

## DISCRETE PIECEWISE MONOTONIC APPROXIMATION BY A STRICTLY CONVEX DISTANCE FUNCTION

I. C. DEMETRIOU

**ABSTRACT.** Theory and algorithms are presented for the following smoothing problem. We are given  $n$  measurements of a real-valued function that have been altered by random errors caused by the deriving process. For a given integer  $k$ , some efficient algorithms are developed that approximate the data by minimizing the sum of strictly convex functions of the errors in such a way that the approximated values are made up of at most  $k$  monotonic sections. If  $k = 1$ , then the problem can be solved by a special strictly convex programming calculation. If  $k > 1$ , then there are  $O(n^k)$  possible choices of the monotonic sections, so that it is impossible to test each one separately. A characterization theorem is derived that allows dynamic programming to be used for dividing the data into optimal disjoint sections of adjacent data, where each section requires a single monotonic calculation. It is remarkable that the theorem reduces the work for a global minimum to  $O(n)$  monotonic calculations to subranges of data and  $O(ks^2)$  computer operations, where  $s - 2$  is the number of sign changes in the sequence of the first divided differences of the data. Further, certain monotonicity properties of the extrema of best approximations with  $k$  and  $k - 1$ , and with  $k$  and  $k - 2$  monotonic sections make the calculation quite efficient. A Fortran 77 program has been written and some numerical results illustrate the performance of the smoothing technique in a variety of data sets.

### 1. INTRODUCTION

Monotonic approximation to discrete data has attracted much interest in the past four decades because of its wide range of applications in various disciplines (see Barlow, Bartholomew, Bremner, and Brunk [1] and Robertson, Wright, and Dykstra [7] for over one thousand references related to this subject). Demetriou and Powell [5] raised the important question of piecewise monotonicity, provided methods of solution for the least squares case, and identified questions for further research. The main difficulty in solving the piecewise monotonic approximation problem is that the optimal positions of the extrema have to be found automatically, among so many combinations of extrema that it is impossible to test each one separately. Here we address this problem by employing a strictly convex distance function and present a theory that results in the development of highly efficient algorithms. The conditions of piecewise monotonicity

Received by the editor February 27, 1992 and, in revised form, March 23, 1993 and December 2, 1993.

1991 *Mathematics Subject Classification.* Primary 41A29, 65D10.

*Key words and phrases.* Data smoothing, piecewise monotonic approximation, isotonic, distance function, dynamic programming.

are equivalent to requiring a prescribed number of sign changes, say  $q$ , in the first divided differences of the approximation. Therefore, the proposed algorithms can be applied for smoothing data that are so rough that the number of sign changes in its first divided differences is much greater than  $q$ . A suitable value for  $q$  can be selected by plotting the data or by forming tables of the first divided differences of the data and checking for sign alterations. One of the advantages of this approach to data smoothing is that it gives properties which occur in a wide range of underlying functions. Furthermore, the approximation technique may be particularly useful in statistical data analysis when estimating conditional expectations subject to piecewise monotonic components.

Let  $n, k$  be positive integers such that  $k < n$ , and let  $\varphi$  be a data vector in  $\mathbb{R}^n$  whose components are  $\{\varphi_i : i = 1, 2, \dots, n\}$ . We assume that each  $\varphi_i$  is a measurement of a function value  $f(x_i)$ , from some unknown underlying function  $f$ , after it has been altered by a random error  $\varepsilon_i$  such that  $\varphi_i = f(x_i) + \varepsilon_i$ . We are concerned with vectors  $\mathbf{y} \in \mathbb{R}^n$  which approximate  $\varphi$  in such a way that the sequence  $\{y_{i+1} - y_i : i = 1, 2, \dots, n-1\}$  has at most  $k-1$  sign changes. We call  $\mathbf{y} \in \mathbb{R}^n$  *feasible* if there are integers  $\{t_j : j = 0, 1, \dots, k\}$  satisfying the conditions

$$(1.1) \quad 1 = t_0 \leq t_1 \leq \dots \leq t_k = n$$

such that for  $j = 1, 2, \dots, k$  the sequence

$$\{y_i : i = t_{j-1}, t_{j-1} + 1, \dots, t_j\}$$

is monotonic increasing if  $j$  is odd, and monotonic decreasing if  $j$  is even. In other words  $\mathbf{y}$  has to satisfy the constraints

$$(1.2) \quad \begin{aligned} y_{t_{j-1}} &\leq y_{t_{j-1}+1} \leq \dots \leq y_{t_j} && \text{if } j \text{ is odd,} \\ y_{t_{j-1}} &\geq y_{t_{j-1}+1} \geq \dots \geq y_{t_j} && \text{if } j \text{ is even.} \end{aligned}$$

We let  $\mathbb{Y}(k, n)$  be the set of all feasible vectors  $\mathbf{y}$  in  $\mathbb{R}^n$  with  $k$  monotonic sections and assume in this paper that the first monotonic section of each  $\mathbf{y}$  in  $\mathbb{Y}(k, n)$  is monotonic increasing.

Let  $h_i$ ,  $1 \leq i \leq n$ , be a strictly convex and continuous function from  $\mathbb{R}$  to  $\mathbb{R}$  whose smallest value is  $h_i(0) = 0$  and  $h_i(\theta) \rightarrow \infty$  as  $|\theta| \rightarrow \infty$ . We call  $\mathbf{y} \in \mathbb{R}^n$  a *best* or *optimal approximation* from  $\mathbb{Y}(k, n)$  to  $\varphi$  if it minimizes the function of the errors  $\{\varphi_i - y_i : i = 1, 2, \dots, n\}$ ,

$$(1.3) \quad \Phi(\mathbf{y}) = \sum_{i=1}^n h_i(\varphi_i - y_i),$$

subject to  $\mathbf{y} \in \mathbb{Y}(k, n)$ . For example, we may let  $\Phi$  be the  $p$ th power of a finite  $l_p$ -norm on  $\mathbb{R}^n$ ,  $1 < p < \infty$ , in which case  $h_i$  is the expression  $h_i(\theta) = |\theta|^p$ ,  $-\infty < \theta < \infty$ . The choice of  $h_i$  allows different weights to be given to different measurements. In this work,  $h_i(\cdot)$  need not be a differentiable function.

For fixed integers  $\{t_j : j = 1, 2, \dots, k-1\}$  satisfying the conditions (1.1) there exists a unique vector  $\mathbf{y} \in \mathbb{R}^n$  that minimizes the function (1.3) subject to the constraints (1.2) (Demetriou [3, Theorem 1]). Since there is only a finite number of possible  $\{t_j : j = 1, 2, \dots, k-1\}$ , we deduce the existence of an optimal approximation from  $\mathbb{Y}(k, n)$  to  $\varphi$ . Equalities are allowed in (1.1), but it is rather unlikely to occur when  $\mathbf{y}$  is optimal because usually  $\varphi \notin \mathbb{Y}(k, n)$  (Demetriou and Powell [5, Lemma 2]).

Efficiency considerations are important for the solution of this problem because, as an example in Demetriou and Powell suggests, the amount of work for calculating an optimal approximation with  $k$  monotonic sections can require about  $(n/2)^{k-1}/(k-1)!$  linearly constrained minimization calculations in  $n$  variables, which becomes prohibitively expensive even for small values of  $n$  and  $k$ . However, in the least squares case an optimal algorithm of Demetriou and Powell [5] requires only  $O(ns + ks \log s)$  computer operations at the expense of  $s^2$  memory positions, where  $s-2$  is the number of sign changes in the sequence  $\{\varphi_{i+1} - \varphi_i : i = 1, 2, \dots, n-1\}$ . This remarkable result is based on the fundamental property that an optimal approximation is composed of separate optimal monotonic sections where each section is calculated independently of the others.

Here we study the case where the objective function has the form (1.3). A characterization theorem that allows an equivalent dynamic programming formulation of the problem is the subject of §2. It establishes a recurrence formula which can determine optimal values for the integer variables  $\{t_j : j = 1, 2, \dots, k-1\}$ . In §3, an effective technique restricts the values of  $\{t_j : j = 1, 2, \dots, k-1\}$  to the indices of the local extrema of the data, because the optimal values of  $\{t_j : j = 1, 2, \dots, k-1\}$  are a subset of the latter set. These considerations lead to Algorithm 1, which is often slower than the Algorithm 2 presented later in §4, but it is useful for showing the efficacy of the following properties. If we denote by  $\tau(k, t)$  the value of the integer variable  $t_{k-1}$  that occurs in an optimal approximation in  $\mathbb{Y}(k, t)$ , then  $\tau(k, t)$  is a monotonically increasing function of  $t$ . Also,  $\tau(k, t)$  is a monotonically increasing function of  $k$  when  $k$  steps by two and  $t$  remains constant. Therefore,  $\tau(k, t)$  can be restricted at or to the right of already calculated values  $\tau(k', t')$ , to fewer data, say  $t' \leq t$ , and fewer monotonic sections, say  $k' \leq k$ . These properties give theoretical support to a considerable improvement of the dynamic programming procedure when applied to practical calculations. Both, the underlying formula and the monotonicity properties, were known already to Demetriou and Powell [5], but the theory here focuses on algorithms that take into account the position of the rightmost integer variable  $t_{k-1}$ . These algorithms solve the problem routinely by employing only  $O(n)$  monotonic calculations to subranges of data. Further, the theory establishes a methodology which helps to understand the behavior of the underlying approximation, but we restrict our discussion to those results which are useful to the development of the algorithms. The efficacy of least squares versions of the algorithms that employ these formulae is tested by extended experimentation in §5 in a variety of data sets. Section 6 contains some concluding remarks. Finally, an appendix presents an algorithm for monotonic approximation that is useful to the main algorithms in this work.

## 2. THE FUNDAMENTAL RECURRENCE FORMULA

This section states an equivalent minimization problem for the problem of §1 that depends on the decomposition of an optimal approximation from  $\mathbb{Y}(k, n)$  to  $\varphi$  into separate optimal monotonic sections that increase and decrease alternately. Since these sections can be calculated independently of each other, we need to consider an algorithm for monotonic approximation. Therefore, for

given positive integers  $p$  and  $q$ , we define the quantities

$$(2.1) \quad \alpha(p, q) = \min_{y_p \leq y_{p+1} \leq \dots \leq y_q} \sum_{i=p}^q h_i(\varphi_i - y_i), \quad 1 \leq p \leq q \leq n,$$

and

$$(2.2) \quad \beta(p, q) = \min_{y_p \geq y_{p+1} \geq \dots \geq y_q} \sum_{i=p}^q h_i(\varphi_i - y_i), \quad 1 \leq p \leq q \leq n.$$

For any  $\{\varphi_i : i = p, p+1, \dots, q\}$  there exists a unique *best monotonic increasing* (respectively, *decreasing*) approximation  $\{y_i : i = p, p+1, \dots, q\}$  whose components satisfy the constraints that occur in the definition of  $\alpha(p, q)$  (respectively,  $\beta(p, q)$ ). An algorithm for this calculation is given in the appendix in order to be used in an inner loop in the algorithms for the main calculation of §1. The monotonic procedure is justified by the following lemma.

**Lemma 1.** *Let  $J$  be any set of integers such that  $J \subset \{1, 2, \dots, n-1\}$ , and let  $\mathbf{z}$  be the vector that minimizes (1.3) subject to the constraints  $\{y_i \leq y_{i+1} : i \in J\}$ . If  $\mathbf{y}$  minimizes (1.3) subject to the constraints  $\{y_i \leq y_{i+1} : i \in [1, n-1]\}$ , and if  $j$  is an integer in  $[1, n-1]$  such that  $y_j < y_{j+1}$ , then  $z_j \leq y_j$  and  $y_{j+1} \leq z_{j+1}$ . Further, if  $j$  is an integer in  $[1, n-1]$  such that  $z_j \geq z_{j+1}$ , then  $y_j = y_{j+1}$ .*

*Proof.* By Lemma 3 and Theorem 2 of Demetriou [3].  $\square$

According to this lemma, if  $\mathbf{z}$  is the best approximation to  $\boldsymbol{\varphi}$  subject to a subset of the monotonicity constraints  $y_1 \leq y_2 \leq \dots \leq y_n$ , then any constraint violated by  $\mathbf{z}$  is satisfied as an equality by the solution. Thus, Lemma 1 suggests the development of an algorithm that inserts one monotonicity constraint at a time into the calculation, and therefore terminates in exactly  $n-1$  iterations, giving the best monotonic increasing approximation to  $\boldsymbol{\varphi}$ . We note that the sequence  $\{\alpha(p, j) : j = p, p+1, \dots, q\}$  is monotonic increasing, because if we remove the constraint  $y_j \leq y_{j+1}$  from the definition of  $\alpha(p, j+1)$ , the value of  $\alpha(p, j+1)$  is not greater than before, and it is equal to  $\alpha(p, j)$ . Similarly,  $\{\beta(j, q) : j = p, p+1, \dots, q\}$  is a monotonic decreasing sequence. This observation will prove useful for the proof of Theorem 1 later. We suppose now that both  $\mathbf{y}$  and the integer variables  $\{t_j : j = 1, 2, \dots, k-1\}$  are optimal; the following lemma states the optimality of each monotonic section of  $\mathbf{y}$ .

**Lemma 2.** *Let  $\Phi$  be the strictly convex function defined by (1.3), and let the integer variables  $\{t_j : j = 1, 2, \dots, k-1\}$  be associated with an optimal approximation  $\mathbf{y}$  from  $\mathbb{Y}(k, n)$  to  $\boldsymbol{\varphi} \notin \mathbb{Y}(k, n)$ . Then  $\{y_i : i = t_{j-1}, t_{j-1}+1, \dots, t_j\}$  is the best monotonic increasing approximation to  $\{\varphi_i : i = t_{j-1}, t_{j-1}+1, \dots, t_j\}$  if  $j$  is odd, and the best monotonic decreasing approximation if  $j$  is even. Further, the interpolation conditions*

$$(2.3) \quad y_{t_j} = \varphi_{t_j}, \quad j = 1, 2, \dots, k-1,$$

*are satisfied.*

*Proof.* By the properties of the functions  $h_i$ , Lemma 3, and Lemma 4 of Demetriou and Powell [5].  $\square$

Lemma 2 states the main property of a piecewise monotonic approximation  $\mathbf{y}$  from  $\mathbb{Y}(k, n)$  to  $\boldsymbol{\varphi}$ . If  $\mathbf{y}$  is optimal, then it consists of optimal disjoint sections of monotonic components that satisfy the interpolation conditions (2.3).

This statement implies that once optimal values of  $\{t_j : j = 1, 2, \dots, k - 1\}$  have been found, then an optimal  $\mathbf{y}$  can be formed by  $k$  separate monotonic approximation calculations between successive  $\varphi_{t_j}$ . The question of how to calculate optimal values of  $\{t_j : j = 1, 2, \dots, k - 1\}$  is answered by Theorem 1 below. There, a recursive formula is derived whose efficiency is entirely due to the small number of operations required in order to identify optimal values for  $\{t_j : j = 1, 2, \dots, k - 1\}$ . However, the method for constructing an optimal approximation will not be described until after the proof of Theorem 1. In order to establish the recursion, we introduce the notation

$$\gamma(k, t) = \min_{\mathbf{y} \in \mathbb{Y}(k, t)} \sum_{i=1}^t h_i(\varphi_i - y_i), \quad t \in [1, n].$$

**Theorem 1.** *Let  $t_{k-1}$  be an optimal integer variable in the usual notation. Then  $\mathbf{y}$  is an optimal approximation from  $\mathbb{Y}(k, n)$  to  $\boldsymbol{\varphi} \in \mathbb{R}^n$  if and only if the equation*

$$(2.4) \quad \gamma(k - 1, t_{k-1}) + d(t_{k-1}, n) = \min_{1 \leq s \leq n} [\gamma(k - 1, s) + d(s, n)]$$

holds, where

$$(2.5) \quad d(t_{k-1}, n) = \begin{cases} \alpha(t_{k-1}, n) & \text{if } k \text{ is odd,} \\ \beta(t_{k-1}, n) & \text{if } k \text{ is even.} \end{cases}$$

*Proof.* Let  $\mathbf{y}$  be an optimal approximation from  $\mathbb{Y}(k, n)$  to  $\boldsymbol{\varphi}$ . Then  $\{y_i : i = 1, 2, \dots, t_{k-1}\}$  is an optimal approximation from  $\mathbb{Y}(k - 1, t_{k-1})$  to  $\{\varphi_i : i = 1, 2, \dots, t_{k-1}\}$ , and  $\{y_i : i = t_{k-1}, t_{k-1} + 1, \dots, n\}$  is the best monotonic increasing approximation to  $\{\varphi_i : i = t_{k-1}, t_{k-1} + 1, \dots, n\}$  if  $k$  is odd, and the best monotonic decreasing approximation if  $k$  is even. The last assertion follows immediately from Lemma 2, while the first is a consequence of the following contradiction. If  $\{\hat{y}_i : i = 1, 2, \dots, t_{k-1}\}$  but not  $\{y_i : i = 1, 2, \dots, t_{k-1}\}$  solves the subproblem that gives an optimal approximation from  $\mathbb{Y}(k - 1, t_{k-1})$  to  $\{\varphi_i : i = 1, 2, \dots, t_{k-1}\}$ , then we can replace each  $y_i$  by  $\hat{y}_i$  for  $i \in [1, t_{k-1}]$  in  $\mathbf{y}$  that preserves the feasibility of the constraints and reduces the value of the objective function (1.3) at  $\mathbf{y}$ . Thus  $\{y_i : i = 1, 2, \dots, t_{k-1}\}$  has to be optimal. Hence, and from the interpolation conditions of Lemma 2 and the equation  $h_{t_{k-1}}(y_{t_{k-1}} - \varphi_{t_{k-1}}) = 0$ , we obtain the equation

$$(2.6) \quad \Phi(\mathbf{y}) = \gamma(k - 1, t_{k-1}) + d(t_{k-1}, n),$$

which implies the inequality

$$(2.7) \quad \Phi(\mathbf{y}) \geq \min_{1 \leq s \leq n} [\gamma(k - 1, s) + d(s, n)].$$

Therefore, the optimality of  $\mathbf{y}$  implies the equation (2.4) provided that (2.7) holds as an equality. Let  $s$  be any integer that satisfies  $1 \leq s \leq n$ , and let  $\mathbf{z} \in \mathbb{R}^n$  be such that the first  $s$  components of  $\mathbf{z}$  give the best approximation from  $\mathbb{Y}(k - 1, s)$  to  $\{\varphi_i : i = 1, 2, \dots, s\}$  and the last  $n - s$  components of  $\mathbf{z}$  give the best monotonic approximation to  $\{\varphi_i : i = s + 1, s + 2, \dots, n\}$ . Then

$$(2.8) \quad \Phi(\mathbf{z}) = \gamma(k - 1, s) + d(s + 1, n),$$

where we define  $d(s + 1, n) = 0$  if  $s + 1 > n$ . Since  $\mathbf{z} \in \mathbb{Y}(k, n)$ , it follows that

$$(2.9) \quad \Phi(\mathbf{y}) \leq \Phi(\mathbf{z}) \leq \gamma(k - 1, s) + d(s, n), \quad s \in [1, n],$$

for any optimal  $\mathbf{y}$  in  $\mathbb{Y}(k, n)$ . Thus the inequality

$$(2.10) \quad \Phi(\mathbf{y}) \leq \min_{1 \leq s \leq n} [\gamma(k - 1, s) + d(s, n)]$$

is satisfied. In view of (2.7) and (2.10), the equation (2.4) follows, and therefore the if-part of the theorem is proved.

To complete the proof of the theorem, we assume that  $t_{k-1}$  satisfies equation (2.4). We note that more than one optimal  $\mathbf{y}$  may yield the value  $\gamma(k, n)$ , but the last  $n - t_{k-1} + 1$  components of  $\mathbf{y}$  always have the values that yield  $d(t_{k-1}, n)$ . Therefore, we must ensure that the  $t_{k-1}$ th component of any optimal vector that occurs in  $\gamma(k - 1, t_{k-1})$  is equal to both  $\varphi_{t_{k-1}}$  and the first component of the unique vector that occurs in  $d(t_{k-1}, n)$ .

We shall show that any vector  $\mathbf{y}$  that occurs in the definition of  $\gamma$  and  $d$  in the left-hand side of the equation (2.4) is an optimal approximation from  $\mathbb{Y}(k, n)$  to  $\varphi$ . Let  $\{y_i^{(\gamma)} : i \in [1, t_{k-1}]\}$  be the components of  $\mathbf{y}$  that occur in the definition of  $\gamma(k - 1, t_{k-1})$ , and let  $\{y_i^{(d)} : i \in [t_{k-1}, n]\}$  be the components of  $\mathbf{y}$  that occur in the definition of  $d(t_{k-1}, n)$ . We set  $t = t_{k-1}$  and construct  $\mathbf{z} \in \mathbb{R}^n$  whose components for  $i \in [1, n]$  are as follows:

$$(2.11) \quad z_i = \begin{cases} \max[y_i^{(\gamma)}, y_i^{(d)}] & \text{if } i = t \text{ for even } k, \\ \min[y_i^{(\gamma)}, y_i^{(d)}] & \text{if } i = t \text{ for odd } k, \\ y_i^{(\gamma)} & \text{if } 1 \leq i \leq t - 1, \\ y_i^{(d)} & \text{if } t + 1 \leq i \leq n, \end{cases}$$

where we omit the penultimate line if  $t - 1 < 1$  and the last line if  $t + 1 > n$ . Thus,  $\mathbf{z}$  is a feasible vector whose rightmost integer variable is at  $t$ . Further,

$$(2.12) \quad \Phi(\mathbf{z}) \leq \gamma(k - 1, t) + d(t, n),$$

but the right-hand side of (2.12) is the least feasible value of  $\Phi(\cdot)$  owing to the first part of the theorem. Therefore, (2.12) must be an equality,

$$(2.13) \quad \Phi(\mathbf{z}) = \gamma(k - 1, t) + d(t, n).$$

We have satisfied optimality for  $\mathbf{z}$ , and we shall show that  $\mathbf{y} = \mathbf{z}$ . For  $\mathbf{y}$  to be optimal, it remains to show that  $y_i^{(\gamma)} = y_i^{(d)}$ . Without loss of generality, and in view of (2.11), we assume that

$$(2.14) \quad z_t = y_t^{(\gamma)} \quad \text{for even } k.$$

Since  $\mathbf{z}$  is optimal (cf. (2.5)),

$$(2.15) \quad \Phi(\mathbf{z}) = \gamma(k - 1, t) + \beta(t + 1, n) \quad \text{for even } k,$$

where we define  $\beta(t + 1, n) = 0$  if  $t + 1 = n$ . In view of (2.13) and (2.15) we obtain  $\beta(t, n) = \beta(t + 1, n)$ . Thus, if  $\varphi_t < y_{t+1}^{(d)}$ , then  $\beta(t, n) > \beta(t + 1, n)$  by the discussion following Lemma 1 on the decreasing monotonicity of  $\beta$ .

It follows that  $\varphi_t \geq y_{t+1}^{(d)}$ , and by optimality,  $y_t^{(d)} = \varphi_t$ . Now (2.14) and the definition of  $z_t$  imply  $y_t^{(\gamma)} \geq \varphi_t$ . If we had

$$(2.16) \quad y_t^{(\gamma)} > \varphi_t,$$

then, in view of Lemma 1, the components  $\{y_i^{(\gamma)} : i \in [1, t]\}$  would have to be calculated subject to the restriction  $y_{t-1} = y_t$ . However, the approximation  $\hat{\mathbf{z}} \in \mathbb{R}^n$  whose components are defined by  $\{\hat{z}_i = y_i^{(\gamma)} : i = 1, 2, \dots, t-1\}$  and  $\{\hat{z}_i = y_i^{(d)} : i = t, t+1, \dots, n\}$  is feasible and relaxes the restriction  $y_{t-1} = y_t$ . It follows that  $\Phi(\hat{\mathbf{z}}) < \Phi(\mathbf{z})$ , which is a contradiction to the optimality of  $\mathbf{z}$ . Thus the inequality (2.16) must be false and the equalities  $y_t^{(\gamma)} = y_t^{(d)} = \varphi_t$  are satisfied. Therefore,  $\mathbf{y}$  is optimal, as was required to be proved. Since the other cases can be treated similarly, the theorem is proved.  $\square$

Theorem 1 establishes an equivalent formulation of the optimization problem of §1 that is highly useful in practice. It shows that the least value of (1.3) subject to the piecewise monotonicity constraints (1.2) can be derived by applying recursively the formula (2.4). Therefore, if  $t$  is an integer in  $[1, n]$ , if  $m$  is an integer in  $[1, k]$ , and if, for a given  $s = s_{m-1}$  in  $[1, t]$ , the variables  $\{s_j : j = 1, 2, \dots, m-2\}$  have been calculated to minimize  $\gamma(m-1, s)$ , then the equation

$$(2.17) \quad \gamma(m, t) = \min_{1 \leq s \leq t} [\gamma(m-1, s) + \alpha(s, t)], \quad m \text{ odd},$$

is satisfied. It follows that the least value of the right-hand side of (2.17) can be found in  $O(t)$  computer operations provided that the sequences  $\{\gamma(m-1, s) : s = 1, 2, \dots, t\}$  and  $\{\alpha(s, t) : s = 1, 2, \dots, t\}$  are available. A similar conclusion follows if  $m$  is even. Therefore,  $\gamma(k, n)$ , which is the least value of (1.3), can be calculated recursively, starting from  $\gamma(1, t) = \alpha(1, t)$  for  $t = 1, 2, \dots, n$ . Then, for  $m = 2, 3, \dots, k$ , having calculated  $\{\gamma(m-1, t) : t = 1, 2, \dots, n\}$ , we derive  $\{\gamma(m, t) : t = 1, 2, \dots, n\}$  by applying the formulae

$$(2.18) \quad \gamma(m, t) = \begin{cases} \min_{1 \leq s \leq t} [\gamma(m-1, s) + \alpha(s, t)], & m \text{ odd}, \\ \min_{1 \leq s \leq t} [\gamma(m-1, s) + \beta(s, t)], & m \text{ even}, \end{cases}$$

and storing the optimal value of  $s$  which occurs in formula (2.18), say  $\tau(m, t)$ . Storing  $\tau(m, t)$  for each value of  $m$  and  $t$  is necessary in order to obtain an optimal sequence  $\{t_j : j = 1, 2, \dots, k-1\}$  by the backward formula

$$(2.19) \quad \begin{aligned} t_k &= n, \\ t_{m-1} &= \tau(m, t_m), \quad m = k, k-1, \dots, 2, \end{aligned}$$

at the end of the recursion. Therefore, the associated optimal approximation to the data is made up of the following sections: monotonic increasing on  $[1, t_1]$  and on  $[t_{j-1}, t_j]$  for odd  $j$  in  $[1, k]$ , and monotonic decreasing on  $[t_{j-1}, t_j]$  for even  $j$  in  $[1, k]$ .

An immediate advantage of formulae (2.18) is their simplicity in implementation, which allows several options. For example, one may calculate initially the numbers (2.1) and (2.2), keep them in store, and then use them in (2.18). Alternatively, one may calculate these numbers before increasing  $t$  in (2.18). However, these formulae are never actually employed in our calculations. They only provide support to the further analysis of improved versions that make use of properties which restrict the range of possible values of

$\{t_j : j = 1, 2, \dots, k-1\}$ . The following two sections are devoted to these improvements.

### 3. RESTRICTING THE RANGE OF SEARCH FOR THE INTEGER VARIABLES

In this section we improve upon the computation by considering values of  $s$  and  $t$  in formulae (2.18) such that  $\varphi_s$  and  $\varphi_t$  are local extrema of the sequence  $\{\varphi_i : i = 1, 2, \dots, n\}$ . Therefore, we need to define the concept of a local extremum of a discrete set of data. For a data ordinate  $\varphi_i$ , let  $l$  and  $g$  be the least and greatest integer, respectively, in  $[1, n]$  such that  $l \leq t \leq g$  and  $\{\varphi_i = \varphi_t : i \in [l, g]\}$ . Then  $\varphi_t$  is a *local maximum* if either  $l = 1$  or  $\varphi_{l-1} < \varphi_l$ , and if either  $g = n$  or  $\varphi_g > \varphi_{g+1}$ . A *local minimum* is defined similarly. The search for  $t_j$  can be restricted to the set of the indices of the local maxima of the data if  $j$  is odd, and to the set of the indices of the local minima if  $j$  is even. Hence, it suffices to calculate  $\alpha(s, t)$  and  $\beta(t, s)$  in (2.18) only when  $s$  is the index of a local minimum and  $t$  is the index of a local maximum of the data. This statement is established by the following lemma.

**Lemma 3.** *Let the integer variables  $\{t_j : j = 1, 2, \dots, k-1\}$  be associated with an optimal approximation  $\mathbf{y}$  from  $\mathbb{Y}(k, n)$  to  $\boldsymbol{\varphi} \notin \mathbb{Y}(k, n)$ . Then for each  $j \in [1, k-1]$  the integer  $t_j$  is the index of a local maximum of  $\{\varphi_i : i = 1, 2, \dots, n\}$  if  $j$  is odd, and the index of a local minimum if  $j$  is even.*

*Proof.* Let  $j$  be an odd integer in  $[1, k-1]$ , and let  $l$  and  $g$  be the least and the greatest integer, respectively, in  $[1, n]$  such that  $l \leq t_j \leq g$  and  $\{y_i = \varphi_{t_j} : l \leq i \leq g\}$ . First we shall show that  $y_{t_j}$  is a local maximum of  $\{y_i : i = 1, 2, \dots, n\}$ , and then we shall show that  $\varphi_{t_j}$  is a local maximum of the data.

Let  $1 < l < g < n$ . We assume that  $y_g < y_{g+1}$  and establish a contradiction. Since the sequence  $\{y_i : i = t_j, t_j + 1, \dots, t_{j+1}\}$  is monotonic decreasing, and since the components of  $\mathbf{y}$  increase to the right of  $g$ , it follows that  $t_{j+1} \in [t_j, g]$ . Hence,  $\mathbf{y}$  consists of at most  $k-2$  monotonic sections. Thus replacing any  $y_i$  such that  $y_i \neq \varphi_i$  by  $\varphi_i$  preserves the feasibility of  $\mathbf{y}$  and reduces the value of  $\Phi(\mathbf{y})$ , which is a contradiction. The argument implies that  $y_g > y_{g+1}$ , and by an extension we obtain  $y_{l-1} < y_l$ , which shows that  $y_{t_j}$  is a local maximum of  $\{y_i : i = 1, 2, \dots, n\}$  as required.

It remains to show that  $\varphi_{t_j}$  is a local maximum of the data. An argument similar to that in the previous paragraph implies that  $t_{j-1}$  and  $t_{j+1} \notin [l, g]$ . We also need to prove that

$$(3.1) \quad \varphi_i = \varphi_{t_j}, \quad l \leq i \leq g.$$

If  $\varphi_i > y_i$  for some  $i \in [l, g]$ , then we can increase  $y_i$ , which preserves the feasibility of  $\mathbf{y}$  and reduces the value of  $\Phi(\mathbf{y})$ . It follows that  $\varphi_i \leq y_i$ ,  $i \in [l, g]$ . Hence, and by Lemma 1, we obtain  $y_l = \varphi_l$ . Thus, let  $r$  be an integer in  $[l+1, t_j-1]$  that gives  $\varphi_r < y_r$ , and let  $\eta$  be the value of  $\theta$  that minimizes the expression  $h_l(\varphi_l - \theta) + \dots + h_{r-1}(\varphi_{r-1} - \theta)$ . Then  $\eta \geq y_r$ , because otherwise we can reduce each of the numbers  $\{y_i : i = l, l+1, \dots, r-1\}$  so little that feasibility of  $\mathbf{y}$  is preserved and  $\Phi(\mathbf{y})$  is reduced. It follows that  $\eta > \varphi_r$ , which, by Lemma 1 and the definition of  $l$  and  $g$ , implies that  $\mathbf{y}$  satisfies the constraint  $y_{r-1} = y_r$ . We may avoid this constraint in the calculation of  $\mathbf{y}$  if we replace each of the components  $\{y_i : i = l, l+1, \dots, r-1\}$  by  $\eta$ , but then we



rule out the optimality of the former  $y$ . A contradiction has been established because of the assumption  $\varphi_r < y_r$ . It follows that  $\{y_i = \varphi_i : i \in [l, t_j]\}$ , and we may similarly show that  $\{y_i = \varphi_i : i \in [t_j, g]\}$ . Thus, (3.1) holds. Hence, and since the components of  $y$  decrease to the left of  $y_l$  and to the right of  $y_g$ , the inequalities

$$(3.2) \quad \varphi_{l-1} < \varphi_l \quad \text{and} \quad \varphi_g > \varphi_{g+1}$$

are implied by Lemma 1. We conclude that, when  $1 < l < g < n$ ,  $\varphi_{l_j}$  is a local maximum of the data. As the other cases may be treated similarly, the proof of the lemma is complete.  $\square$

Demetriou and Powell [5] define  $\mathcal{U}$  and  $\mathcal{L}$  to be ascendingly ordered subsets of  $\{1, 2, \dots, n\}$  as follows:  $\mathcal{U}$  consists of all integers  $t \in [1, n-1]$  such that  $t$  is the index of a local maximum of the data and  $\varphi_t > \varphi_{t+1}$ . Further,  $n \in \mathcal{U}$  if and only if it is the index of a local maximum and  $k$  is even. If  $u$  is the smallest element of  $\mathcal{U}$ , then  $\mathcal{L}$  consists of all integers  $t \in [u+1, n-1]$  such that  $t$  is the index of a local minimum of the data and  $\varphi_t < \varphi_{t+1}$ . Further,  $n \in \mathcal{L}$  if and only if it is the index of a local minimum and  $k$  is odd. Note that  $\mathcal{L}$  and  $\mathcal{U}$  are disjoint, their elements interlace, and they can be formed in  $O(n)$  operations.

These considerations, in view of Lemma 3, allow formulae (2.18) to be replaced by

$$(3.3) \quad \gamma(m, t) = \begin{cases} \min_{s \in [1, t] \cap \mathcal{L}} [\gamma(m-1, s) + \alpha(s, t)], & m \text{ odd,} \\ \min_{s \in [1, t] \cap \mathcal{U}} [\gamma(m-1, s) + \beta(s, t)], & m \text{ even.} \end{cases}$$

Provided that we have calculated the quantities  $\{\alpha(s, t) : 1 \leq s < t, (s, t) \in \mathcal{L} \times \mathcal{U}\}$  and  $\{\beta(s, t) : 1 \leq s < t, (s, t) \in \mathcal{U} \times \mathcal{L}\}$  in (3.3), we derive  $\{\gamma(m, t) : t \in \mathcal{U} \text{ (or } \mathcal{L})\}$  from  $\{\gamma(m, t) : t \in \mathcal{L} \text{ (or } \mathcal{U})\}$  in the order of  $|\mathcal{U}|^2$  operations, which completes the calculation of  $\gamma(k, n)$  in  $O(k|\mathcal{U}|^2)$  operations. We can either work with  $O(|\mathcal{L}|^2)$  storage space or just  $O(k|\mathcal{L}|)$ . If we calculate  $\alpha(s, t)$  and  $\beta(s, t)$  in advance and keep them in store, this would increase by up to  $|\mathcal{L}|^2$  the required memory positions and may be uneconomical on many computer installations, especially if  $|\mathcal{L}| \approx n/2$ . Alternatively, calculating  $\alpha(s, t)$  and  $\beta(s, t)$  before incrementing  $t$  in Algorithm 1 would require only one  $|\mathcal{U}|$ -vector to store  $\{\gamma(m, t) : t \in \mathcal{U}\}$  and one  $|\mathcal{L}|$ -vector to store  $\{\alpha(s, t) : s \in [1, t] \cap \mathcal{L}\}$  or  $\{\beta(s, t) : s \in [1, t] \cap \mathcal{U}\}$  and one  $(k \times |\mathcal{L}|)$ -array to store  $\{t(m, t) : 1 \leq m < k, t \in \mathcal{U} \text{ if } m \text{ is odd or } t \in \mathcal{L} \text{ if } m \text{ is even}\}$ . Hence, a total of  $O(k|\mathcal{L}|)$  memory positions would be required for these arrays. The main disadvantage of this technique is unnecessary work, because each of the required quantities  $\alpha(s, t)$  and  $\beta(s, t)$  is calculated about  $k/2$  times.

However,  $\alpha(s, t)$  and  $\beta(s, t)$  are independent of  $m$ , so they need not be calculated more than once, and they need not be preserved as  $t$  changes, provided that they are used before incrementing  $t$  in (3.3) for all values of  $m$  in  $[1, k-1]$ . Hence, if  $\{\alpha(s, t) : s \in [1, t] \cap \mathcal{L}\}$  are available for some  $t$ , then we should calculate  $\{\gamma(m, t) : m \text{ odd in } [1, k-1]\}$  for this fixed  $t$ . Similarly for  $\{\beta(s, t) : s \in [1, t] \cap \mathcal{U}\}$ . This suggests the most economical way of implementing formulae (3.3). Taking these considerations into account, we derive Algorithm 1, which may be regarded as a version of Algorithm 2 of Demetriou

and Powell [5]. We recall that the first monotonic section of an optimal approximation is required to be increasing. Even when the number of the extrema of the data is at most as large as the required number of extrema of the smoothed values, the first monotonic section is still increasing, because by definition the smallest element in  $\mathcal{L}$  is larger than the smallest element of  $\mathcal{U}$ . Algorithm 3 of the Appendix provides the monotonic calculation.

**Algorithm 1** (Best piecewise monotonic approximation).

*Step 0.* Form the sets  $\mathcal{U}$  and  $\mathcal{L}$ . If  $|\mathcal{U}| + |\mathcal{L}| < k$ , then terminate because  $\mathbf{y} \in \mathbb{Y}(k, n)$ .

*Step 1.* Set  $t$  to the smallest element of  $\mathcal{U}$ . For each odd value of  $m$  in  $[1, k-1]$  store  $\gamma(m, t) = 0$  and  $\tau(m, t) = 1$ .

*Step 2.* If no element of  $\mathcal{L}$  is greater than  $t$ , then branch to Step 9. Otherwise make the least increase in  $t$  such that  $t \in \mathcal{L}$ .

*Step 3.* Calculate the numbers  $\{\beta(s, t) : s = 1, 2, \dots, t\}$  by applying Algorithm 3 to the data  $\{\varphi_s : s = t, t-1, \dots, 1\}$ .

*Step 4.* For each even value of  $m$  in  $[1, k-1]$  store  $\gamma(m, t)$  and  $\tau(m, t)$ , which are obtained from formulae (3.3).

*Step 5.* If no element of  $\mathcal{U}$  is greater than  $t$ , then branch to Step 9. Otherwise make the least increase in  $t$  such that  $t \in \mathcal{U}$ .

*Step 6.* Calculate the numbers  $\{\alpha(s, t) : s = 1, 2, \dots, t\}$  by applying Algorithm 3 to the data  $\{-\varphi_s : s = t, t-1, \dots, 1\}$ .

*Step 7.* For each odd value of  $m$  in  $[3, k-1]$  store  $\gamma(m, t)$  and  $\tau(m, t)$ , which are obtained from formulae (3.3).

*Step 8.* Go to Step 2.

*Step 9.* Set  $t$  and  $t_k$  to  $n$  and  $m = k$ . Let  $t_{k-1}$  be a value of  $s$  that minimizes (3.3).

*Step 10.* For  $j = k-2, k-3, \dots, 0$ , set  $t_j = \tau(j+1, t_{j+1})$ .

*Step 11.* Apply Algorithm 3  $k$  times in order to calculate the components of the optimal approximation which are monotonic increasing on  $[1, t_1]$  and on  $[t_{j-1}, t]$  for odd  $j$  in  $[1, k]$ , and monotonic decreasing on  $[t_{j-1}, t_j]$  for even  $j$  in  $[1, k]$ .  $\square$

The main expenses of Algorithm 1 occur in Steps 3 and 6, where the monotonic algorithm is called  $|\mathcal{U}| - 1$  and  $|\mathcal{L}|$  times, respectively, and  $k$  more times in Step 11. Steps 4 and 7 require  $O(k|\mathcal{U}|^2)$  operations, and the remaining calculation is of the order of  $n$  operations. Therefore, if  $w$  is an upper bound to the numerical work required by Algorithm 3 in Step 3 and Step 6 of Algorithm 1, then Algorithm 1 requires  $O(w|\mathcal{U}| + k|\mathcal{U}|^2)$  computer operations for calculating an optimal approximation with  $k$  monotonic sections to  $\varphi \in \mathbb{R}^n$ . We should note that the main costs of the monotonic calculations depend upon the particular algorithm that is employed for calculating the best approximation by a constant to the data sequence  $\{\varphi_i : i = t, t+1, \dots, j\}$  as it appears in Step 4 of Algorithm 3.

When the least squares version of the monotonic algorithm is applied (see Algorithm 1 of Demetriou and Powell [5]), then the complexity of Algorithm 1 reduces to  $O(n|\mathcal{U}| + k|\mathcal{U}|^2)$ . Extensive numerical experimentation testing the performance of this version on randomly generated data is presented in §5.

An efficient extension of Algorithm 1 is developed in the next section by

restricting the range of the variable  $s$  in formulae (3.3); in this way much unnecessary calculation is avoided. Hence, Algorithm 1 is often slower than Algorithm 2 of §4, but it is useful for comparisons.

4. MONOTONICITY PROPERTIES OF THE RIGHTMOST INTEGER VARIABLE

This section investigates two monotonicity properties of the integer variable  $t_{k-1}$  when considering optimal approximations to subranges of data. They are particularly successful in reducing the numerical work in formulae (3.3), by imposing lower bounds on the variable  $s$ .

In the subsequent discussion we assume that  $\tau(m, t)$  is always made as small as possible whenever there is a choice. The first property is presented in Theorem 3 after we state some technical details in Lemma 4. Let  $s$  and  $j$  be integers such that  $1 \leq s \leq j < n$ , and let  $\Phi_j(s)$  be the least sum of functional changes by a best approximation from  $\mathbb{Y}(m, j)$  to  $\{\varphi_i : i = 1, 2, \dots, j\}$  subject to the restriction that  $t_{m-1} = s$ ; that is,

$$(4.1) \quad \Phi_j(s) = \min_{\substack{\mathbf{y} \in \mathbb{Y}(m, j) \\ t_{m-1} = s}} \sum_{i=1}^j h_i(\varphi_i - y_i).$$

**Lemma 4.** *Let  $h_i$  and  $\Phi_j(s)$  be as already defined, and let  $r$  be any integer such that  $1 \leq r \leq n - j$ . Then the expression*

$$(4.2) \quad [\Phi_{j+r}(s) - \Phi_j(s)], \quad 1 \leq s \leq j,$$

*is a monotonic decreasing function of the integer variable  $s$ .*

*Proof.* We assume that  $m$  is an even integer in  $[1, k]$ ,  $s, t$ , and  $j$  are any integers such that  $1 \leq s < t \leq j < n$ , and  $\mathbf{y}^{(j)} \in \mathbb{R}^j$  and  $\mathbf{y}^{(j+r)} \in \mathbb{R}^{j+r}$  are vectors that occur in the definitions of  $\Phi_j(t)$  and  $\Phi_{j+r}(s)$ , respectively. It suffices to show that

$$(4.3) \quad \Phi_{j+r}(s) + \Phi_j(t) \geq \Phi_{j+r}(t) + \Phi_j(s).$$

Starting from any vectors  $\mathbf{y}^{(j)}$  and  $\mathbf{y}^{(j+r)}$ , we shall construct vectors that are feasible relating to each term in the right-hand side of (4.3). Since  $s < t \leq j < j + r$ , the last monotonic sections of these vectors, in view of Lemma 1, give the inequality  $y_t^{(j)} \geq y_t^{(j+r)}$ . Let  $\nu \in [1, j]$  be the integer  $t - 1$  if  $y_t^{(j)} = y_t^{(j+r)}$ ; otherwise  $\nu$  is any integer in  $[t, j]$  that satisfies  $y_\nu^{(j)} > y_\nu^{(j+r)}$  and either  $\nu = j$  or  $y_{\nu+1}^{(j)} \leq y_{\nu+1}^{(j+r)}$  whenever  $\nu < j$ . We consider  $\mathbf{z}^{(j)} \in \mathbb{R}^j$  and  $\mathbf{z}^{(j+r)} \in \mathbb{R}^{j+r}$  defined by

$$(4.4) \quad z_i^{(j)} = \begin{cases} y_i^{(j+r)}, & i = 1, 2, \dots, s, \dots, \nu, \\ y_i^{(j)}, & i = \nu + 1, \nu + 2, \dots, j, \end{cases}$$

and by

$$(4.5) \quad z_i^{(j+r)} = \begin{cases} y_i^{(j)}, & i = 1, 2, \dots, \nu, \\ y_i^{(j+r)}, & i = \nu + 1, \nu + 2, \dots, j + r, \end{cases}$$

respectively, where we omit the second line of (4.4) if  $\nu = j$ .

By construction,  $z_{s-1}^{(j)} \leq z_s^{(j)}$ , and we have to show that  $\{z_i^{(j)} : i = s, s + 1, \dots, j\}$  is a monotonic decreasing sequence, which is clear if  $\nu = j$ . If

$\nu < j$ , then the sequences  $\{z_i^{(j)} : i = s, s+1, \dots, \nu\}$  and  $\{z_i^{(j)} : i = \nu+1, \nu+2, \dots, j\}$  are monotonically decreasing. It suffices to show that  $z_\nu^{(j)} \geq z_{\nu+1}^{(j)}$ , which is true, because in view of the definitions of  $s$  and  $\nu$ , we obtain  $z_\nu^{(j)} = y_\nu^{(j+r)} \geq y_{\nu+1}^{(j+r)} \geq y_{\nu+1}^{(j)} = z_{\nu+1}^{(j)}$ . Thus,  $\mathbf{z}^{(j)} \in \mathbb{Y}(m, j)$ , and its rightmost integer variable,  $t_{m-1}$  say, satisfies the equation  $t_{m-1} = s$ .

Also, by construction,  $z_{t-1}^{(j+r)} \leq z_t^{(j+r)}$ , and we have to show that  $\{z_i^{(j+r)} : i = t, t+1, \dots, j+r\}$  is a monotonic decreasing sequence, which follows immediately when  $\nu = t-1$ , because  $y_t^{(j)} = y_t^{(j+r)}$  and the sequence  $\{y_i^{(j+r)} : i = t, t+1, \dots, j+r\}$  is monotonically decreasing. If  $\nu \geq t$ , then the sequences  $\{z_i^{(j+r)} : i = t, t+1, \dots, \nu\}$  and  $\{z_i^{(j+r)} : i = \nu+1, \nu+2, \dots, j+r\}$  are monotonically decreasing. It suffices to show that  $z_\nu^{(j+r)} \geq z_{\nu+1}^{(j+r)}$ , which follows from  $z_\nu^{(j+r)} = y_\nu^{(j+r)} > y_\nu^{(j+r)} \geq y_{\nu+1}^{(j+r)} = z_{\nu+1}^{(j+r)}$ . Thus,  $\mathbf{z}^{(j+r)} \in \mathbb{Y}(m, j+r)$ , and its rightmost integer variable,  $t_{m-1}$  say, satisfies the equation  $t_{m-1} = t$ .

Further, in view of the definitions (4.4) and (4.5) we obtain the identity

$$(4.6) \quad \sum_{i=s}^{j+r} h_i(\varphi_i - y_i^{(j+r)}) + \sum_{i=t}^j h_i(\varphi_i - y_i^{(j)}) = \sum_{i=s}^j h_i(\varphi_i - z_i^{(j)}) + \sum_{i=t}^{j+r} h_i(\varphi_i - z_i^{(j+r)}),$$

where we let  $\sum_{i=\nu+1}^j h_i(\varphi_i - y_i^{(j)}) = 0$  if  $\nu = j$ .

Now in view of (4.4), (4.5), and (4.6) the sum  $[\Phi_{j+r}(s) + \Phi_j(t)]$  is expressed in the form

$$(4.7) \quad \Phi_{j+r}(s) + \Phi_j(t) = \sum_{i=1}^j h_i(\varphi_i - z_i^{(j)}) + \sum_{i=1}^{j+r} h_i(\varphi_i - z_i^{(j+r)}).$$

Since  $\mathbf{z}^{(j)}$  and  $\mathbf{z}^{(j+r)}$  are feasible relating to the definitions of  $\Phi_j(s)$  and  $\Phi_{j+r}(t)$ , it follows that the right-hand side of (4.7) is bounded below by  $[\Phi_j(s) + \Phi_{j+r}(t)]$ . Hence we deduce the inequality (4.3). Since the case when  $m$  is odd may be treated similarly, the proof of the lemma is complete.  $\square$

**Theorem 2.** *Let  $\tau(m, l)$  be any value of  $s$  that minimizes (3.3), where  $t$  is replaced by  $l$  and  $\tau(m, l)$  is least if there is any choice. If  $t$  is any integer such that  $t > l$ , then every optimal approximation from  $\mathbb{Y}(m, t)$  to the data  $\{\varphi_i : i = 1, 2, \dots, t\}$  satisfies the condition*

$$(4.8) \quad \tau(m, t) \geq \tau(m, l).$$

*Proof.* Suppose that there exists an optimal approximation from  $\mathbb{Y}(m, t)$  to  $\{\varphi_i : i = 1, 2, \dots, t\}$  such that

$$(4.9) \quad \tau(m, t) < \tau(m, l).$$

Then in view of the optimality and the choice of  $\tau(m, l)$  we have  $\Phi_l(\tau(m, t)) > \Phi_l(\tau(m, l))$ , and in view of Lemma 4 we derive

$$\Phi_l(\tau(m, t)) - \Phi_l(\tau(m, t)) \geq \Phi_l(\tau(m, l)) - \Phi_l(\tau(m, l)).$$

The last two inequalities imply the bound  $\Phi_l(\tau(m, t)) > \Phi_l(\tau(m, l))$ , which contradicts the definition of  $\tau(m, t)$ . The conclusion of the theorem follows.  $\square$

This theorem shows that the integer variable  $t_{m-1} = \tau(m, t)$  of every optimal approximation from  $\mathbb{Y}(m, t)$  to the first  $t$  data is at, or to the right of, the

least integer variable  $\tau(m, l)$  of an optimal approximation to fewer, say the first  $l$ , data. Therefore, it suffices to restrict  $s$  in formulae (3.3) at, or to the right of, the least known value of the numbers  $\{\tau(m, l) : l < t\}$ , where each  $\tau(m, l)$  is chosen to be as small as possible whenever there is a choice. This choice of  $\tau(m, l)$  will prove to be useful to Algorithm 2 below, because no intermediate optimal approximation from  $\mathbb{Y}(m, l)$ ,  $l \leq n$ , to  $\{\varphi_i : i = 1, 2, \dots, l\}$  is excluded from consideration in the construction of the final optimal approximation from  $\mathbb{Y}(k, n)$ ,  $k \geq m$ , to  $\{\varphi_i : i = 1, 2, \dots, n\}$ .

Additional restrictions to the range of  $s$  in (3.3) can be achieved by relating the rightmost integer variables of optimal approximations whose monotonic sections differ by two. Theorem 3 and its supporting lemma below show that all possible values of  $\tau(m + 2, n)$  of optimal approximations from  $\mathbb{Y}(m + 2, n)$  to  $\varphi$  are bounded below by the least value of  $\tau(m, n)$  if there is any choice among the best approximations from  $\mathbb{Y}(m, n)$  to  $\varphi$ . Let  $m$  and  $s$  be integers such that  $1 \leq s \leq n$  and  $1 \leq m \leq k$ , and let  $\Phi(m, s)$  be the sum of functional changes by a best approximation from  $\mathbb{Y}(m, n)$  to  $\varphi$  subject to the restriction that  $t_{m-1} = s$ ; that is,

$$(4.10) \quad \Phi(m, s) = \min_{\substack{\mathbf{y} \in \mathbb{Y}(m, n) \\ t_{m-1} = s}} \sum_{i=1}^n h_i(\varphi_i - y_i).$$

**Lemma 5.** *Let  $h_i$  and  $\Phi(m, s)$  be as already defined. Then the expression*

$$(4.11) \quad [\Phi(m + 2, s) - \Phi(m, s)], \quad 1 \leq s \leq n,$$

*is a monotonic decreasing function of the integer variable  $s$ .*

*Proof.* We assume that  $m$  is an even integer in  $[1, k]$ ,  $t$  and  $s$  are any integers such that  $1 \leq s < t \leq n$ , and  $\mathbf{y}^{(m+2)}$  and  $\mathbf{y}^{(m)}$  are vectors in  $\mathbb{R}^n$  that occur in the definitions of  $\Phi(m + 2, s)$  and  $\Phi(m, t)$ , respectively. It suffices to show that

$$(4.12) \quad \Phi(m + 2, s) + \Phi(m, t) \geq \Phi(m + 2, t) + \Phi(m, s).$$

Starting from any vectors  $\mathbf{y}^{(m+2)}$  and  $\mathbf{y}^{(m)}$ , we shall construct vectors that are feasible relating to the definitions of  $\Phi(m + 2, t)$  and  $\Phi(m, s)$ . Let  $\{t_i^{(m+2)} : i = 0, 1, \dots, m + 2\}$  and  $\{t_i^{(m)} : i = 0, 1, \dots, m\}$  be the values of the integer variables associated with  $\mathbf{y}^{(m+2)}$  and  $\mathbf{y}^{(m)}$ , respectively. Hence,  $t = t_{m-1}^{(m)}$  and  $s = t_{m+1}^{(m+2)}$ , and by assumption

$$(4.13) \quad t_{m+1}^{(m+2)} < t_{m-1}^{(m)}.$$

Let  $j$  be the largest integer in  $[1, m - 1]$  such that

$$(4.14) \quad t_i^{(m)} \leq t_{i+2}^{(m+2)}, \quad i = 0, 1, \dots, j,$$

where without loss of generality we assume that  $j$  is even. It follows that

$$(4.15) \quad t_{j+3}^{(m+2)} < t_{j+1}^{(m)},$$

the existence of the  $j$  being implied by (4.13). The inequalities (4.14) and (4.15) imply the inclusion relation

$$(4.16) \quad [t_{j+2}^{(m+2)}, t_{j+3}^{(m+2)}] \subset [t_j^{(m)}, t_{j+1}^{(m)}].$$

Hence, and from Lemma 1, there exists an integer  $\tau$  in  $[t_{j+2}^{(m+2)}, t_{j+3}^{(m+2)}]$  that satisfies  $y_\tau^{(m+2)} \geq y_\tau^{(m)}$  and either  $\tau = t_{j+2}^{(m+2)}$  or  $y_{\tau-1}^{(m+2)} \leq y_{\tau-1}^{(m)}$ , or, alternatively,  $y_\tau^{(m)} \geq y_\tau^{(m+2)}$  for  $\tau = t_{j+3}^{(m+2)}$ . The definition of  $\tau$  allows  $\{y_i^{(m+2)} : i = t_{j+3}^{(m+2)}, t_{j+3}^{(m+2)} + 1, \dots, n\}$  and  $\{y_i^{(m)} : i = t_{j+1}^{(m)}, t_{j+1}^{(m)} + 1, \dots, n\}$  to have  $m - j - 1$  monotonic sections. Therefore, we consider the vectors  $\mathbf{z}^{(m+2)}$  and  $\mathbf{z}^{(m)}$  in  $\mathbb{R}^n$  defined by

$$(4.17) \quad z_i^{(m+2)} = \begin{cases} y_i^{(m+2)}, & i = 1, 2, \dots, \tau - 1, \\ y_i^{(m)}, & i = \tau, \tau + 1, \dots, t, \dots, n, \end{cases}$$

and

$$(4.18) \quad z_i^{(m)} = \begin{cases} y_i^{(m)}, & i = 1, 2, \dots, \tau - 1, \\ y_i^{(m+2)}, & i = \tau, \tau + 1, \dots, s, \dots, n. \end{cases}$$

Since  $t = t_{m-1}^{(m)} \geq t_{j+1}^{(m)} \geq \tau$ , by an argument similar to that in the paragraph following relation (4.5), the sequence  $\{z_i^{(m+2)} : i = t, t + 1, \dots, n\}$  is monotonic decreasing, and since  $s = t_{m+1}^{(m+2)} \geq t_{j+3}^{(m+2)} \geq \tau$ , the sequence  $\{z_i^{(m)} : i = s, s + 1, \dots, n\}$  is monotonic decreasing too. It follows that  $\mathbf{z}^{(m+2)}$  is an approximation from  $\mathbb{Y}(m+2, n)$  to  $\varphi$  whose rightmost integer variable,  $t_{m+1}$  say, satisfies the equation  $t_{m+1} = t$ , and  $\mathbf{z}^{(m)}$  is an approximation from  $\mathbb{Y}(m, n)$  to  $\varphi$  whose rightmost integer variable,  $t_{m-1}$  say, satisfies the equation  $t_{m-1} = s$ . Then, in view of the definitions of  $\mathbf{z}^{(m+2)}$  and  $\Phi(m+2, t)$ , and  $\mathbf{z}^{(m)}$  and  $\Phi(m, s)$ , we obtain the inequalities

$$(4.19) \quad \sum_{i=1}^n h_i(\varphi_i - z_i^{(m+2)}) \geq \Phi(m+2, t) \quad \text{and} \quad \sum_{i=1}^n h_i(\varphi_i - z_i^{(m)}) \geq \Phi(m, s),$$

respectively. Hence, by using (4.17), (4.18), and (4.19) we derive the expression

$$\begin{aligned} \Phi(m+2, s) + \Phi(m, t) &= \sum_{i=1}^n h_i(\varphi_i - z_i^{(m+2)}) + \sum_{i=1}^n h_i(\varphi_i - z_i^{(m)}) \\ &\geq \Phi(m+2, t) + \Phi(m, s), \end{aligned}$$

as was required. Since the case when  $m$  is odd may be treated similarly, the proof of the lemma is complete.  $\square$

**Theorem 3.** *Let  $\{t_j : j = 1, 2, \dots, m-1\}$  be values of the integer variables associated with an optimal approximation from  $\mathbb{Y}(m, n)$  to  $\varphi$  such that  $t_{m-1} = \tau(m, n)$ , where  $\tau(m, n)$  is as small as possible if there is any choice. Then every optimal approximation from  $\mathbb{Y}(m+2, n)$  to  $\varphi$  satisfies the condition*

$$(4.20) \quad \tau(m+2, n) \geq \tau(m, n)$$

*Proof.* Suppose that there exists an optimal approximation from  $\mathbb{Y}(m+2, n)$  to  $\varphi$  such that  $\tau(m+2, n) < \tau(m, n)$ . Then in view of the optimality and the choice of  $\tau(m+2, n)$  we have

$$\Phi(m+2, \tau(m+2, n)) < \Phi(m+2, \tau(m, n)),$$

and in view of Lemma 5 we derive

$$\begin{aligned} & \Phi(m+2, \tau(m+2, n)) - \Phi(m, \tau(m+2, n)) \\ & \geq \Phi(m+2, \tau(m, n)) - \Phi(m, \tau(m, n)). \end{aligned}$$

The last two inequalities imply the bound  $\Phi(m, \tau(m+2, n)) < \Phi(m, \tau(m, n))$ , which contradicts the definition of  $\tau(m, n)$ . The conclusion of the theorem follows.  $\square$

As was noted just before Lemma 5, the integer variable  $t_{m-1} = \tau(m, n)$  of every best approximation with  $m$  monotonic sections for  $m \geq 4$  may be found at, or to the right of, the rightmost integer variable of least value of a best approximation when  $m$  is reduced by 2. Hence, in view of Theorems 2 and 3, if  $l < t$  and if both  $\tau(m, l)$  and  $\tau(m-2, t)$  are known, we need not consider values of  $s$  in formulae (3.3) which are less than  $\max[\tau(m, l), \tau(m-2, t)]$ , where we set  $\tau(0, t) = 1$ . Thus (3.3) may be expressed in the form,

$$(4.21) \quad \gamma(m, t) = \begin{cases} \min_{s \in [\max[\theta(m), \tau(m-2, t)], t] \cap \mathcal{L}} [\gamma(m-1, s) + \alpha(s, t)], & m \text{ odd,} \\ \min_{s \in [\max[\theta(m), \tau(m-2, t)], t] \cap \mathcal{Z}} [\gamma(m-1, s) + \beta(s, t)], & m \text{ even,} \end{cases}$$

where  $\theta(m)$  is the greatest value of  $\{\tau(m, l) : l < t\}$  that has already been calculated, and as usual  $\tau(m, l)$  is as small as possible.

Further, the values  $\theta(2)$  and  $\theta(3)$  are suitable lower bounds on the values of  $s$  which occur in Step 3 and Step 6, respectively, of Algorithm 1, because they are lower bounds of  $s$  in formulae (4.21).

The results of this section are incorporated into Algorithm 1 and yield the following improved version.

**Algorithm 2** (An improved version for best piecewise monotonic approximation).

*Step 0.* Form the sets  $\mathcal{U}$  and  $\mathcal{L}$ . If  $|\mathcal{U}| + |\mathcal{L}| < k$ , then terminate because  $\mathbf{y} \in \mathbb{Y}(k, n)$ .

*Step 1.* For  $m = 2, 3, \dots, k$  set  $\theta(m) = 1$ . Calculate the numbers  $\{\alpha(1, s) : s = 1, 2, \dots, n\}$  by applying Algorithm 3 to the data  $\{\varphi_s : s = 1, 2, \dots, n\}$ . For each  $t$  in  $\mathcal{U}$ , store  $\gamma(1, t) = \alpha(1, t)$  and  $\tau(1, t) = 1$ . Set  $t$  to the smallest element of  $\mathcal{U}$ . For each odd value of  $m$  in  $[1, k-1]$  store  $\gamma(m, t) = 0$  and  $\tau(m, t) = 1$ . For each  $t \in \mathcal{L}$  store  $\tau(0, t) = 1$ .

*Step 2.* If no element of  $\mathcal{L}$  is greater than  $t$  or if  $k = 2$ , then branch to Step 9. Otherwise make the least increase in  $t$  such that  $t \in \mathcal{L}$ .

*Step 3.* Calculate the numbers  $\{\beta(s, t) : s = \theta(2), \theta(2) + 1, \dots, t\}$  by applying Algorithm 3 to the data  $\{\varphi_s : s = t, t-1, \dots, \theta(2)\}$ .

*Step 4.* For each even value of  $m$  in  $[1, k-1]$ , store  $\gamma(m, t)$  and  $\tau(m, t)$ , which are obtained from formulae (4.21), and set  $\theta(m) = \tau(m, t)$ .

*Step 5.* If no element of  $\mathcal{U}$  is greater than  $t$ , then branch to Step 9. Otherwise make the least increase in  $t$  such that  $t \in \mathcal{U}$ .

*Step 6.* Calculate the numbers  $\{\alpha(s, t) : s = \theta(3), \theta(3) + 1, \dots, t\}$  by applying Algorithm 3 to the data  $\{-\varphi_s : s = t, t+1, \dots, \theta(3)\}$ .

*Step 7.* For each odd value of  $m$  in  $[3, k-1]$ , store  $\gamma(m, t)$  and  $\tau(m, t)$ , which are obtained from formulae (4.21), and set  $\theta(m) = \tau(m, t)$ .

*Step 8.* Go to Step 2.

*Step 9.* Set  $t$  and  $t_k$  to  $n$  and  $m = k$ . If  $k$  is odd, then in equation (4.21) replace  $\max[\theta(m), \tau(m-2, t)]$  by  $\tau(m-2, t^*)$ , where  $t^*$  is the greatest

element in  $\mathcal{L}$ . If  $k$  is even, perform a similar replacement in (4.21). Let  $t_{k-1}$  be a value of  $s$  that minimizes (4.21).

*Step 10–11.* Apply Steps 10–11 of Algorithm 1.  $\square$

Algorithm 2 treats directly the special case  $k = 2$  in Step 2 by ignoring the Steps 3–8. This case requires that the monotonic algorithm be applied only twice to all  $n$  components of  $\varphi$  in order to identify  $t_1$ , which implies optimal complexity. Further, we see that the results of §3 yield formulae (4.21), which identify the main difference between this algorithm and Algorithm 1. Therefore, Algorithm 2 is also a finite procedure for calculating an optimal approximation with  $k$  monotonic sections to  $\varphi \in \mathbb{R}^n$ . However, it is not easy to disentangle the operations count for Algorithm 2 from that for Algorithm 1, because the monotonicity properties of the extrema are incorporated throughout the process of this calculation. Thus, Algorithm 2 is at least as good as Algorithm 1. Therefore, in the absence of any structure in the data, the superiority of Algorithm 2 can only be determined empirically.

The use, however, of formulae (4.21) in practical calculations is quite advantageous, because they provide a calculation that depends on the position of the rightmost integer variable  $t_{m-1}$  of the smoothed values. In effect, as the data increases, the variable  $t_{m-1}$  also increases, which reduces the size of the required monotonic calculations, except if most of the monotonic sections are bunched in the beginning of the range of the data.

The efficacy of the properties of the last two sections is tested extensively in the next section on sets of substantially different data. The numerical results indicate that the formulae (4.21) are far more effective in practice than their complexity suggests in theory.

## 5. NUMERICAL RESULTS AND DISCUSSION

This section presents some details of numerical examples that show the performance of the algorithms of this paper for calculating least squares piecewise monotonic approximations to data contaminated with random errors. The least squares calculations were chosen because they are useful, tractable, easy to apply, and require extremely little time for termination. The monotonic approximations were obtained by Algorithm 1 of Demetriou and Powell [5]. Since all algorithms for piecewise monotonic approximation give a global solution, the measure of the effectiveness of each algorithm is the computational time required for smoothing a variety of data sets.

We consider measurements of the function  $f(x) = \sin(\pi x)$  on the equally spaced grids  $0 = x_1 < x_2 < \dots < x_n = 4$ ,  $0 = x_1 < x_2 < \dots < x_n = 7$ , and  $0 = x_1 < x_2 < \dots < x_n = 14$ , and we generate data sets by adding to the measurements numbers uniformly distributed over the interval  $[-r/n, r/n]$ , where  $r$  is allowed to take the values 0, 10, and 100. We let the number of data take the values  $n = 20, 100, 1000, 5000$ , and 10000. We require best approximations with five, eight, and fifteen monotonic sections on the first, second, and third grid, respectively. We tabulate the CPU times required by Fortran 77 versions of Algorithms 1 and 2 to carry out these calculations on a VAX 8810 computer system.

In Table 1 we present  $n$  versus the data derived for a specific  $r$  as follows: the first, the second, and the third entry of the first column of each  $r$  give



TABLE 1. Tabulation of  $|\mathcal{U}|$ , CPU times to form  $\mathcal{U}$  and  $\mathcal{L}$ , and CPU times to apply the Steps 9–11 of Algorithm 1 (Time is measured in SECONDS.CENTISECONDS on a VAX 8810 computer)

$r$	0			10/n			100/n		
	$k=5$	$k=8$	$k=15$	$k=5$	$k=8$	$k=15$	$k=5$	$k=8$	$k=15$
20	3	4	8	4	4	8	8	7	8
	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01
100	3	4	8	16	10	8	34	32	32
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1000	3	4	8	178	84	39	332	318	299
	0.00	0.01	0.01	0.00	0.01	0.00	0.01	0.00	0.00
5000	3	4	8	867	452	221	1659	1640	1509
	0.02	0.02	0.02	0.02	0.02	0.01	0.02	0.02	0.02
10000	3	4	8	1776	913	433	3335	3267	3042
	0.04	0.03	0.04	0.03	0.03	0.03	0.03	0.03	0.03
	0.06	0.06	0.06	0.10	0.08	0.06	0.13	0.13	0.12

TABLE 2. CPU times required by Algorithm 1 and Algorithm 2 when  $k = 5, k = 8, k = 15$ , and  $r = 0, 10/n, 100/n$  (Time is measured in MINUTES:SECONDS.CENTISECONDS on a VAX 8810 computer)

$r$	$n$	$k = 5$				$k = 8$				$k = 15$			
		$ \mathcal{U} $	Algorithm		$ \mathcal{U} $	Algorithm		$ \mathcal{U} $	Algorithm				
			1	2		1	2		1	2			
$r = 0$	20	3	00:00.00	00:00.01	4	00:00.00	00:00.00	8	00:00.01	00:00.00			
	100	3	00:00.01	00:00.01	4	00:00.01	00:00.01	8	00:00.02	00:00.01			
	1000	3	00:00.06	00:00.05	4	00:00.08	00:00.05	8	00:00.14	00:00.06			
	5000	3	00:00.29	00:00.23	4	00:00.40	00:00.33	8	00:00.68	00:00.48			
	10000	3	00:00.61	00:00.56	4	00:00.80	00:00.53	8	00:01.39	00:00.53			
$r = 10/n$	20	4	00:00.00	00:00.00	4	00:00.01	00:00.00	8	00:00.00	00:00.00			
	100	16	00:00.02	00:00.01	10	00:00.02	00:00.01	8	00:00.02	00:00.01			
	1000	178	00:02.57	00:00.91	84	00:01.32	00:00.41	39	00:00.64	00:00.31			
	5000	867	01:01.71	00:30.28	452	00:33.52	00:19.38	221	00:17.35	00:12.68			
	10000	1776	04:22.55	02:07.48	913	02:15.61	00:42.31	433	01:06.07	00:19.77			
$r = 100/n$	20	8	00:00.00	00:00.00	7	00:00.01	00:00.01	8	00:00.00	00:00.00			
	100	34	00:00.06	00:00.02	32	00:00.05	00:00.02	32	00:00.06	00:00.03			
	1000	332	00:05.14	00:02.29	318	00:04.96	00:01.96	299	00:04.94	00:02.52			
	5000	1659	02:10.80	00:55.60	1640	02:12.79	01:04.24	1509	02:13.61	01:40.72			
	10000	3335	08:57.55	05:10.90	3267	09:03.70	05:10.90	3042	08:38.04	03:42.33			

the number of the local maxima of the data ( $|\mathcal{U}|$ ), the CPU time required to form the sets  $\mathcal{U}$  and  $\mathcal{L}$ , and the CPU time required to calculate an optimal approximation by Steps 9–11 of Algorithm 1, when  $k = 5$ . Similarly for the second column, when  $k = 8$ , and for the third column, when  $k = 15$ . These times show that the corresponding operations make a small contribution to the computational complexity of the algorithm of §§3 and 4. The reason is that only  $O(n)$  operations are needed to form the sets  $\mathcal{U}$  and  $\mathcal{L}$  and to apply the monotonic algorithm.

Next we give Table 2 in order to compare Algorithm 1 with Algorithm 2. Algorithm 1 is useful mainly for testing the practical value of the theory which was established in §§3 and 4 and resulted in Algorithm 2. Table 2 consists of

a triplex of columns for the cases  $k = 5$ ,  $k = 8$ , and  $k = 15$ . Each column-triplex gives the CPU times of Algorithm 1 and Algorithm 2, one column for each algorithm, and includes another column for  $|\mathcal{Z}|$ . Also this table consists of a triplex of rows for the cases  $r = 0$ ,  $r = 10/n$ , and  $r = 100/n$ . Moving down the columns of each row-triplex, we can see the individual performance of each algorithm for different values of  $n$ ; while moving along the rows of each column-triplex, we can compare the algorithms by their execution times on the same data. Both algorithms performed quite well when  $|\mathcal{Z}|$  was small compared with  $n$ , but Algorithm 1 is clearly the most expensive one. Furthermore, Table 2 confirms the strong relation between the required computational time and the number of local extrema of the data.

Although the complexity of Algorithm 1 is  $O(n|\mathcal{Z}| + k|\mathcal{Z}|^2)$ , by inspecting data sets that possess approximately the same  $|\mathcal{Z}|$ , we see that much shorter computation times are achieved in practice as  $k$  increases. The results become by far better when Algorithm 2 is employed. Thus, the effectiveness of the theory presented in §§3 and 4 is confirmed numerically as well.

Comparing the performance of Algorithm 1 with that of Algorithm 2, we see that the second algorithm is twice as fast as the former when applied to the same data sets. Algorithm 2 should be expected to be faster by more than a factor of two over Algorithm 1 when the number of data points increases. The reason is that the monotonic calculations in Steps 3 and 6 of Algorithm 1 involve all the current data, while the size of the monotonic calculations in Algorithm 2 depends on the positions of the rightmost integer variable and the currently rightmost data point. The main advantage of Algorithm 2 is that it provides a calculation, where, as the data increases, a rightmost integer variable of the approximation also increases, which subsequently shortens the ranges of the required monotonic approximations.

In Figures 1, 2, 3, 4, 5, and 6 we draw some best approximations to certain data values.

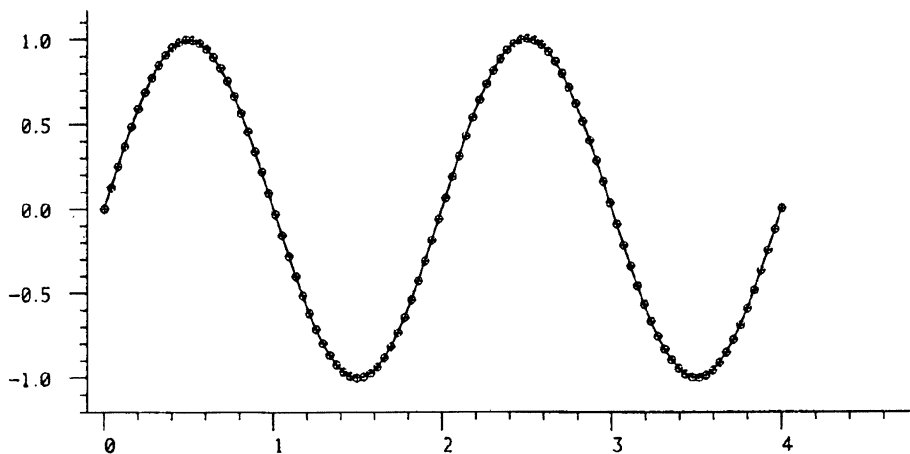


FIGURE 1. Best approximation from  $\mathbb{Y}(k = 5, n = 100)$  to data points derived from  $f(x) = \sin(\pi x)$ ,  $r = 0$

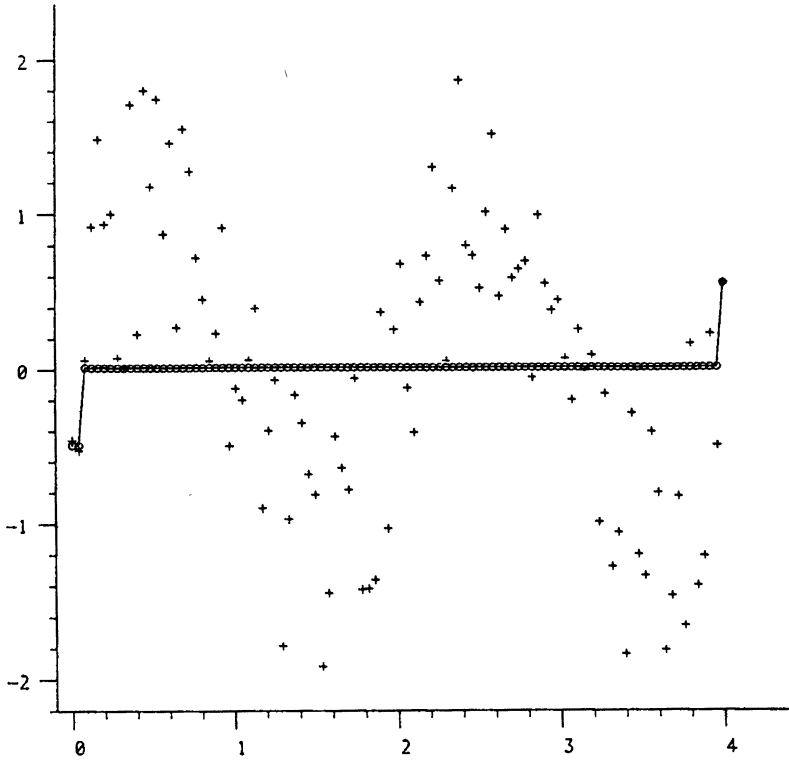


FIGURE 2. As in Figure 1, but  $k = 1, r = 1$

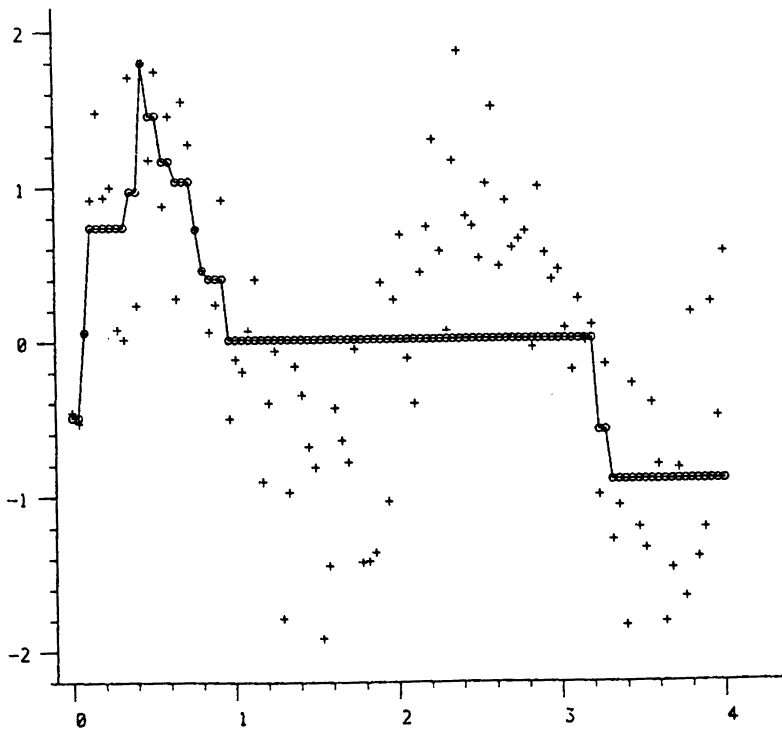
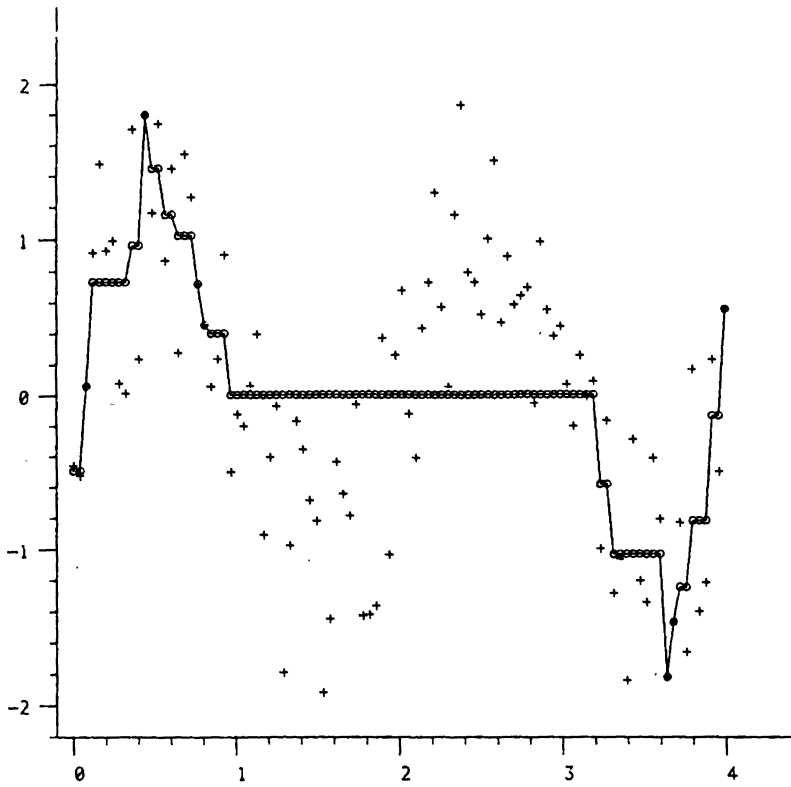
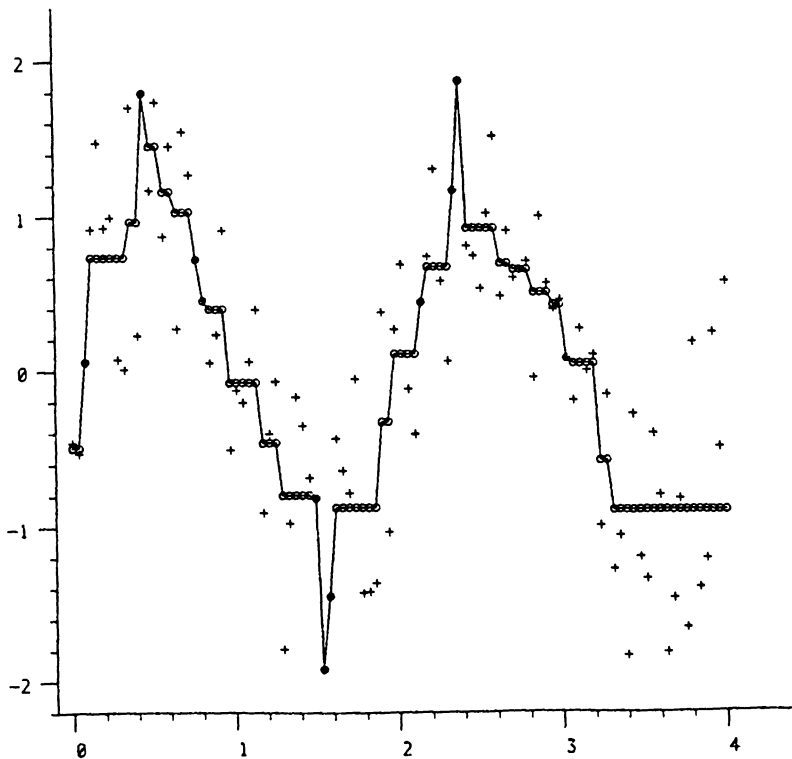
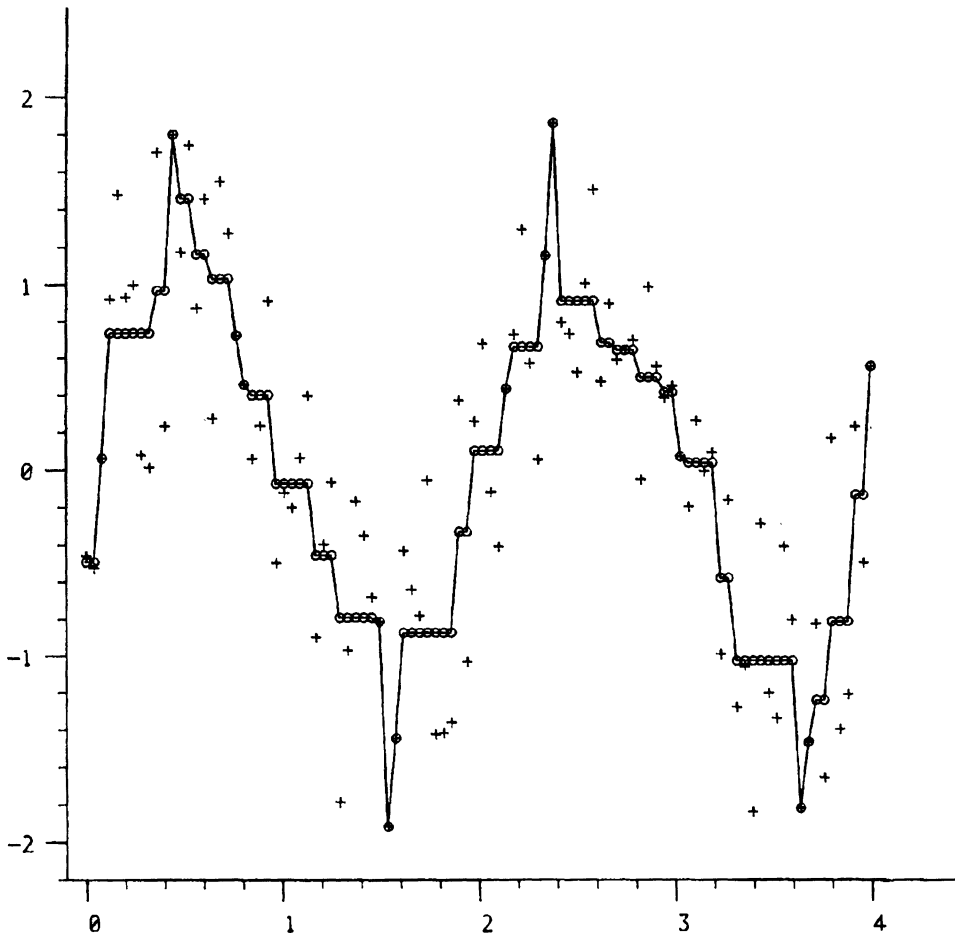


FIGURE 3. As in Figure 1, but  $k = 2, r = 1$

FIGURE 4. As in Figure 1, but  $k = 3$ ,  $r = 1$ FIGURE 5. As in Figure 1, but  $k = 4$ ,  $r = 1$

FIGURE 6. As in Figure 1, but  $k = 5$ ,  $r = 1$ 

## 6. CONCLUDING REMARKS

We have presented theory, algorithms, and numerical evidence for the problem of piecewise monotonic data smoothing. The algorithms are highly efficient in computer time and storage requirements, giving an exact answer in spite of the fact that  $O(n^k)$  local minima may occur in the calculation of an optimal approximation from  $\mathbb{Y}(k, n)$  to  $\varphi$ . Since the complexity of formulae (4.21) is  $O(k|\mathcal{X}|^2)$ , the main cost of the algorithms depends upon the particular form of the function  $h_i$  that appears in the definition of (2.1). Particular special procedures that take into account the form of  $h_i$  may be developed, as is noted at the end of the Appendix.

Our approach to data smoothing is different from many other smoothing approaches. We avoid the assumption that the underlying function  $f$  depends on a set of parameters and take the view that some useful smoothing should be possible if the data fails to satisfy a property that is usually obtained by  $f$ . We

suggest using the presented smoothing technique when the errors in the data are so large that they can be detected by the first divided differences. An advantage of this technique is that it yields piecewise monotonicity that occurs in a wide range of underlying functions  $f$ . Further, by considering the function that is defined by piecewise linear interpolation to the smoothed values, it is easy to show that it satisfies the same sign condition on the first derivative as the sign condition on the first divided differences that is employed for smoothing.

A disadvantage of the smoothing technique is that some significant errors in the data may be too small to be detected by the first divided differences. Also, the data errors are preserved at the integer variables  $\{t_j : j = 1, 2, \dots, k-1\}$  of an optimal approximation, owing to the interpolation relations (2.3). The criticism may be further directed at the presence of the ranges of constant values in the components of an optimal approximation.

However, approximations that give smoother values arise in the work of Demetriou [2], where a limit on the number of sign changes in the second divided differences is imposed.

Finally, quite a compact Fortran 77 code has been written for implementation (Demetriou [4]) that makes the presented smoothing technique easy and attractive to use. Since there is a strong need for data smoothing methods in many experimental laboratories, the code has been documented in order to be suitable for submission to an international software library.

#### APPENDIX. AN ALGORITHM FOR MONOTONIC APPROXIMATION BY A STRICTLY CONVEX DISTANCE FUNCTION

The purpose of this appendix is to present a general algorithm for monotonic approximation that is useful in this paper. Algorithm 3 below can be used to calculate the numbers  $\{\alpha(p, j) : j = p, p+1, \dots, q\}$  together with the monotonic components that occur in (2.1). The numbers  $\{\beta(p, j) : j = p, p+1, \dots, q\}$  can be calculated by applying the same algorithm to the data  $\{\varphi_q, \varphi_{q-1}, \dots, \varphi_p\}$ . Algorithm 3 generalizes the corresponding least squares version of Demetriou and Powell [5] and is justified by Demetriou [3]. Below, the auxiliary  $s$ -vector  $\psi$ ,  $1 \leq s \leq p - q + 1$ , is regarded as a representation of the best approximation that occurs in the definition of  $\alpha(p, j)$ , where each component  $\psi_i$  is repeated  $w_i$  times. One may consult Demetriou and Powell for an extended discussion on its implementation in the least squares case.

#### Algorithm 3 (The monotonic case).

*Step 0.* Set  $j = p$ ,  $s = 1$ ,  $\psi_s = \varphi_j$ ,  $w_s = 1$ , and  $\alpha(p, j) = 0$ .

*Step 1.* Go to Step 2 if  $j < q$ . Otherwise, either terminate because the numbers (2.1) have been calculated or, if required, go to Step 5 in order to calculate the best monotonic increasing approximation to  $\{\varphi_i : i = p, p+1, \dots, q\}$ .

*Step 2.* Increase  $j$  and  $s$  by one. Then set  $\psi_s = \varphi_j$ ,  $w_s = 1$ ,  $\alpha(p, j) = \alpha(p, j-1)$ , and  $t = j$ .

*Step 3.* If  $s = 1$  or  $\psi_s > \psi_{s-1}$ , replace  $\alpha(p, j)$  by  $\alpha(p, t-1) + h_t(\varphi_t - \psi_s) + h_{t+1}(\varphi_{t+1} - \psi_s) + \dots + h_j(\varphi_j - \psi_s)$  and go to Step 1.

*Step 4.* Set  $t = j + 1 - (w_s + w_{s-1})$ . Then reduce  $s$  by one, and let  $\psi_t$  be the value of  $\eta$  that minimizes the sum  $h_t(\varphi_t - \eta) + h_{t+1}(\varphi_{t+1} - \eta) + \cdots + h_j(\varphi_j - \eta)$ . Then replace  $w_s$  by  $w_s + w_{s-1}$  and go to Step 3.

*Step 5.* Set  $j = p$  and  $i = 1$ .

*Step 6.* Set  $\{y(k) = \psi(i) : k = j, \dots, j + w(i) - 1\}$  and  $j = j + w(i)$ . Terminate if  $i = s$  because  $\{y_k : k = p, p + 1, \dots, q\}$  is the required approximation. Otherwise increase  $i$  by 1 and branch to Step 6.  $\square$

We let  $\nu = q - p + 1$  and we show that Algorithm 3 terminates after  $O(\nu)$  applications of its steps. Since the integer  $j$  denotes the current data index, it indicates the number of times Step 1 is reached. Thus, Step 1 is reached exactly  $\nu$  times, and if  $j < q$ , it exits at Step 2. Since Step 2 is always preceded by Step 1, it is reached  $\nu - 1$  times and each time increases the values of  $j$  and  $s$  by one. Since  $s$  can be at most equal to  $\nu$ , and since each application of Step 4 reduces the value of  $s$  by one, Step 4 can be applied at most  $\nu - 1$  times. Step 3 is entered either from Step 2 or from Step 4, and therefore it can be reached at most  $2\nu - 2$  times. Finally, Steps 5 and 6 require  $O(\nu)$  operations in order to provide the best monotonic approximation to  $\{\varphi_i : i = p, p + 1, \dots, q\}$ . Therefore, Algorithm 3 is a finite procedure for calculating the numbers  $\{\alpha(p, j) : j = p, p + 1, \dots, q\}$  and the best monotonic increasing approximation to  $\{\varphi_i : i = p, p + 1, \dots, q\}$ .

We note that the main costs of Algorithm 3 occur in Step 4, where the best approximation by a constant to the data  $\{\varphi_i : i = t, t + 1, \dots, j\}$  needs to be calculated. Several methods for this calculation exist in the literature (see, for example, McCormick [6]).

#### ACKNOWLEDGMENTS

Part of this paper overlaps with work in my Ph.D. dissertation that has been based on ideas of M. J. D. Powell of Cambridge University. I am grateful to him for supervision, valuable advice, and generous support he offered for being my only source of information.

I would also like to thank the referee for helpful comments on a previous version of this paper.

#### BIBLIOGRAPHY

1. R. E. Barlow, D. J. Bartholomew, J. M. Bremner, and H. D. Brunk, *Statistical inference under order restrictions. The theory and application of isotonic regression*, Wiley, Chichester, 1980.
2. I. C. Demetriou, *Data smoothing by piecewise monotonic divided differences*, Ph.D. dissertation, University of Cambridge, England, 1985.
3. ———, *A characterization theorem for the discrete best monotonic approximation problem*, *Math. Comp.* **55** (1990), 191–195.
4. ———, *L2PMA: Fortran 77 subroutines for least squares piecewise monotonic data fitting*, Report of the University of Athens, Dept. of Economic Sciences, ICD-93-2, Greece, 1993.
5. I. C. Demetriou and M. J. D. Powell, *Least squares smoothing of univariate data to achieve piecewise monotonicity*, *IMA J. Numer. Anal.* **11** (1991), 411–432.

6. G. P. McCormick, *Nonlinear programming. Theory and applications*, Wiley, New York, 1983.
7. T. Robertson, F. T. Wright, and R. L. Dykstra, *Order restricted statistical inference*, Wiley, Chichester, 1988.

DEPARTMENT OF ECONOMIC SCIENCES, UNIVERSITY OF ATHENS, 8 PESMAZOGLOU STREET,  
105 59 ATHENS, GREECE  
*E-mail address:* demetri@aub.ariadne-t.gr