)}$ constant, $m \geq 2$, $j = 1, \ldots, J$ and ask what functions $\phi_{j1}(x_1)$, $j = 1, \ldots, J$, minimize

$$E(y - \phi_{11}(x_1)\Pi_{m\geq 2}\phi_{1m}^{(0)}(x_m) - \cdots - \phi_{J1}(x_1)\prod_{m\geq 2}\phi_{Jm}^{(0)}(x_m))^2.$$

Form the matrix

$$W_{jj'}(x_1) = E[(\prod_{m\geq 2}\phi_{jm}^{(0)}(x_m))(\prod_{m\geq 2}\phi_{j'm}^{(0)}(x_m))\,|\,x_1]$$

and the vector variable

$$V_j(x_1) = E[y\prod_{m\geq 2}\phi_{jm}^{(0)}(x_m)\,|\,x_1].$$

Then

$$\phi_{.1}(x_1) = [W(x_1)]^{-1}V(x_1)$$

and we can go through the iteration cycle as before. The stationary equations are:

$$\sum_{j'}\phi_{j'm}(x_m)E(\prod_{m'\neq m}\phi_{j'm'}\phi_{jm'}\,|\,x_m) = E(y\prod_{m'\neq m}\phi_{jm'}\,|\,x_m).$$

The iterative process for minimizing $E(y - \Pi_1 - \cdots - \Pi_J)$ can be carried out simultaneously, as outlined above. An alternative is to minimize $E(y - \Pi)^2$. Let the minimizing Π be $\Pi_1^{(0)}$. Now minimize $E(y - \Pi_1^{(0)} - \Pi)^2$ over Π, and call the minimizing product $\Pi_2^{(0)}$, etc. At the end of the first cycle this stepwise procedure gives $\Pi_1^{(0)} + \cdots + \Pi_J^{(0)}$. Now, keeping $\Pi_2^{(0)}, \ldots, \Pi_J^{(0)}$ fixed, minimize over Π_1 to get $\Pi_1^{(1)}$. Then, keep $\Pi_1^{(1)}, \Pi_3^{(0)}, \ldots, \Pi_J^{(0)}$ fixed and minimize over Π_2 getting $\Pi_2^{(1)}$. The quantity $E(y - \Pi_1 - \cdots - \Pi_J)$ keeps decreasing.

Assuming convergence to Π_j, $j = 1, \ldots, J$:

Proposition 2.4. *The* $\{\Pi_j\}$ *satisfy the stationary equations for the simultaneous minimization of*

$$E(y - \Pi_1 - \cdots - \Pi_J)^2.$$

Proof. Straightforward verification.

What is difficult is not convergence, but there may be local minima, and that the algorithm may converge to one of these. We give more attention to this problem in the bivariate case:

2.2. *Bivariate Case for Independent* x_1, x_2.

Suppose that x_1, x_2 are independent, i.e. $P(dx_1, dx_2) = P_1(dx_1)P_2(dx_2)$. For minimizing $E(y - \Pi)^2$ the stationary equations (2.3) are

$$\phi_1(x_1) = \frac{E(y\phi_2 | x_1)}{E\phi_2^2}$$

$$\phi_2(x_2) = \frac{E(y\phi_1 | x_2)}{E\phi_1^2}.$$

These can be combined into the linear equation

$$\lambda\phi_1(x_1) = E[yE(y\phi_1 | x_2) | x_1] \tag{2.5}$$

where $\lambda = E\phi_1^2 \cdot E\phi_2^2$. Suppose $P_m(dx_m) = h_m(x_m)\,dx_m$, then (2.5) has the form

$$\lambda\phi_1(x_1) = \int K(x_1, x_1')\,h_1(x_1')\,\phi_1(x_1')\,dx_1' \tag{2.6}$$

where

$$K(x_1, x_1') = \int E(y|x_1, x_2) E(y|x_1', x_2) h_2(x_2) dx_2 .$$

Now (2.6) has eigenfunctions $\{\phi_{j1}(x_1)\}$ orthonormal with respect to h_1 and eigenvalues $\{\lambda_j\}$. The corresponding $\{\phi_{j2}(x_2)\}$ are given by

$$\phi_{j2}(x_2) = E(y\,\phi_{j1}|x_2) .$$

Schmidt (1907) showed (essentially) that the sum of the $\Pi_j = \phi_{j1}(x_1)\phi_{j2}(x_2)$, $j = 1, \ldots, J$, is the unique solution to the problem of minimizing

$$E(y - \Pi_1 - \cdots - \Pi_J)^2$$

and that for these minimizing products,

$$E(y - \Pi_1 - \cdots - \Pi_J)^2 = \sum_{J+1}^{\infty} \lambda_j + \sigma^2,$$

where $\sigma^2 = E(y - E(y|\mathbf{x}))^2$. In terms of minimizing $E(y - \Pi)^2$, the product of the dominant eigenfunctions gives the minimum, there are no other local minima, and the products of the other eigenfunctions are saddle points.

But for x_1, x_2 not independent, there seems to have been no work regarding the solutions of minimizing $E(y - \Pi_1 - \cdots - \Pi_J)^2$. For a while, we hoped that in general there was only one minimum. But while doing numerical work on minimizing $E(y - \Pi)^2$ with dependent x_1, x_2 we discovered an example with multiple local minima. Given the existence of local minima, the establishment of good initial starting functions becomes important, (see Appendix I).

2.3. *Two vs Higher Dimensions.*

Suppose that

$$y = \sum_{j=1}^{J} \prod_{m=1}^{M} \theta_{j,m}(x_m) + \varepsilon$$

with $\{\varepsilon\}$ independent noise. If we fit y using the Π-method does it follow that the fitted $\{\phi_{j,m}\}$ are essentially equal to the $\{\theta_{j,m}\}$?

In two dimensions the answer is: usually not. For each i, m, $\phi_{i,m}$ will be a linear combination of the $\{\theta_{j,m}\}$, j = 1, . . . , J with coefficients such that

$$\sum_{1}^{J} \prod_{m} \phi_{j,m} \equiv \sum_{1}^{J} \prod_{m} \theta_{j,m}. \tag{2.7}$$

In general, there are many such linear combinations. But by writing down the equations and trying to solve, one can verify that in 3 or more dimensions, there generally does not exist any linear combinations of the $\{\theta_{jm}\}$ such that (2.7) holds.

Thus, we are led to the tentative conclusion that in two dimensions, even if

$$f(\mathbf{x}) = \sum_{j=1}^{J} \Pi_m \theta_{jm}(x_m)$$

the Π-method will usually produce an estimate

$$\sum_{j=1}^{J} \Pi_m \phi_{j,m}(x_m) \cong \sum_{j=1}^{J} \Pi_m \theta_{j,m}(x_m)$$

such that $\Pi_m \phi_{j,m}(x_m)$ may have little individual resemblance to any of the products $\Pi_m \theta_{jm}(x_m)$.

However, in 3 or more dimensions, if

$$\sum_{j=1}^{J} \Pi_m \phi_{j,m}(x_m) = \sum_{j=1}^{J} \Pi_m \theta_{j,m}(x_m),$$

then the products of the $\phi_{j,m}$ must usually approximate some permutation of the products of the $\theta_{j,m}$. These heuristic conclusions are supported by a simulation reported on in Section 4.2.

3.0. **Implementation - the PIMPLE Program.**

Implementing a data version of the Π-method (PIMPLE = pi-implementation) had two phases. The first is the general implementation idea outlined in section 3.1. The second is in the refinement of this implementation so as to give more control over the degrees of freedom used. Some of the ideas in Breiman (1989) are used and we refer the reader to this previous paper for details.

3.1. *The General Implementation Idea.*

Given a finite data set $(y_n, \mathbf{x}_n)$, $n = 1, \ldots, N$, $\mathbf{x}_n = (x_{in}, \ldots, x_{Mn})$ the basic implementation of the Π-method is as follows: let $\{g_{m,k}\}$, $k = 1, \ldots, k_m$ be univariate functions such that for all k, $g_{m,k}(x)$ is defined on an interval containing the points $\{x_{mn}\}$, $n = 1, \ldots, N$. Consider functions $\phi_m(x)$ of the form $\sum_k \beta_{m,k}\, g_{m,k}(x)$ and look at the problem of minimizing

$$\sum_n (y_n - \prod_m \phi_m(x_{mn}))^2$$

over $\{\beta_{m,k}\}$.

If we hold $\phi_2, \ldots, \phi_M$ constant, then the problem is one of minimizing

$$\sum_n (y_n - \sum_k \beta_{1,k} g_{1,k}(x_{1n}) \prod_{m \neq 1} \phi_m(x_{m,n}))^2 .$$

This is a straightforward regression problem and the $\{\beta_{1,k}\}$ are easily determined. Now hold all ϕ_m but ϕ_2 constant and solve for the $\{\beta_{2,k}\}$. Keep iterating this process until the residual sum-of-squares does not appreciably decrease. Suppose that $J - 1$ products have been estimated, and denote the residuals $y - \Pi_1 - \cdots - \Pi_{J-1}$ by $\{r_n\}$. Repeat the iterative process to find the product Π_J minimizing $\Sigma(r_n - \Pi_J(n))^2$. Then "backfitting" is used. Hold $\Pi_2, \ldots, \Pi_J$ constant and take the functions in Π_1 to minimize

$$\sum_n (y_n - \Pi_1(n) - \cdots - \Pi_J(n))^{2;} .$$

Repeat holding $\Pi_1, \Pi_3, \ldots, \Pi_J$ constant, etc. Keep circulating until the residual sum-of-squares does not appreciably decrease. Call this residual sum-of-squares RSS(J). If RSS(J) is not sufficiently smaller than RSS(J-1), in a sense made precise later, then use only $J - 1$ products in fitting $\{y_n\}$.

The basic algorithm, then, is an iterated sequence of ordinary linear regressions of fairly low dimensionality. Generally 10 or fewer functions are used to fit each variable. The matrix inversions in the regressions are done using double precision Gaussian sweeps, and the major portion of the computing time is in the computation of the updates of the X'X matrix.

3.2. *Strategy for Controlling the Degrees of Freedom.*

In fitting multidimensional functions, controlling the degrees of freedom is an imperative. For example, suppose we used a basis of, say, 7 functions per variable in fitting a 3-dimensional function. Each product requires the estimation of 19 parameters. With three products 57 parameters are being estimated. If a modest data set of size 100, say, is being fit, this virtually guarantees high variance and a noisy overfit. An essential part of the implementation of the Π-method is a strategy for controlling the degrees of freedom. This consists of two parts:

1) Controlling the number of products used in the fit and the dimensionality of the initial basis.

2) Deleting basis elements not important to the fit.

These two parts interact with each other. The larger the number of initial basis elements, the more the choice of which elements are deleted depends on the noise rather than on the underlying function. But with too few initial basis elements the fit to the underlying function may not be adequate.

The criterion we use in both phases is the "generalized cross-validation" estimate of prediction error given by

$$PE_{GCV} = RSS/(1 - NP/N)^2$$

where RSS is the residual sum-of-squares and NP is number of parameters estimated.

Fixing the initial number of basis elements, let $PE_{GCV}(J)$ be the value of PE_{GCV}

using J products. As J increases, if $PE_{GCV}(J) \geq PE_{GCV}(J-1)$, then only $J-1$ products are used in the fit.

Now suppose K basis elements are used per coordinate, and the number of products is determined as above. Denote the resulting value of PE_{GCV} by $PE_{GCV}(K)$. Then the strategy is to start with a small value of K and increase until we find the K^* which minimizes $PE_{GCV}(K)$. At this stage, there are J^* products, each based on K^* basis elements on every coordinate, and the fit has been optimized by iteration and backfitting.

The next process is similar to stepwise variable deletion in regression. In each of the J products $\Pi_1, \ldots, \Pi_J$ the basis element whose removal would cause the smallest increase in RSS is located. Among these, the one causing the smallest rise is deleted, a refitting-backfitting cycle carried out, and the new value of PE_{GCV} computed. At times, two or more elements in the same or different product may be deleted in the same pass. This occurs when, in sequence, their deletion causes almost the same rise in RSS. A logical approach might be to adopt that fit with the minimum PE_{GCV} value. This has the following difficulty: the sequence of PE_{GCV} values is initially decreasing but is also noisy. At some stage there is a rapid increase as basis elements important to the fit are removed. The problem is not to fall into a nonsignificant local minimum, but also not to allow too much deletion.

The approach we take is similar to that used in bivariate smoothing with knot deletion (Breiman and Peters (1988)). Set a threshold value th > 0, let $\hat{\sigma}^2$ be the noise variance estimated from the fit prior to deletion and let $\underline{PE}_{GCV}$ be the minimum value in the PE_{GCV} sequence. Adopt the fit with the fewest number of parameters satisfying

$$PE_{GCV} \leq \underline{PE}_{GCV} + th \cdot \hat{\sigma}^2.$$

We take th in the range 0 to 10 and usually examine the output to decide.

The final fit clearly depends on the number of products used, the number of initial basis elements and the extent of deletion. To assist in the determination of these, we experimented with 5-fold or 10-fold cross-validation. This has a price in computing time, with 5-fold cross-validation taking about 3 times as long as the unvalidated procedure. Still, an improvement in accuracy would be worth the additional cycles.

Unfortunately, cross-validation provided only a small improvement over the PE_{GCV} selection method. In the additive model construction described in Breiman (1989) cross-validation is an essential tool. One difference is that in the additive procedure there is extensive deletion. In the present situation, the deletion is more modest, so that standard measures based on classical analogies are not so biased.

3.3. *The Spline Basis.*

The functions used as the basis are the cubic spline functions 1, x, $[(x - t)^+]^3$, where the initial knots are distributed by the algorithm described in Breiman (1989)

and the conditions $\phi''(x) = 0$ at the endpoints are imposed. Because of the constraint of linear ends, two initial knots is the minimum for a nonlinear fit. The search described in Section (3.2) is started with a linear fit on each variable and then uses two knots per variable three knots, four knots, etc.

The knot deletion process poses some algorithmic complexity as the constraint of linear ends is kept imposed throughout the deletion process. That is, the spline fit is constrained to be linear to the right of the last undeleted knot on the right and similarly on the left, and continuity of the 2nd derivative is kept enforced.

The advantage of the spline basis combined with deletion for fitting univariate functions has been documented in Breiman and Peters (1988). The basic idea is that in most sets of basis functions, i.e. polynomials, deleting one basis function has a global effect on the fit. However, if a knot is deleted, i.e. one of $[(x - t)^+]^3$, then the effect is localized to the vicinity of the knot. Thus, knots will be deleted in regions where the function is smooth, and retained in intervals of rapid change.

4.0. **Examples.**

The first two examples used to illustrate the Π-method and PIMPLE are data sets discussed by Cleveland and Devlin (1988). No matter how many simulated data sets are run on a methodology, actual data continues to be surprising and complex. After, these two examples, we illustrate some aspects of PIMPLE on simulated data.

4.1. *NOX Data.*

These data are from an experiment in which a single-cylinder engine was run with ethanol, and comprise 88 measurements of NOX (Nitrous Oxides) in the exhaust, the equivalence ratio (E), and the compression ratio (C). The purpose of the analysis was to examine how the NOX depended on the two ratios E and C. In their analysis, Cleveland and Devlin used $(NOX)^{1/3}$ as the response variable. We follow this except that we also subtract from the response its median value. The main reason is to get univariate graphs in the products that are easier to interpret.

The original experiment was reported in Brinkman (1981). The data was analyzed by Rodriguez (1985) who fit an additive model using ACE. Cleveland and Devlin (1988) pointed out that a graphical analysis indicated an ExC interaction.

Figure 4 is a scatter plot of E vs C. The compression ratio has only 5 distinct values. With five distinct values, the basis for C consisted only of 2 knots in addition to a constant and linear term. The data in E could support a higher dimensional basis. Starting from 2 knots on up for E, we got the following results:

No. knots	2	3	4	5	6	7
No. products	2	2	2	2	2	2
PE_{GCV}	14.6	16.7	3.5	3.2	3.5	3.5

Using 5 initial knots, the results of the deletion process were

df.	18	17	16	15	14	13	11	10
PE_{GCV}	3.21	3.11	3.07	2.98	2.89	2.90	3.03	3.05

With $\hat{\sigma}^2 = .029$, the best candidate is the fit with 13 df, and $R^2 = .981$.

Figure 5 gives the graphs of the univariate functions in the two products. All are on the same scale but with no location adjustment. The x-axis is labelled so that zero is at the minimum of the corresponding data values for that x and one is at the maximum. The first product is almost completely a main effect due to the E-ratio. It is large and positive in the midrange of the E values and negative for low and high E.

The functions in the 2nd product are not as large as in the first. In fact, if we define

$$\text{Imp}(i,j) = \sum_n (\Pi_i(n) - \bar{\Pi}_i)(\Pi_j(n) - \bar{\Pi}_j)/\text{RSS}$$

then the Imp matrix is

$$\begin{matrix} 53.6 & -2.4 \\ -2.4 & 3.4 \end{matrix}$$

The Π_2 factor is a correction for the lower E values. The correction is positive for low values of the compression ratio, negative for high values.

Figure 6 is a surface plot of the fit. The dominant feature is the E main effect, but the corrections can also be perceived. These results support the assertion by Cleveland and Devlin that there is a nonremovable interaction in these data.

Figure 4

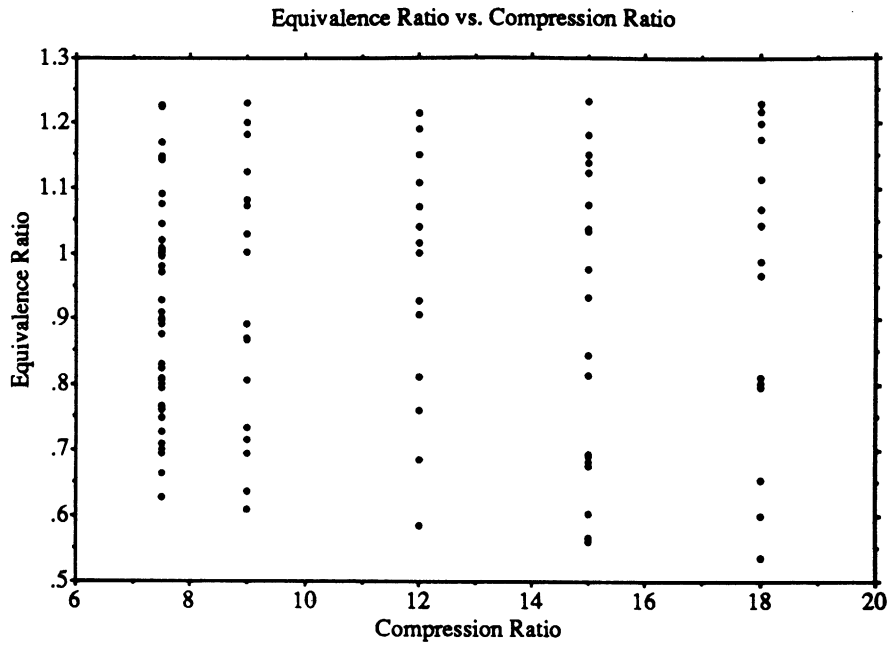
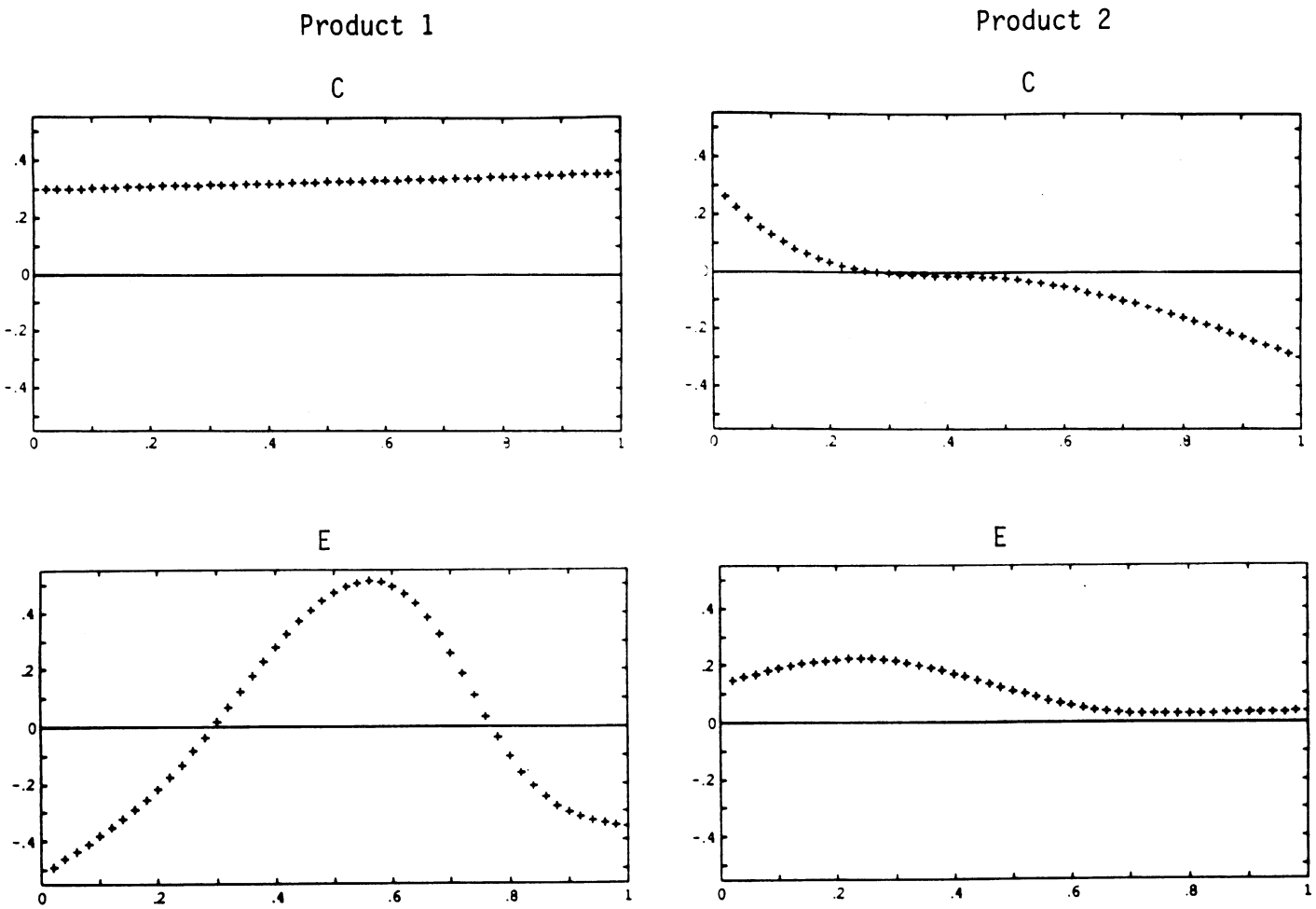

Equivalence Ratio vs. Compression Ratio
Equivalence Ratio
1.3
1.2
1.1
1
.9
.8
.7
.6
.5
6
8
10
12
14
16
18
20
Compression Ratio

Figure 5



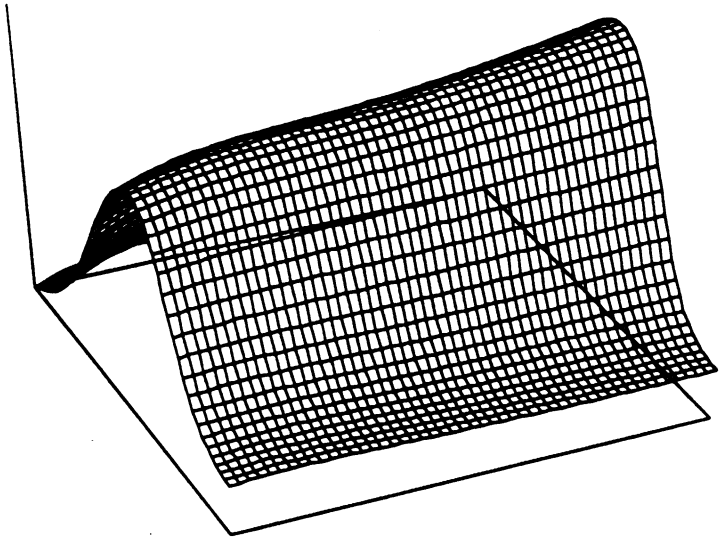


Figure 6

4.2. *Ozone Data.*

This is a data set consisting of 111 measurements of four variables: ozone, solar radiation, temperature, and wind speed on 111 days between May 1 and September 30 of 1973 in New York City (Bruntz et al. (1974)). The purpose is to look at the dependence of ozone on the other variables. Following Cleveland and Devlin again, the one-third power of ozone is used as the dependent variables but with the median subtracted. Varying the number of initial knots from 2 on up gives

No. Knots	2	3	4	5	6
No. Products	2	2	2	2	2
PE_{GCV}	26.1	24.3	23.3	24.1	26.0

After some exploration (see the next section) we decided to subtract the 25th percentile instead of the median. Using 4 knots, the summary of the deletion process is

df	26	20	19	16	15	13	12	11	10	8
PE_{GCV}	22.4	19.3	19.1	19.6	19.8	19.1	19.1	18.6	20.2	21.8

Noting that $\hat{\sigma}^2 = .15$, the candidate of choice is the fit with 11 df and $R^2 = .83$. The plots of the functions in the two products are given in figure 7. The Imp matrix of the two products is:

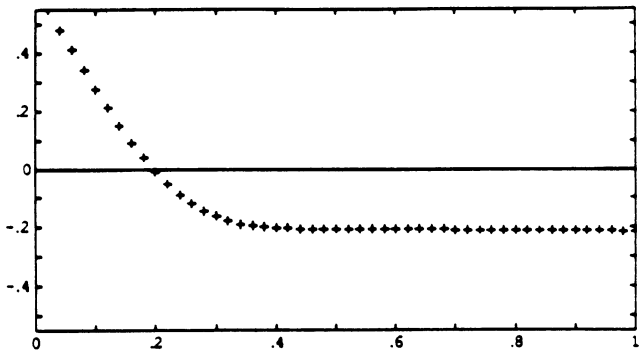
$$\begin{matrix} 2.5 & .7 \\ .7 & .9 \end{matrix}$$

In interpreting figure 7, recall that the response variable (the one-third power of

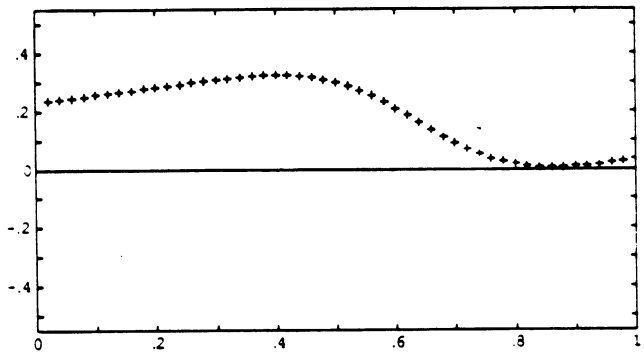
Figure 7

Product 1

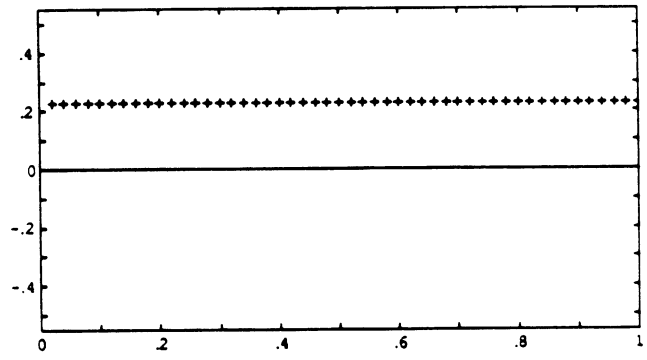
Windspeed



Temperature

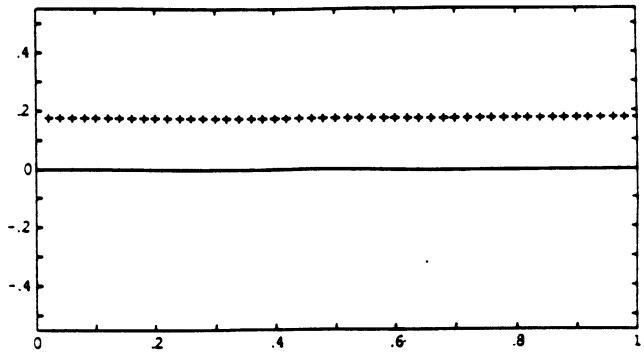


Radiation

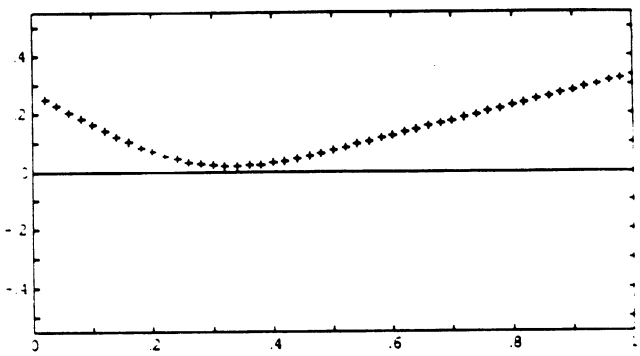


Product 2

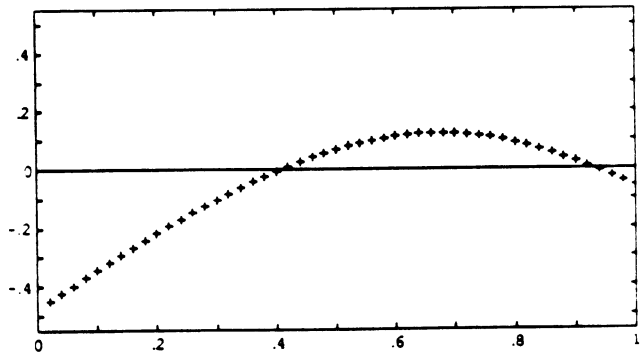
Windspeed



Temperature



Radiation



ozone) is centered at its 25th percentile. The positivity or negativity of the products predict responses relative to the 25th percentile level. The two products are actually bivariate interactions. The first between temperature and wind speed, and the second between temperature and radiation.

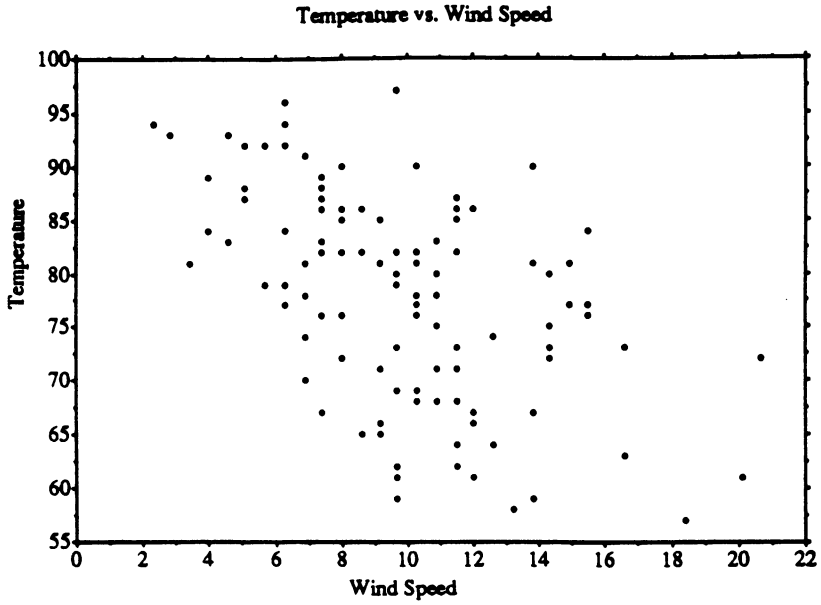
The temperature-windspeed interaction, on first inspection, seems to consist of a large positive component for low wind speeds. A look at the temperature-windspeed scatterplot (Figure 8a) corrects this impression. At low wind speeds the temperature is always in its upper range — precisely where the temperature curve in the first product is close to zero. The dominant contribution of the first product is a negative correction for wind speeds exceeding a certain level.

Looking at the 2nd product, the increase in the temperature function for low values of the temperature seems strange until the temperature-radiation scatterplot (Figure 8b) is examined. Since low temperatures and low radiation generally occur together, we conclude that the lower parts of the temperature and radiation curves work together to produce negative product values in this part of the data.

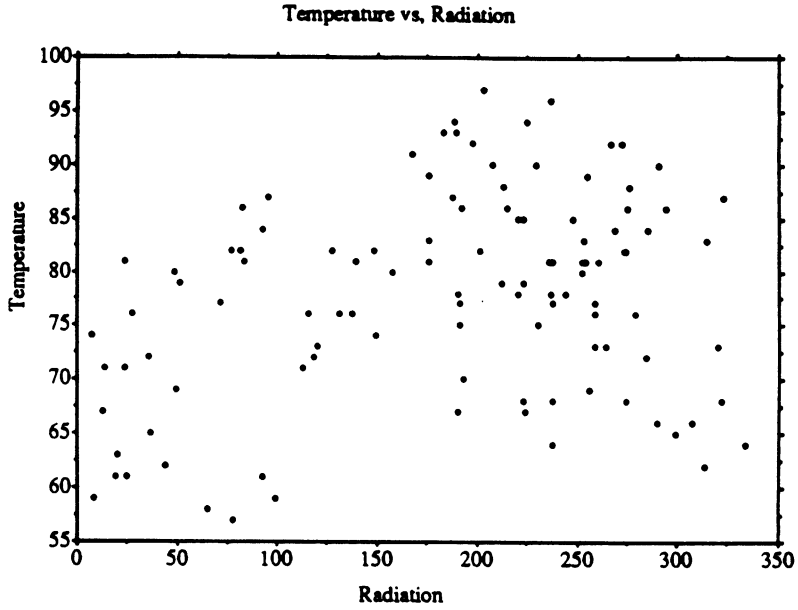
For radiation above a certain threshold the contribution becomes positive, increasingly so as temperature increases. There is an interesting decrease in the radiation curve at the highest radiation levels. That this odd phenomenon is not an artifact in PIMPLE can be verified by looking at the cube-root ozone vs radiation scatterplot

Figure 8

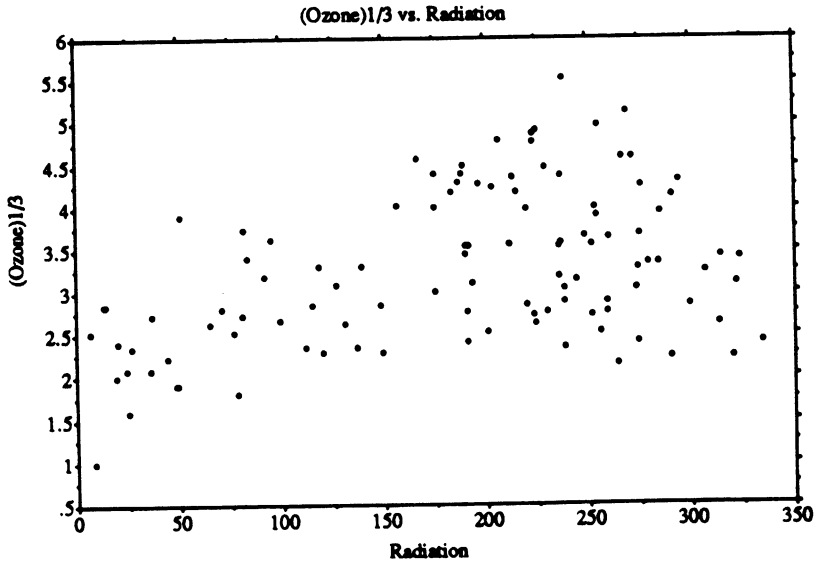
8a



8b



8c



(Figure 8c). Note that past a radiation level of about 250, the cube-root ozone values tend to decrease as radiation increases.

For this data set one would be tempted to construct an additive model. Certainly the author was. Using the methods of Breiman (1989), an additive model for the data was found. Eight df. were used. The plots of the main effect functions are given in Figure 9. The PE_{GCV} for this fit is 23.8, considerably higher than that of the 11df interaction fit. Efforts were also made to fit interaction models to the residuals from the additive fit. These decreased the PE_{GCV} slightly but, not to the level of the 11 df. fit while adding 6-11 more df. Our conclusion is that the two bivariate interaction fit provides a simple and accurate picture of the data.

4.3. *Some Simulated Examples.*

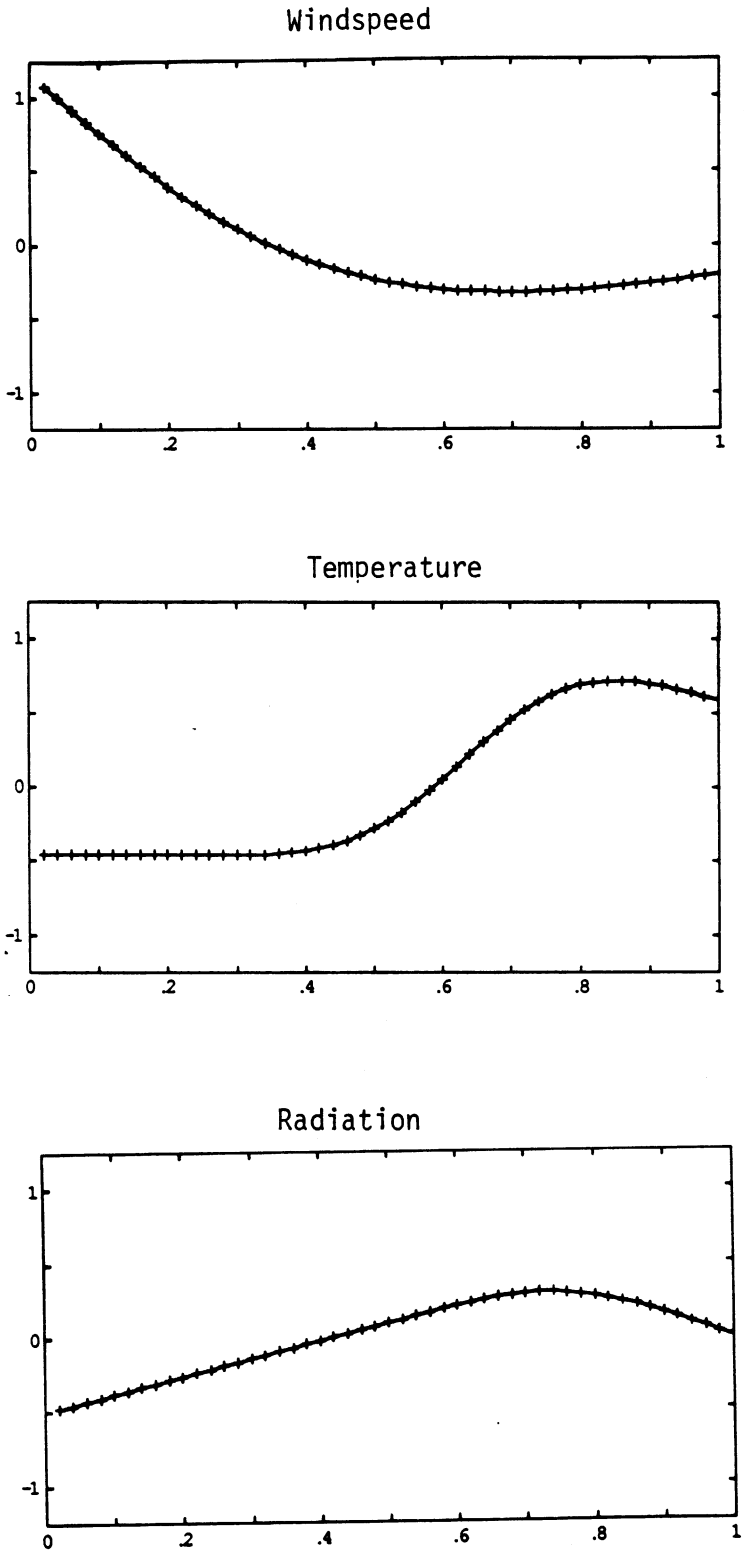
In section 2.4, we noted that if $y = \sum_j \Pi_j + \varepsilon$, then the fit $\sum_j \Pi_j'$ could have different characteristics in two versus higher dimensions. In two dimensions, while $\sum_j \Pi_j' \cong \sum_j \Pi_j$, the individual products $\{\Pi_j'\}$ do not necessarily resemble the $\{\Pi_j\}$. In higher dimensions the situation seems to be that the individual products are similar.

To illustrate this, define

$$\phi(x) = x + x^2$$
$$\theta(x) = x^2$$

and form the functions

Figure 9


Windspeed
1
0
-1
0
.2
.4
.6
.8
1
Temperature
1
0
-1
0
.2
.4
.6
.8
1
Radiation
1
0
-1
0
.2
.4
.6
.8
1

$$f(\mathbf{x}) = \prod_{m=1}^{M} \phi(x_m) + \prod_{m=1}^{M} \theta(x_m)$$

for M = 2,3. The data $\{\mathbf{x}_n\}$ consists of 100 points uniformly distributed on the square or cube with sides $[-1, 1]$, and $y = f(\mathbf{x}) + \varepsilon$, where the noise variance is adjusted so that the signal to noise ratio is around 4.0.

Figure 10 gives the plots of the functions in the two products for three dimensions. The original product functions have been accurately duplicated. Figure 11 gives the plots in the two dimensional case. The functions in the fitted products are considerably altered from the functions in the original products. As one referee remarks, the linear combinations of ϕ and θ that include a linear function of x and match the sum of products are

$$\phi^*(x) = x/\sqrt{2}$$
$$\theta^*(x) = (x/\sqrt{2}) + x^2\sqrt{2}.$$

The functions graphed in figure 11 are close to ϕ^*, θ^*. It is odd and interesting that the two dimensional situation should differ qualitatively from that in higher dimensions.

Another question is how much detail can PIMPLE resolve. Of course, this depends on the signal/noise ratio. But even with high signal/noise ratios, the density of $\{\mathbf{x}_n\}$ points in the region is critical. For instance, consider data

$$y_n = f(\mathbf{x}_n) + \varepsilon_n, \quad n = 1, \ldots, 100$$

with $f(\mathbf{x}) = \exp[x_1 \sin(x_2)]$, using 100 (x_1, x_2) points uniformly distributed on the

Figure 10. Three Dimensions

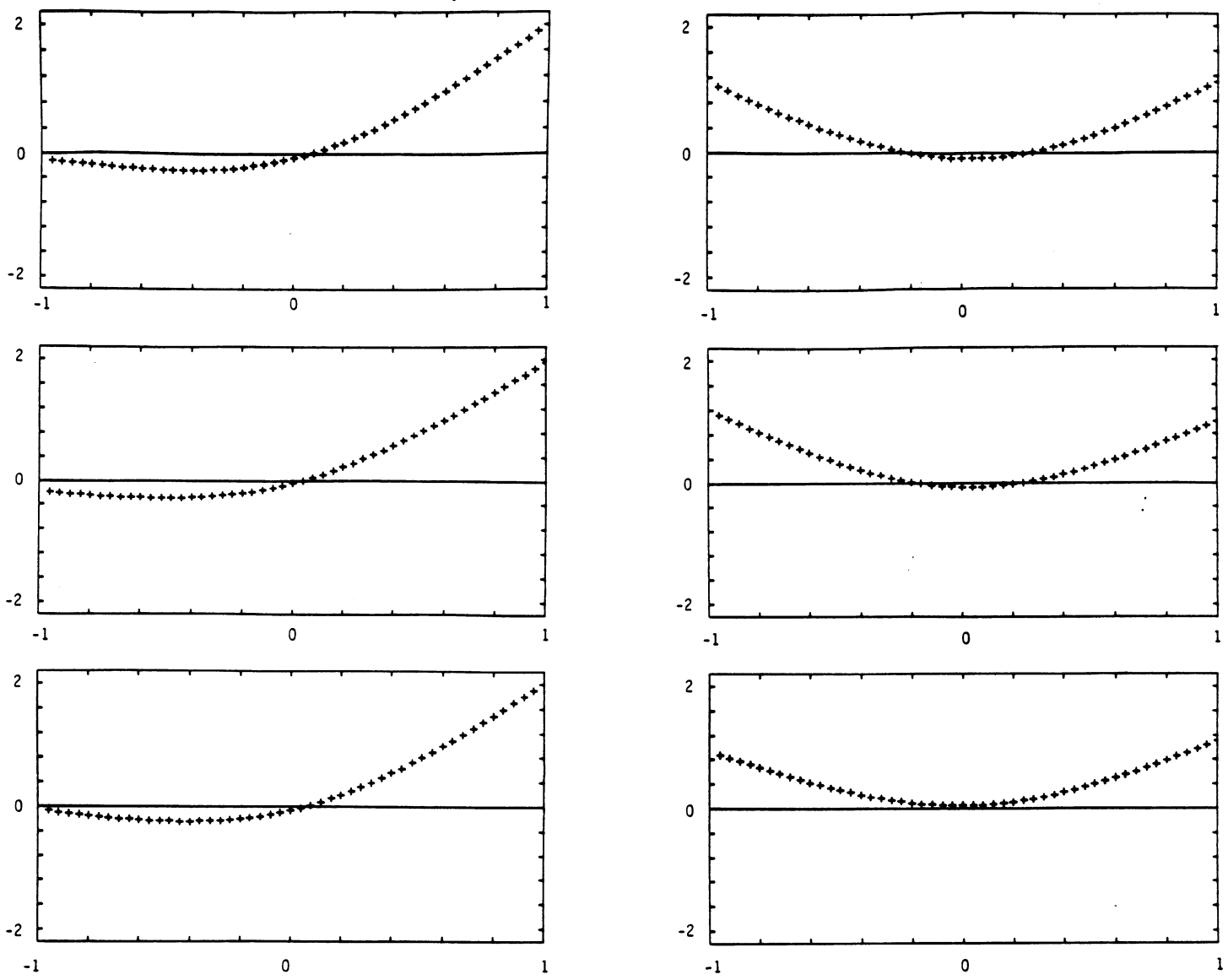
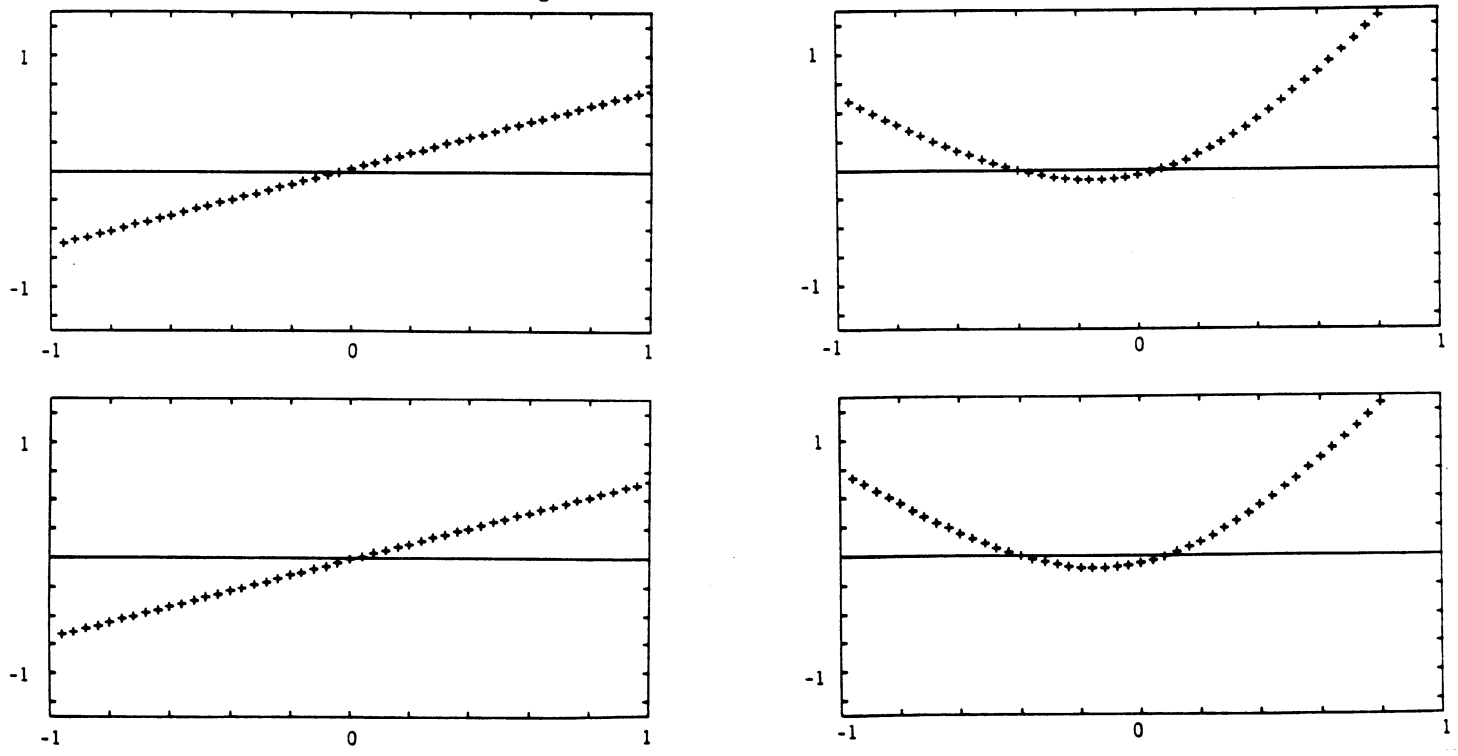


Figure 11 Two Dimensions



square with sides $[-1,+1]$, and s/n $\cong$ 4.

Figure 12a is the plot of $f(\mathbf{x})$ on the given square and 12b is the PIMPLE fit. Now enlarge the square to have sides $[-2,2]$. Figure 12c is the plot of $f(\mathbf{x})$ on this larger square, giving a more complex function. Figure 12d is the PIMPLE fit, again using 100 data points on the square. The fit is not good. Then data using 300 points uniformly distributed on the square was generated. The fit to this data used 3 products and 37 df, and is shown in figure 12e.

4.4. *Some Benchmarks.*

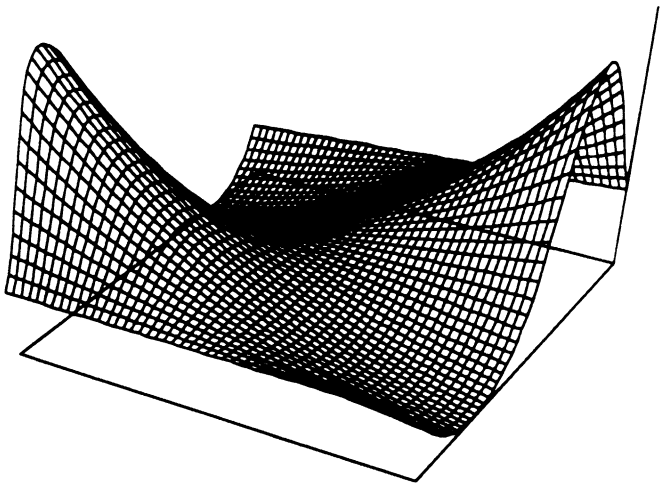
Since a few methods for estimating multivariate functions are in the existing literature and more are liable to appear, some benchmarks for performance are useful. Unfortunately, many papers on smoothing techniques show a few pictures and let it go at that. Others state results for one set of simulated data, which makes comparison impossible unless the same random number generator and same seed is used.

We give some benchmarks below for 100 repetitions of runs on a number of functions in two and three dimensions. Our figure of merit is average root-mean-squared error. That is, in each run with underlying function $f(\mathbf{x})$, data points $\{\mathbf{x}_n\}$, $n = 1, \ldots, N$ and fitted function $\hat{f}(\mathbf{x})$, the RMSE error is defined as

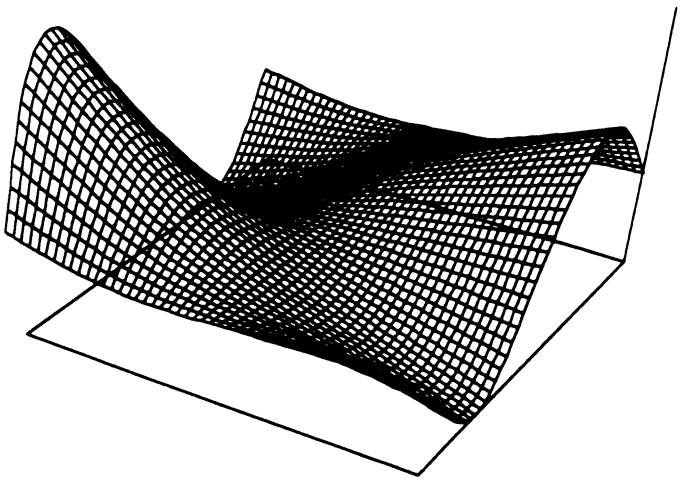
$$[\frac{1}{N}\sum_n (f(\mathbf{x}_n) - \hat{f}(\mathbf{x}_n))^2]^{1/2}.$$

This is then averaged over the 100 runs. The standard error of the RMSE is also

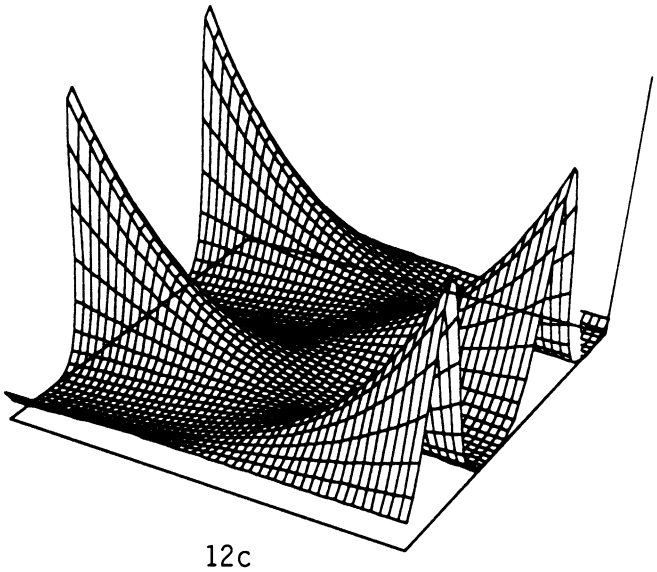
Figure 12



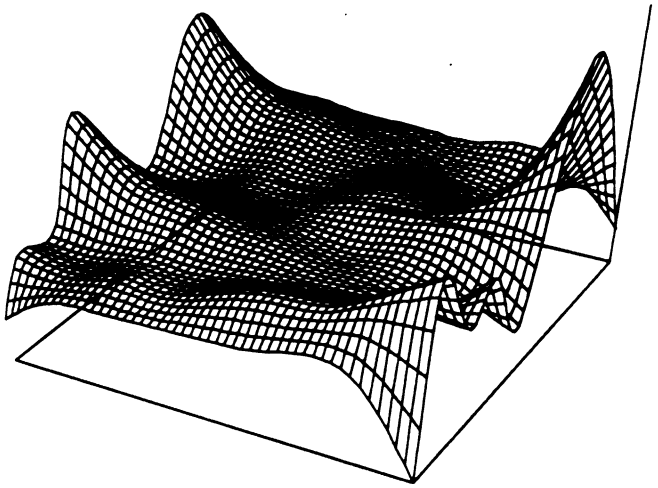
12a



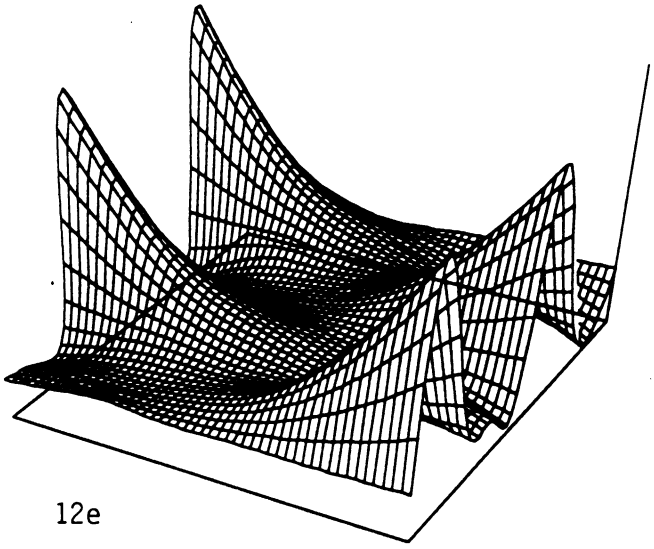
12b



12c



12d



12e

reported. All of the x-regions are squares or cubes with sides as specified, the $\{\mathbf{x}_n\}$ are taken to be 100 uniformly distributed points on the region specified, and the noise is iid $N(0, \sigma^2)$. The list is:

No.	Dimension	Function	Figure	Side	S/N	σ
1	2	$\exp[x_1 \sin(\pi x_2)]$	12(a)	[-1,1]	.9	.5
2	2	$3\sin(x_1 x_2)$	1(b)	[-2,2]	1.9	1.0
3	2	GBCW*	2(c)	[0,1]	3.1	1.0
4	3	$\exp[x_1 x_2 \sin(\pi x_3)]$	--	[-1,1]	1.2	.2
5	3	$x_1 x_2 x_3$	--	[-2,2]	1.5	1.0

*see Appendix II for definition.

The procedure used was that specified in Section 3, and all runs were made with the deletion threshold set to zero.

The RMSE results are:

No.	1	2	3	4	5
Av. RMSE	.192	.388	.565	.115	.219
SE	.005	.009	.008	.003	.009

Other parameters were also computed in these runs:

av product: the average number of products used in the fit.

av knots: the average number of knots used in fitting.

av df: the average number of degrees of freedom used in the fit.

The ''true'' prediction error in a fit is defined as $PE = \sigma^2 + (RMSE)^2$. The final value of PE_{GCV} (after deletion) is an estimate of PE. In each set of 100 repetitions, the following were also computed: av true pe; av est pe; rms pe diff. These values are tabled

below:

No.	1	2	3	4	5
av product	1.1	2.0	3.0	1.9	1.2
av knots	3.5	3.3	3.4	3.4	2.3
av df	6.5	13.1	17.4	16.4	5.0
av true pe	.29	1.16	1.33	.054	1.06
av est pe	.27	1.13	1.33	.041	1.05
rms pe diff.	.05	.22	.23	.020	.17

The fifth example is used as a control. Only one product is necessary to approximate $x_1x_2x_3$. If it is known that the interaction is linear, and if it is fitted by $\Pi_m(\alpha_n x_m + \beta_m)$, then only 4 degrees of freedom are used and the RMSE from 100 runs is .20.

5.0. **Fitting y-c.**

The value of $E(y - c - \phi_1(x_1)\phi_2(x_2))^2$ minimized over ϕ_1, ϕ_2 depends on c, say R(c). Look at the two dimensional example with

$$f(\mathbf{x}) = 3.7\exp(\|\mathbf{x} + \mathbf{e}\|^2) + 2.7\exp(\|\mathbf{x} - \mathbf{e}\|^2), \quad \mathbf{e} = (1,1)$$

with s/n $\cong$ 1, and the $\{\mathbf{x}_n\}$ 100 points uniformly distributed on the square with sides $[-2.5, 2.5]$. Denoting $PE_{GCV}(c)$ as the result of fitting $\{y_n - c\}$, Figure 13 is a graph of $PE_{GCV}(c)$ v.s. c using 5 knots and no deletion.

The odd change in figure 13 has a rational explanation. Consider the situation where x_1, x_2 are independent

$$y = f_1(x_1)g_1(x_2) + f_2(x_1)g_2(x_2) + \varepsilon$$

and we are trying to fit $y - c$ with $\phi_{11}(x_1)\phi_{12}(x_2) + \phi_{21}(x_1)\phi_{22}(x_2)$. Assume, to simplify matters, that $Ef_m^2 = Eg_m^2 = 1$, $Ef_m = Eg_m = \mu$, and $Ef_1f_2 = Eg_1g_2 = \rho$. Without going into details, the integral equation (2.5) then gives a cubic equation for λ, with three real roots, denoted $\lambda_1(c)$, $\lambda_2(c)$, $\lambda_3(c)$. Recall that for the minimizing Π_1, Π_2

$$R(c) = E(y - c - \Pi_1 - \Pi_2)^2 = \min_j(\lambda_j(c)) + \sigma^2.$$

What happens is that as c increases, the two smallest eigenvalues cross each other, causing a sharp change in the $R(c)$ curve. Figure 14 is a graph of $R(c)$ vs. c for $\mu = .3$, $\rho = .2$, and σ taken to give s/n $\cong$ 1.

This odd dependence on c is due to the fact that the simulated data is the sum of two products. The behavior with real data is more stable. In the NOX data the values of both the undeleted and deleted PE_{GCV} were virtually constant over the range of c from min (y) to max (y).

The ozone data shows almost the same constancy over the range min (y) to max (y) with the following exception: the minimum deleted values of PE_{GCV} occur at a smaller number of df in the upper mid part of the range of c. Figure 15 shows a graph of the undeleted PE_{GCV} value over the range of y (1.0 to 5.5). This is the solid line. The dotted line is the minimum PE_{GCV} value for fits between 11 and 13 df. The lowest point occurs at c = 4 which is about the 25th percentile of the y-values. Given our

desire for simplicity, this was the centering chosen.

We have generally resolved the question of what c to use by trying a few values between min (y) and max (y). However, in an important special case this search may not be necessary.

Suppose that lower order interaction affects have already been subtracted. For instance, in the bivariate case, let $\theta_1(x_1)$, $\theta_2(x_2)$ minimize

$$E[y - \theta_1(x_1) - \theta_2(x_2)]^2$$

and what is desired is the bivariate fit to

$$\tilde{y} = y - \theta_1 - \theta_2.$$

The reduced $\tilde{y}$ has the property that

$$E(\tilde{y}|x_1) = E(\tilde{y}|x_2) \equiv 0.$$

Consider the minimization of

$$E(\tilde{y} - c - \phi_1(x_1)\phi_2(x_2))^2 \tag{5.1}$$

assuming (x_1, x_2) independent and the minimizing ϕ_1, ϕ_2 not constant. The minimizing value of $c = E\phi_1 \cdot E\phi_2$ and for this value of c, (5.1) becomes

$$E\tilde{y}^2 - 2E\tilde{y}\phi_1\phi_2 + E\phi_1^2 \cdot E\phi_2^2 - (E\phi_1 \cdot E\phi_2)^2.$$

Minimizing over ϕ_2 gives the equation

$$-E(\tilde{y}\phi_1|x_2) + (E\phi_1^2)\phi_2 - (E\phi_1)^2 E\phi_2 = 0.$$

Taking expectation with respect to x_2 leads to $E\phi_2 = 0$, hence to $c = 0$.

Figure 13

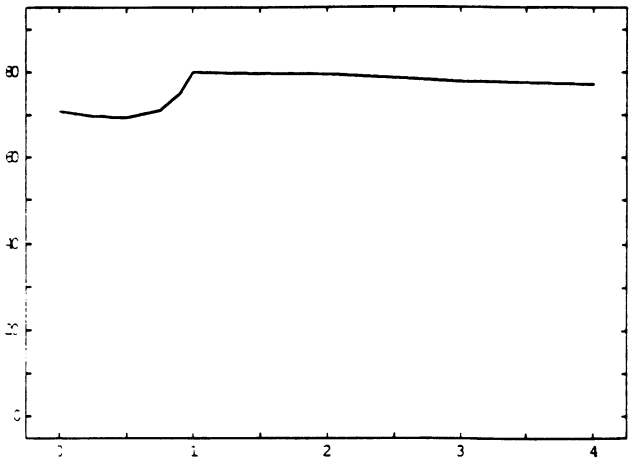


Figure 14

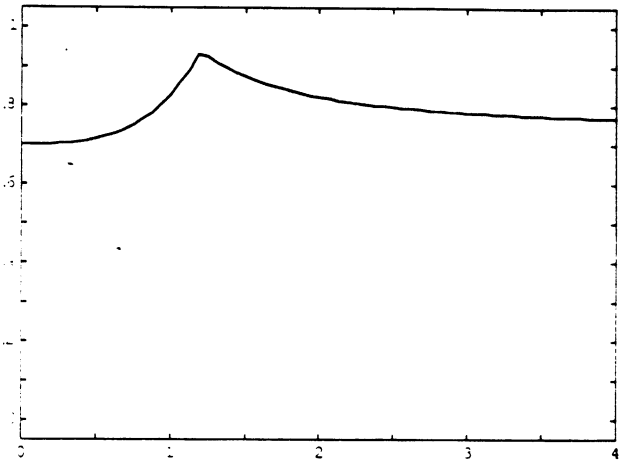
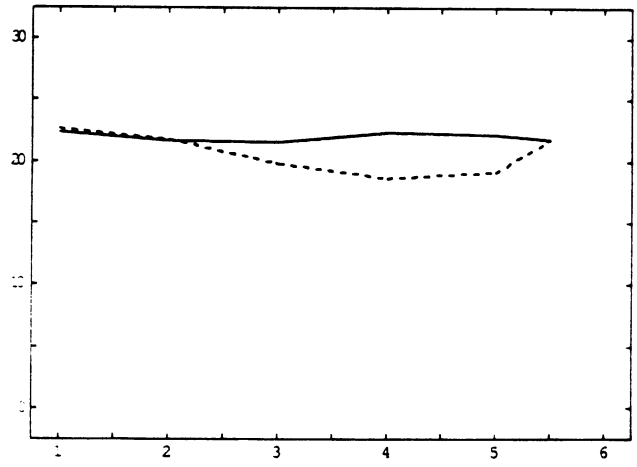


Figure 15



While this argument requires the independence of x_1, x_2, it does provide reason to believe that when lower order interactions have been subtracted out, fitting the reduced y as is will give good results.

6.0. **Conclusions.**

The Π-method as embodied in the PIMPLE program gives an effective method of fitting multidimensional surfaces using sparse noisy data. Tight control is kept on the degrees of freedom used in the fit by adding additional products only if they significantly improve the fit, and by the process of knot deletion. The results are accurate fits and illuminating pictures.

We hope that this will not be the final word on the Π-method. Interesting questions remain, such as the difference between two and higher dimensions. The FORTRAN code for PIMPLE will be available from the author.

7.0. **Acknowledgement.**

This research partially supported by National Science Foundation Grant DMS-8718362

Bibliography

Breiman, L. [1989], Fitting additive models to regression data, Technical Report No. 209, Statistics Department, Univ. of California, Berkeley.

Breiman L., and Peters, S. [1988], Comparing automatic bivariate smoothers, Technical Report No. 161, Statistics Department, University of California, Berkeley.

Breiman, L., and Friedman, J.H. [1985], Estimating optimal transformations for multiple regression and correlation (with discussion), *J. Amer. Statist. Assoc.*, **80**, 580-619.

Brinkman, N.D. [1981], Ethanol fuel - a single-cylinder engine study of efficiency and exhaust emissions, *SAE Transactions*, **90**, N.810345, 1410-1424.

Bruntz, S.M., Cleveland, W.S., Kleines, B. and Wornes, J.L. [1974], The dependence of ambient ozone on solar radiation, wind, temperature, and mixing height, in *Symposium on Atmospheric Diffusion and Air Pollution*, Boston, *American Meteorological Society*, 125-128.

Cleveland, W.S. and Devlin, S. [1988], Locally weighted regression: An approach to regression analysis by local fitting, *J. Amer. Statist. Assoc.*, **83**, 596-640.

Friedman, J.H. [1988], Multivariate adaptive regression splines, Technical Report No. 102, Department of Statistics, Stanford University.

Friedman, J.H. and Stuetzle, W. [1981], Projection pursuit regression, *J. Amer. Statist.*

Assoc., **76**, 817-823.

Gu, C. Bates, D.M., Chen, Z., and Wahba, G. [1988], The computation of GCV functions through Householder Tridiagonalization with application to the fitting of interaction spline models, Technical Report No. 823, Department of Statistics, University of Wisconsin.

Rodriguez, R.N. [1985], A comparison of the ACE and MORALS algorithms in an application to engine exhaust emissions modeling, in *Computer Science and Statistics: Proceedings of the Sixth Symposium on the Interface*, ed. L. Billard, North Holland, N.Y. 159-167.

Schmidt, E. [1907], Zur theorie der linearen und nichtlinearen indegral gleichungen I Teil, *Math. Ann.*, **63**, 433-476.

Schumaker, L.L. [1976], Fitting surfaces to scattered data, in *Approximation Theory* III, G.G. Lorentz, C.K. Chui, and L.L. Schumaker, eds. Academic Press, N.Y. 203-268

Schumaker, L.L. [1984], On spaces of piecewise polynomials in two variables, in *Approximation and Spline Functions*, S.P. Singh - et.al (eds) D. Reidel Publishing Co. 151-197.

Appendix I

Starting Values

It is important to have good starting values for the iterative process used in PIM-PLE.
We use two methods, one in two dimensions and the other in higher dimensions.

In two dimensions we want to minimize

$$\sum_n (y_n - \phi_1(n)\,\phi_2(n))^2$$

where (changing notation a bit)

$$\phi_1(n) = \sum_k \alpha_k f_k(x_{1n})$$

$$\phi_2(n) = \sum_j \beta_j g_j(x_{2n}).$$

Taking partials w.r. to the $\{\alpha_k\}$ gives

$$\sum_n y_n \phi_2(n) f_k(x_{1n}) = \sum_{n,k'} \phi_1^2(n) f_k(x_{1n}) f_{k'}(x_{1n}) \alpha_{k'}.$$

Put

$$H_{kk'} = \sum_n f_k(x_{1n}) f_{k'}(x_{1n}).$$

$$A_{kj} = \sum_n y_n f_k(x_{1n}) g_j(x_{2n}).$$

Using the approximation

$$\sum_n \phi_1^2(n) f_k(x_{1n}) f_{k'}(x_{2n}) \cong \bar{\phi}_1^2 F_{kk'}$$

gives

$$A\beta = \bar{\phi}_1^2 F\alpha$$

Taking partials w.r. to $\{\beta_j\}$ gives the similar equation

$$A'\alpha = \bar{\phi}_2^2 G\beta$$

where $G_{jj'} = \sum_n g_j(x_{2n})\, g_{j'}(x_{2n})$. Solving gives

$$AG^{-1}A'\alpha = \lambda F\alpha, \quad \lambda = \bar{\phi}_1^2\bar{\phi}_2^2.$$

The solution $\{\alpha\}$ required is that corresponding to the highest eigenvalue of this equation. PIMPLE solves this equation and uses the resulting ϕ_1, ϕ_2 as initial values.

The above approach is specific to two dimensions. In higher dimensions a cruder approximation is used. Assuming independence the stationary equations for a single minimizing product are

$$\phi_m(x_m) = cE(y \prod_{m' \neq m} \phi_{m'}(x_{m'}) \mid x_m)$$

or

$$\phi_m^2(x_m) = cE(y \prod_{m'} \phi_{m'}(x_{m'}) \mid x_m).$$

If $\Pi\phi_m$ has a significant component in the y-direction, then a reasonable approximation is

$$\phi_m^2(x_m) \cong cE(y^2 \mid x_m).$$

This suggests that $\sqrt{E(y^2 \mid x_m)}$ be used as initial values for ϕ_m. This is not really satisfactory, since then all initial ϕ_m are non-negative.

A variant of this idea is used in the data implementation: Denote by $\{g_k\}$ the basis functions for x_m. Take $\phi_m^{(1)} = |\sum_k \beta_k g_k|^{1/2}$ where the β_k minimize

$$\sum_k [(y_n^2 - \sum_k \beta_k g_k(x_{nm})]^2.$$

Think of the $\{x_{nm}\}$, $n = 1, \ldots, N$, as being sorted so that $x_{m,n} \le x_{m,n+1}$. Let $\phi_m^{(1)}(x_{mn_0}) = \min_n \phi_m(x_{mn})$. Along with $\phi_m^{(1)}$, consider the alternative function

$$\phi_m^{(2)}(x_{mn}) = \begin{cases} \phi_m^{(1)}(x_{mn}), & x_{mn} \ge x_{mn_0} \\ -\phi_m^{(1)}(x_{mn}), & x_{mn} < x_{mn_0} \end{cases}$$

Find $\alpha, i_1, \ldots, i_M$ to minimize

$$\Sigma (y_n - \alpha \Pi_m \phi_m^{(i_m)}(x_{nm}))^2,$$

and denote the minimizing $\phi_m^{(i_m)}$ by ϕ_m. Find constants $\alpha, d_1, \ldots, d_M$ to minimize

$$\sum_n (y_n - \alpha \prod_m (\phi_m(x_{mn}) - d_m))^2$$

and take the $\phi_m - d_m$ as the initial functions in 3 or more dimensions. This algorithm allows for one sign change in the initial functions. An extension to two or more sign changes would not be difficult. We will do this when we get more experience m PIMPLE's behavior. It seems that generally the global minimum of $\Sigma(y - \Pi)^2$ is in a fairly large valley, so that the initial values for the iteration are not critical as long as they are not drastically distant from the global minimum. Using the above initial values in hundreds of simulated data sets has not given a single detected failure to converge to the global minimum. In three and higher dimensions, we experimented with different methods for computing initial values. The major difference was that PIMPLE converged more rapidly with better initial values.

Appendix II

The functions graphed in figure 1 a)-f) Section 10, and their domains are

1(a). $f(\mathbf{x}) = 3.7 \exp(\|\mathbf{x} + \mathbf{e}\|^2) + 2.7(\|\mathbf{x} - \mathbf{e}\|^2)$, $\mathbf{e} = (1,1)$

Square side: $[-2.5, 2.5]$

1(b). $f(\mathbf{x}) = 3 \sin(x_1 x_2)$

Square side: $[-2, 2]$

1(c). $f(\mathbf{x}) = \exp[-x_1^2 - 3x_2^2 - 4x_1 x_2]$

Square side: $[-1, 1]$

1(d).

$$r_1^2 = (x_1 - 1)^2 + (x_2 - 1)^2$$

$$r_2^2 = (x_1 - 1)^2 + (x_2 - 3)^2$$

$$r_3^2 = (x_1 - 3)^2 + (x_1 - 1)^2$$

$$r_4^2 = (x_1 - 3)^2 + (x_2 - 3)^2$$

$$f(\mathbf{x}) = 4 \exp(-5r_1^4) + 2(-3r_2^4) + 2 \exp(-2r_3^4) + 2 \exp(-4r_4^4)$$

Square side: $[0, 4]$

1(e). $f(\mathbf{x}) = \exp(x_1 \sin(\pi x_2))$

Square side: $[-1.5, 1.5]$

1(f).

$$r_1^2 = (x_1 - 1)^2 + (x_2 - 1)^2$$

$$r_2^2 = (x_1 - 1)^2 + (x_2 - 1)^2$$

$$f(\mathbf{x}) = 1.7 \cos(1.75r_2^2) \exp[-.4r_2^2] + 3.7 \cos(1.75r_1^2) \exp(-.8r_1^2)$$

Square side: $[-2, 2]$.

The starred function GBCW in the benchmarks (Section 4.4) is the function given in Chu et. al [1988] and graphed in figure 2. Its equation is: let

$$a = 40\exp[\,8((x_1 - .5)^2 + (x_2 - .5)^2)\,]$$
$$b = \exp[\,8((x_1 - .2)^2 + (x_2 - .7)^2)\,]$$
$$c = \exp[\,8((x_1 - .7)^2 + (x_2 - .2)^2)\,],$$

then

$$f(\mathbf{x}) = a/(b + c)$$