USER GUIDE FOR ALGORITH XXX: L2WPMA

IOANNIS C. DEMETRIOU
University of Athens

L2WPMA is a package of Fortran 77 subroutines that calculates a weighted least squares piecewise monotonic approximation to univariate data contaminated by random errors [3]. This report provides an overview of the subroutines and gives instructions for using the package that specify its interface with the calling program.

Categories and Subject Descriptors: G.1.2 [**Numerical analysis**]: Approximation - least squares approximation - smoothing; G.2.1 [**Combinatorics**]: Recurrences

General Terms: Algorithms
Additional Key Words and Phrases: approximation, data smoothing, divided difference, dynamic programming, fitting, turning point

1. PROBLEM DEFINITION AND OUTLINE OF THE METHOD

L2WPMA is a package of Fortran subroutines that is presented by [3]. It calculates a weighted least squares piecewise monotonic approximation to n univariate data contaminated by random errors, which is defined as follows. If a real function $f(x)$ is measured at the abscissae $x_1 < x_2 < \cdots < x_n$ and the measurements $\{\phi_i \approx f(x_i) : i = 1, 2, \ldots, n\}$ contain large uncorrelated random errors, then L2WPMA can be used to calculate values for $\{y_i : i = 1, 2, \ldots, n\}$ that minimize the weighted sum of the squares of the errors

$$\Phi(y) = \sum_{i=1} w_i(\phi_i - y_i)^2 \qquad (1)$$

so that the sequence $\{y_{i+1} - y_i : i = 1, 2, \ldots, n-1\}$ has at most $k - 1$ sign changes, where k is a given positive integer smaller than n. Without loss of generality, we assume that the first nonzero difference $y_{i+1} - y_i$ is positive. So the constraints are [4]

$$\left.\begin{array}{l} y_{t_j} \le y_{t_j+1} \le \cdots \le y_{t_{j+1}}, \; j \text{ even} \\ y_{t_j} \ge y_{t_j+1} \ge \cdots \ge y_{t_{j+1}}, \; j \text{ odd} \end{array}\right\}, \qquad (2)$$

Unit of Mathematics and Informatics, Department of Economics, University of Athens, 8 Pesmazoglou Street, 105 59 Athens, Greece, e-mail: demetri@econ.uoa.gr

where $\{t_j : j = 1, 2, \ldots, k-1\}$ are integers that satisfy the conditions

$$1 = t_0 \le t_1 \le \cdots \le t_k = n; \tag{3}$$

and where the numbers (weights) w_i satisfy the inequalities $w_i > 0$, $i = 1, 2, \ldots, n$. While k is provided by the user, $\{t_j : j = 1, 2, \ldots, k-1\}$ are variables of the minimization calculation together with $\{y_i : i = 1, 2, \ldots, n\}$. It is convenient to the subsequent presentation to consider $\{\phi_i : i = 1, 2, \ldots, n\}$ and $\{y_i : i = 1, 2, \ldots, n\}$ as the components of vectors ϕ and y in R^n respectively.

L2WPMA consists of five Fortran subroutines for the calculation of an optimal fit to ϕ. The user may specify whether the first monotonic section is increasing or decreasing. The software package allows a monotonic section to degenerate to a single component. The underlying method is described by [2] except that L2WPMA allows the data have positive weights and at the end of the calculation it provides a spline representation of the fit and the corresponding Lagrange multipliers. The entry point of our package, which also names the software package, is subroutine L2WPMA. Section 2 specifies the interface of the software with the calling program. Section 3 presents the purpose of each subroutine. Section 4 presents output of a simple example. Section 5 gives information for further documentation.

In order to explain the purpose of the arguments of the user interface, we outline below the method of calculation, but for details one may consult [2]. For positive integers p and q, we let

$$\alpha(p, q) = \min_{y_p \le y_{p+1} \le \cdots \le y_q} \sum_{i=p} w_i(\phi_i - y_i)^2; \qquad 1 \le p \le q \le n, \tag{4}$$

$$\beta(p, q) = \min_{y_p \ge y_{p+1} \ge \cdots \ge y_q} \sum_{i=p} w_i(\phi_i - y_i)^2; \qquad 1 \le p \le q \le n, \tag{5}$$

and for any integers $m \in [1, k]$ and $t \in [1, n]$ we let

$$G(m, t) = \{\min_{z \in R^t} \sum_{i=1} w_i(\phi_i - z_i)^2; \ \ z \text{ has } m \text{ monotonic sections}\}.$$

In order to calculate $G(k, n)$, which is the least value of (1), we begin the calculation from $G(1, t) = \alpha(1, t)$, for $t = 1, 2, \ldots, n$, and proceed by applying the dynamic programming formulae

$$G(m, t) = \begin{cases} \min\limits_{s \in [\max\{\vartheta(m), \tau(m-2, t)\}, t] \cap L} [G(m-1, s) + \alpha(s, t)], & m \text{ odd}, \\ \min\limits_{s \in [\max\{\vartheta(m), \tau(m-2, t)\}, t] \cap U} [G(m-1, s) + \beta(s, t)], & m \text{ even}, \end{cases} \tag{6}$$

for all $m \in [2, k]$, while t increments in L or U, where: L and U are the sets of the indices of local minima and local maxima of the data respectively (for the definitions of L and U see [4]); $\tau(m, t)$ is the value of s that minimizes expression (6), for each value of m and t; and, $\vartheta(m)$ is the greatest value of $\{\tau(m, \ell) : \ell < t\}$ that has already been calculated while $\tau(m, \ell)$ is chosen as small as possible. At the end of the process $m = k$ occurs, the value $\tau(k, n)$ is the integer t_{k-1} and we obtain the sequence of optimal values $\{t_j : j = 1, 2, \ldots, k-1\}$ by the backward formula

$$t_k = n; \; t_{m-1} = \tau(m, t_m), \quad m = k, k-1, \ldots, 2. \tag{7}$$

Then, the components of an optimal fit are monotonic increasing on $[1, t_1]$ and on $[t_j, t_{j+1}]$ for even j in $[1, k-1]$ and decreasing on $[t_j, t_{j+1}]$ for odd j in $[1, k-1]$. In Section 4 a numerical example demonstrates the derivation of the t_is for $k > 2$, by means of (7).

L2WPMA provides also the Lagrange multipliers $\{\lambda_i : i = 2, 3, \ldots, n\}$ (although they are not required for obtaining the optimal fit). They are defined as follows. Having obtained the optimal sequence of integers $\{t_i : i = 2, 3, \ldots, k-1\}$ and the associated optimal fit y, the Karush-Kuhn-Tucker conditions for the problem that minimizes (1) subject to the constraints (2) state that the equation

$$\mathrm{grad}\Phi(y) = \sum_{i \in A} \lambda_i (e^{i-1} - e^i) \tag{8}$$

holds, where A is the subset $A = \{i : y_{i-1} - y_i = 0\}$ of the constraint indices $\{2, 3, \ldots, n\}$, e^i is the ith coordinate vector in R^n and $\mathrm{grad}\Phi(y)$ is the gradient of $\Phi(y)$; and, $\{\lambda_i : i \in A\}$ are nonnegative, when $i \in [2, t_1] \cap A$ and $i \in [t_j + 1, t_{j+1}] \cap A$ for j even, and nonpositive, when $i \in [t_j + 1, t_{j+1}] \cap A$ for j odd. Further, we define $\lambda_i = 0$ for all integers i in $[2, n] \setminus A$, so that λ is a $(n-1)$-vector.

Finally, we may express the optimal fit y in the form of first-order B-splines (see [1: p.89]). To be specific, we represent y by a triple $(\kappa; \omega; \eta)$. Here κ is a positive integer and ω and η are vectors in R^κ, where the components of ω are positive integers whose sum is n. This triple denotes the vector $y = y(\kappa; \omega; \eta) \in R^n$ that has ω_1 components equal to η_1, ω_2 components equal to η_2 and so on up to ω_κ components equal to η_κ. Hence we define the knots $\xi_1 = x_1$, $\xi_2 = x_{\omega_1+1}$, $\xi_3 = x_{\omega_1+\omega_2+1}$ and so on up to $\xi_\kappa = x_{\omega_1+\omega_2+\cdots+\omega_{\kappa-1}+1}$, we let $\eta_1 = y_1$, $\eta_2 = y_{\omega_1+1}$, $\eta_3 = y_{\omega_1+\omega_2+1}$ and so on up to $\eta_\kappa = y_{\omega_1+\omega_2+\cdots+\omega_{\kappa-1}+1}$, and we obtain the spline representation $s(x)$ of vector y by

$$s(x) = \sum_{j=1}^{\kappa} \eta_j B_j(x); \qquad x_1 \le x \le x_n; \tag{9}$$

where $B_j(x) = 1$, if $\xi_j \le x < \xi_{j+1}$, and $B_j(x) = 0$ otherwise. L2WPMA at the end of the calculation provides the data indices of the knots, so the user may obtain the sequences $\{\xi_i : i = 1, 2, \ldots, \kappa\}$ and $\{\eta_i : i = 1, 2, \ldots, \kappa\}$ (see argument IAKN of subroutine L2WPMA in Section 2).

2. USER INTERFACE

The main subroutine that provides interface to the user is declared by

```
  SUBROUTINE L2WPMA(I1, N, X, F, WF, MODEWF, KSECTN, IORDER,
+Y, WY, NK, IAKN, NACT, IACT, PAR, ITAU, ITHETA, MODE, SS, G, RG,
+LOWER, IUPPER, INDX, FT, WFT, FTNEG, WY1, Z, WZ, IW, IAKNW)
```

Subroutine L2WPMA implements Algorithm 1 of [2] with certain enhancements that provide an optimal fit to ϕ. The calculation starts by calling subroutine

TRIVIA (which is referred to in Section 3) in order to check on certain trivial cases that may cause termination of the smoothing process and ends by providing the knots of the spline representation (9) of the optimal fit and the associated Lagrange multipliers.

The purpose of each argument of L2WPMA follows. We allow the range of the data indices be [I1,N] instead of $[1;n]$ and, henceforth, we shall refer to the formulae of Section 1 by following this convention. All subroutines referred to in the following list are explained in Section 3.

INPUT (they must be set by the user)

I1 Integer variable, lower data index, I1=1.

N Integer variable, upper data index, corresponds to n, $N \geq I1$.

X(I1:N) Real array of the abscissae x_i, i=I1,I1+1,...,N. The use of X is optional (see, MODEWF below).

F(I1:N) Real array of data values ϕ_i, i=I1,I1+1...,N.

WF(I1:N) Real array of positive weights w_i, i=I1,I1+1,...N, where w_i is associated with ϕ_i. The use of WF is optional as we explain in MODEWF below.

MODEWF Integer variable that specifies the weights WF(.) as follows:

MODEWF=0 Specifying that WF(.) is supplied by the user (the default value).

MODEWF=1 The components of WF(.) are set to unity by the program, WF(i)=1, i=I1,I1+1,...,N.

MODEWF=2 The components of WF(.) are set automatically to

$$WF(i) = \sigma_i / \sum_{i=I1} \sigma_i, \quad i = I1, \ldots, N, \tag{10}$$

where

$$\sigma_i = \begin{cases} 1/\nabla x_i, & i = I1+1, \ldots, N \\ \sum_{i=I1+1}^{N} \sigma_i/(N - I1), & i = I1 \end{cases} \tag{11}$$

and $\nabla x_i = x_i - x_{i-1}$.

MODEWF=3 The components of WF(.) are defined as in MODEWF=2, but

$$\sigma_i = \begin{cases} 1/\Delta x_i, & i = I1, \ldots, N-1 \\ \sum_{i=I1}^{N-1} \sigma_i/(N - I1), & i = N \end{cases}, \tag{12}$$

where $\Delta x_i = x_{i+1} - x_i$.

MODEWF=4 The components of WF(.) are set automatically to

$$WF(i) = \begin{cases} \Delta x_{I1}/2, & i = I1 \\ (\Delta x_{i-1} + \Delta x_i)/2, & i = I1+1, \ldots, N-1 \\ \Delta x_{N-1}/2, & i = N \end{cases}. \tag{13}$$

MODEWF=5 Only the data values ϕ_i, i =I1,I1+1,...,N, are given. The abscissae are set automatically to X(i)=i, i=I1,I1+1,...,N, and the weights are set automatically to WF(i)=1, i=I1,I1+1,...,N.

KSECTN Integer variable that specifies the number of monotonic sections, corresponds to k, $1 \le$ KSECTN $\le$ N $-$ I1.

IORDER Integer variable, whose default value is IORDER=0 and specifies that the first monotonic section is increasing. If IORDER= -1, then the first monotonic section is decreasing.

OUTPUT

Y(I1:N) Real array containing an optimal fit to F(.) at the end of the calculation. It corresponds to vector y.

WY(I1:N) Real array containing the weights of the optimal fit at the end of the calculation.

NK Integer variable that is set to the number of knots of the spline representation of the optimal fit as follows: If $\omega > 1$, then NK= $\kappa + 1$, otherwise NK= κ, where ω and κ are defined just before formula (9). Therefore its value is in the range [I1,N].

IAKN(I1:NK) Integer array containing the data indices of the knots of the spline representation of the optimal fit, where we set IAKN(NK)=N, if $\omega > 1$ (note that IAKN(NK)=N, whenever $\omega = 1$). Therefore the knots are ξ_i =X(IAKN(i)), i=I1,I1+1,...,NK, where $\xi_1 = x_1$ and $\xi_{NK} = x_n$. Similarly, the coefficients in expression (9) are η_i =Y(IAKN(i)), i=I1,I1+1,...,NK, where $\eta_1 = y_1$ and $\eta_{NK} = y_n$.

NACT Integer variable, the number of constraints that are satisfied as equations at the end of the calculation, $0 \le$ NACT $\le$ N $-$ I1.

IACT(1:NACT) Integer array that provides the indices of the constraints satisfied as equations at termination. It corresponds to set A.

PAR(I1+1:N) Real array containing the Lagrange parameters associated with the constraints that are satisfied as equations by Y(.) at the end of the calculation. Specifically, these parameters are PAR(IACT(k)), k=1,2,...,NACT, and correspond to λ_k, $k \in A$.

ITAU(0:KSECTN,I1:(N $-$ I1+1)/2+1) Integer array such that ITAU($m; j$) is the index of the ($m - 1$)th extremum of an optimal fit with m monotonic sections to the first LOWER(j) or IUPPER(j) data, where $1 \le m \le$ KSECTN, I1 $\le j \le$ (N $-$ I1+1)/2 +1, and the arrays LOWER and IUPPER are defined in the "working space" section below. ITAU($m; j$) corresponds to $\tau(m; j)$.

ITHETA(0:KSECTN) Integer array that holds the values of the sequence $\theta(.)$ employed in formulae (6). At the end of the calculation ITHETA holds the optimal values of the integer variables $\{t_j : j = 0, 1, \ldots, \text{KSECTN}\}$.

MODE Integer variable indicating the status of subroutine termination as follows:

MODE=0 Unsuccessful return of L2WPMA, because KSECTN, the number of monotonic sections, is smaller than one.

MODE=1 Successful return of L2WPMA.

MODE=2 Successful return of L2WPMA due to $|U|+|L| \le$ KSECTN+1. The data itself provides the required optimal fit.

WORKING SPACE

SS(I1:N) Real array, argument of subroutine L2WMON (see Section 3), that keeps either the values $\{\alpha(p,j) : j = p, p+1, \ldots, q\}$ or the values $\{\beta(j,q) : j = p, p+1, \ldots, q\}$ as they are defined by formulae (4) and (5). SS(N) contains the value of the objective function (1) at the end of the calculation.

G(0:KSECTN,I1:(N − I1+1)/2+1) Real array that keeps in $G(m,j)$ the weighted sum of squares of residuals of the fit associated with $ITAU(m,j)$. G(.,.) is defined in Section 1.

RG(I1:(N − I1+1)/2+1) Real array that provides temporary storage for the sum included in the brackets of formula (6).

LOWER(I1:(N − I1+1)/2+1) Integer array that keeps the indices of the local minima of F(.) denoted by L in Section 1. It is formed by subroutine XTREMA (see Section 3).

IUPPER(I1:(N − I1+1)/2+1) Integer array that keeps the indices of the local maxima of F(.) denoted by U in Section 1. It is formed by subroutine XTREMA.

INDX(I1:N) Integer array that gives the data index of a local minimum or a local maximum of F(.), such that INDX(LOWER(i))=i and INDX(IUPPER(i))=i.

FT(I1:N), WFT(I1:N), FTNEG(I1:N), WY1(I1:N) Real arrays that are explained in the comments of subroutine L2WPMA.

Z(1:N − I1+1), WZ(1:N − I1+1), IW(1:N − I1+1), IAKNW(1,N − I1+1) Arrays that provide working space for subroutine L2WMON.

Subroutine L2WPMA is a finite procedure. Unsuccessful return (MODE=0) is caused only if KSECTN<1. Otherwise its return is successful, Y(.) satisfies the constraints (2), ITHETA(.) satisfies the conditions (3) and MODE is set to 1 or 2. The termination status is explained by certain messages.

The data may be weighted in one of five ways depending on the value assigned to MODEWF. When each data point (x_i, ϕ_i) consists of an average of K, say, observations at x_i, one may wish to set the weight value of this point to the inverse square of the standard deviation of these K observations, in which case MODEWF has to be set to 0. If the weights are equal to 1, then we set MODEWF=1 and the user need only supply X and F. When we set MODEWF=2, the weights are defined by (10) automatically so as to reflect possible differences in the abscissae spacing. Specifically, the closer is X(i) to X(i − 1), the larger is the value we assign to the weight WF(i), according to formulae (10). Moreover, in order to define WF(I1) we require that the weights are normalized so that their sum equals 1. Similarly for the case MODEWF=3, but formula (12) is used instead of (11). If the abscissae are equally spaced, then the options MODEWF=2 and MODEWF=3 imply WF(i)=1/(N − I1+1), i=I1,I1+1,...,N, and the value of (1) gives an estimator of the variance of the population that provided the sample ϕ_i, i=I1,I1+1,...,N. Further, since the length of [X(i − 1),X(i)] is a measure of our information over the interval, a typical choice of weights (see [1: p.220]) is provided by formulae (13) that is activated by setting MODEWF=4. Finally, if only the data values ϕ_i, i=I1,I1+1,...,N, are available, where we assume that they have been derived with the natural order {I1,I1+1,...,N}, then X and WF are defined automatically by setting MODEWF=5.

3. THE PURPOSE OF THE SUBROUTINES

The Fortran software for the calculation described by [3] consists of five subroutines. Common blocks and private array storage are avoided. The working space is directed through the argument list of each subroutine.

Single and double precision versions have been developed for practical use. All subroutines begin with comments that explain the input and output arguments, the working space and the method followed. The entire code consists of 1455 Fortran 77 lines including comments. The number of lines of code for each subroutine is: L2WPMA 740, TRIVIA 183, XTREMA 137, L2WMON 250 and MESSGW 145. The purpose of each subroutine is as follows.

Subroutine L2WPMA Interface to the user. Given I1, N, X(.), F(.), WF(.) and KSECTN, it calculates a best weighted least squares approximation Y(.) to F(.) as outlined in [3]. The use of X and WF is optional. In addition, the user may specify the order of the first monotonic section and may allow WF to be calculated automatically from the abscissae spacing. The underlying method is described by [2], except that L2WPMA employs weights and at the end of the calculation it provides the knots of the spline representation (9) of the solution and the corresponding Lagrange multipliers. It calls subroutines TRIVIA, XTREMA, L2WMON and MESSGW. The complexity of L2WPMA is $O(n|U| + k|U|^2)$, when $k \geq 3$, where n is the number of data points and $|U|$ is the number of the local maxima of the data, which is always bounded by $n/2$. This complexity reduces to $O(n)$ when $k = 1$ or $k = 2$.

Subroutine TRIVIA In the beginning of subroutine L2WPMA, subroutine TRIVIA is called to check on the following trivial cases. If KSECTN< 1 then a MODE=0 return of L2WPMA is caused. If KSECTN=1 then TRIVIA returns to subroutine L2WPMA and subroutine L2WMON is called to calculate the best monotonic increasing or decreasing approximation to the data. If KSECTN $\geq$ 1 and N=I1 or $F(I1) \leq F(I1+1) \leq \cdots \leq F(N)$ or $F(I1) \geq F(I1+1) \geq \cdots \geq F(N)$ then the data satisfies the constraints, thus it is the required optimal approximation, and on return to subroutine L2WPMA termination occurs. The complexity of subroutine TRIVIA is $O(n)$, where n is the number of data.

Subroutine L2WMON Given F(.), WF(.) and integers L1 and LN such that $I1 \leq L1 \leq LN \leq N$, subroutine L2WMON calculates the values $\alpha(L1,i)$, i=L1,L1+1,..., LN, where $\alpha(.,.)$ is defined by (4). The return of the solution components that occur in $\alpha(L1,LN)$ depends on a flag, whose value is set by the calling subroutine L2WPMA. L2WMON for this calculation implements a modification of Algorithm 1 of [4] that allows a weight to each ϕ_i. Its complexity is $O(n)$, where $n = LN - L1+1$. The values $\beta(i,LN)$, i=L1,L1+1,...,LN, defined by formula (5), can be calculated by applying L2WMON to the data FT(i), i=L1,L1+1,...,LN, associated with the weights WFT(i), i=L1,I1+1,...,LN, where FT and WFT are arrays that keep the elements of F and WF in reverse order. Furthermore, L2WMON, together with the monotonic components that occur at $\alpha(L1,LN)$, provides the knots of the corresponding spline representation and the Lagrange multipliers associated with the solution.

Subroutine XTREMA It forms the sets L and U, LOWER and IUPPER respectively, that hold the indices of the local minima and the indices of the local maxima of the data $\{\phi_i : i=I1, I1+1, \dots, N\}$ as described by [4]. Its complexity is $O(n)$, where n is the number of the data.

Subroutine MESSGW It contains certain messages associated with the operation of subroutines L2WPMA and TRIVIA.

4. OUTPUT FROM A TEST EXAMPLE

This section presents an example of the use of the package L2WPMA. The calculations were performed on a personal computer with an Intel 733 MHz processor (32 bits word length), operating with MS Windows 98 and using the Compaq Visual FORTRAN 6.1 compiler in single precision arithmetic. A simple driver program of L2WPMA uses I1=1, N=14, $\{(x_i;\phi_i) : i = 1;2;\dots;N\}$, where $\{x_i = i : i = 1;2;\dots;N\}$, $\phi_1 = -0.1$, $\phi_2 = 0.71$, $\phi_3 = 0.69$, $\phi_4 = 0.87$, $\phi_5 = -1$, $\phi_6 = -1.11$, $\phi_7 = \phi_8 = 1$, $\phi_9 = \phi_{10} = -1$, $\phi_{11} = 0.68$, $\phi_{12} = 0.73$, $\phi_{13} = 0.70$ and $\phi_{14} = 0.50$, and $\{w_i = 1 : i = 1;2;\dots;N\}$. We applied the driver program by requiring KSECTN=4 monotonic sections and L2WPMA carried out the calculation terminating with the output displayed in Fig. 1. Due to the (even) value of KSECTN, the computer program formed the sets $L = \{1;3;6;9;14\}$ and $U = \{2;4;7;12\}$, while at termination, the KSECTN$\times|U|$ array τ=ITAU is

$$\tau = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 4 & 7 & 7 \\ 1 & 3 & 6 & 9 & 0 \\ 0 & 0 & 0 & 0 & 7 \end{pmatrix} \tag{14}$$

The integers t_i=ITHETA(i), $i = 0;1;\dots;$KSECTN, presented in Fig. 1, are derived by combining (7) and (14) as follows. Initially, we have $t_4 = 14$, which is the 5th element of L. In view of (7), we obtain $t_3 = \tau(4;5) = 7$, which is the 3rd element of U, and subsequently $t_2 = \tau(3;3) = 6$, which is the 3rd element of L, $t_1 = \tau(2;3) = 4$, which is the 2nd element of U, and finally $t_0 = \tau(1;2) = 1$. Associated with the t_is, let $y \in R^{14}$ be the fit to ϕ that has $k = 4$ monotonic sections, whose components are shown in the column labeled '(Y)' in Fig. 1. We see that y_i, $i = 1;2;\dots;14$, satisfy the constraints $y_1 \le y_2 \le y_3 \le y_4$, $y_4 \ge y_5 \ge y_6$, $y_6 \le y_7$ and $y_7 \ge y_8 \ge \cdots \ge y_{14}$, and we are going to prove that y does minimize (1) subject to these constraints. Indeed, first due to formula (8), we obtain the identities $2w_1(y_1 - \phi_1) = -\lambda_2$, $2w_2(y_2 - \phi_2) = \lambda_2 - \lambda_3; \dots; 2w_{13}(y_{13} - \phi_{13}) = \lambda_{14} - \lambda_{13}$ and $2w_{14}(y_{14} - \phi_{14}) = \lambda_{14}$. Then, in view of the values (see column labeled 'Y' in Fig. 1) $y_1 = \phi_1$, $y_2 = y_3 = 0.7$, $y_i = \phi_i$, for $i = 4;5;\dots;8$, and $y_9 = y_{10} = \cdots = y_{14} = 0.1017$, it is straightforward to verify (the actual formulae are presented in Section 2 of [3]) that the numbers shown in the column labeled '(PAR)' in Fig. 1 are the Lagrange multipliers $\lambda_2 = 0$, $\lambda_3 = 0.02$, $\lambda_i = 0$, for $i = 4;5;\dots;8$, $\lambda_{10} = -2.20$, $\lambda_{11} = -4.41$, $\lambda_{12} = -3.25$, $\lambda_{13} = -1.99$ and $\lambda_{14} = -0.80$. Because $\lambda_i \ge 0$, for $i \in [2;t_1] \cup [t_2+1;t_3]$, and $\lambda_i \le 0$, for $i \in [t_1+1;t_2] \cup [t_3+1;14]$, the Karush-Kuhn-Tucker conditions for the solution of the quadratic programming problem stated above are satisfied.

Next, we applied the driver program to the same data as those in Fig. 1 for KSECTN=1,2,..., and the corresponding smoothed values are presented in Table 1 under the headings $k = 1; k = 2;\dots; k \ge 8$. The first three columns of Table 1 present the vectors x, ϕ and w, while the last row gives the value of the objective function (1) at each approximation. Certain features of the optimal approximation are demonstrated by this example. They are that as k increases, most of the extrema of the fits are preserved, the ranges of constant components are reduced, some monotonic sections may degenerate to a single point (as when $k = 7$, where the corresponding approximation consists of 6 monotonic sections) and ϕ is the optimal fit whenever KSECTN ≥ 8, because ϕ satisfies the constraints.

Table 2 presents output from an experiment similar to that in Table 1, except that the abscissae take the nonuniformly spaced values of column 2. We applied the driver program with MODEWF=2 and the weights presented in column 4 were generated automatically due to (10) giving WF(1)=0.0714, WF(2)=0.0116 and so on up to WF(14)=0.7244. First we see that the best weighted approximations of Table 2 exhibit features broadly similar to the approximations in Table 1. In particular, let $y \in R^n$ be the best weighted approximation when $k = 4$ in Table 2, which is associated with $t_1 = 7; t_2 = 9$ and $t_3 = 12$, obtained from

$$\tau = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 4 & 7 & 12 \\ 1 & 3 & 6 & 9 & 0 \\ 0 & 0 & 0 & 0 & 7 \end{pmatrix} \qquad (15)$$

by analogy with (14). Consequently the components of y satisfy the constraints $y_1 \leq y_2 \leq \cdots \leq y_7$, $y_7 \geq y_8 \geq y_9$, $y_9 \leq y_{10} \leq y_{11} \leq y_{12}$ and $y_{12} \geq y_{13} \geq y_{14}$, and by arguments similar to those in the paragraph following (14), we can show that the values $y_1 = y_2 = \cdots = y_6 = -0.1298$ and $y_i = \phi_i; i = 7, 8, \ldots, 14$, minimize (1) subject to these constraints. Indeed, it is straightforward to verify that (8) is satisfied with $\lambda_2 = 0.0043; \lambda_3 = 0.0237; \lambda_4 = 0.0292; \lambda_5 = 0.0463; \lambda_6 = 0.0067$, and $\lambda_i = 0; i = 7, 8, \ldots, 14$ and since $\lambda_i \geq 0$, $i \in [2, t_1] \cup [t_2 + 1, t_3]$; and $\lambda_i \leq 0$, $i \in [t_1 + 1, t_2] \cup [t_3 + 1, 14]$, the Karush-Kuhn-Tucker conditions for the solution of this quadratic programming problem are satisfied. We conclude that the differences between the corresponding approximations of Tables 1 and 2 are the result of the use of the weights.

Figs 2 and 3 illustrate the best approximations obtained by applying subroutine L2WPMA to certain data sets. Fig. 2 presents an optimal fit with $k = 6$ monotonic sections, which may be viewed as the result of an intermediate step of a calculation that may further improve the fit. Here, a disadvantage of the smoothing technique is shown at the rightmost monotonic section of this fit, where the data errors are too small to be detected by the first differences. Further, the particular k of Fig. 3 allows L2WPMA to achieve the piecewise monotonicity property it sets out to achieve and, generally, any degree of undulation in the data can be accommodated by choosing a suitable k.

<Tables 1 and 2 and Figs 1, 2 and 3 belong to this section>

5. DOCUMENTATION

Distribution material that includes single and double precision versions of the code, driver programs, numerical examples with output in order to help new users of the method and documentation is available in ASCII form accompanying [3].

The documentation, namely file INSTDE04.txt of the distribution material, includes description of the driver programs, comments on the output of several test examples that help the usage of L2WPMA, provides technical details about installation, compilation, linking, running and testing of the Fortran codes, and remarks on the Fortran listings.

REFERENCES

[1] DE BOOR C. 2001. A Practical Guide to Splines. Revised Edition. Springer-Verlag, Applied Mathematical Sciences, Vol. 27, NY.

[2] DEMETRIOU I.C. 1995. Discrete piecewise monotonic approximation by a strictly convex distance function. Mathematics of Computation, Vol. 64, No 209, pp. 157-180.

[3] DEMETRIOU I.C. 2006. ALgorithm XXX: L2WPMA, a Fortran 77 Package for Weighted Least Squares Piecewise Monotonic Data Approximation. To appear in ACM Transactions on Mathematical Software.

[4] DEMETRIOU I.C. AND POWELL M.J.D. 1991. Least squares smoothing of univariate data to achieve piecewise monotonicity. IMA J. of Numer. Anal., Vol. 11, No 3, pp. 411-432.

Received Month Year; revised Month Year; accepted Month Year

Table 1 Best approximations to the data *x*-*φ*-*w*, where the weights *w* are set equal to unity. The local maxima of an approximation are displayed with bold characters and the local minima with underlined ones. The best approximation for *k*=6 coincides with that when *k*=7, and for every *k*≥8, it coincides with the data.

Data			Best approximations with *k* monotonic sections							
x	*φ*	*w*	*k*=1	*k*=2	*k*=3	*k*=4	*k*=5	*k*=6	*k*=7	*k*≥8
1.00	–0.10	1	–0.1000	–0.1000	–0.1000	–0.1000	–0.1000	–0.1000	–0.1000	–0.1000
2.00	0.71	1	0.0178	0.0320	0.0320	0.7000	0.7000	0.7000	0.7000	**0.7100**
3.00	0.69	1	0.0178	0.0320	0.0320	0.7000	0.7000	0.7000	0.7000	$\underline{0.6900}$
4.00	0.87	1	0.0178	0.0320	0.0320	**0.8700**	**0.8700**	**0.8700**	**0.8700**	**0.8700**
5.00	–1.00	1	0.0178	0.0320	0.0320	–1.0000	–1.0000	–1.0000	–1.0000	–1.0000
6.00	–1.11	1	0.0178	0.0320	0.0320	$\underline{-1.1100}$	$\underline{-1.1100}$	$\underline{-1.1100}$	$\underline{-1.1100}$	$\underline{-1.1100}$
7.00	1.00	1	0.0178	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
8.00	1.00	1	0.0178	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
9.00	–1.00	1	0.0178	0.1017	$\underline{-1.0000}$	0.1017	$\underline{-1.0000}$	$\underline{-1.0000}$	$\underline{-1.0000}$	$\underline{-1.0000}$
10.00	–1.00	1	0.0178	0.1017	–1.0000	0.1017	–1.0000	–1.0000	–1.0000	–1.0000
11.00	0.68	1	0.6525	0.1017	0.6525	0.1017	0.6525	0.6800	0.6800	0.6800
12.00	0.73	1	0.6525	0.1017	0.6525	0.1017	0.6525	**0.7300**	**0.7300**	**0.7300**
13.00	0.70	1	0.6525	0.1017	0.6525	0.1017	0.6525	0.7000	0.7000	0.7000
14.00	0.50	1	0.6525	0.1017	0.6525	0.1017	0.6525	0.5000	0.5000	0.5000
Sum of squares of residuals (1):			7.9986	7.6374	3.9964	3.6735	0.0325	0.0002	0.0002	0.0000

Table 2 Best weighted approximations to the data *x*-*φ*-*w*, where the weights *w* are calculated by formulae (10). The notation of Table 1 is used.

Data				Best weighted approximations with *k* monotonic sections							
i	*x*	*φ*	*w*	*k*=1	*k*=2	*k*=3	*k*=4	*k*=5	*k*=6	*k*=7	*k*≥8
1	0.0021	–0.10	0.0714	–0.2341	–0.2341	–0.1298	–0.1298	–0.1000	–0.1000	–0.1000	–0.1000
2	0.7338	0.71	0.0116	–0.2341	–0.2341	–0.1298	–0.1298	0.7055	0.7055	0.7055	**0.7100**
3	3.2849	0.69	0.0033	–0.2341	–0.2341	–0.1298	–0.1298	0.7055	0.7055	0.7055	$\underline{0.6900}$
4	4.2757	0.87	0.0086	–0.2341	–0.2341	–0.1298	–0.1298	**0.8700**	**0.8700**	**0.8700**	**0.8700**
5	4.6491	–1.00	0.0227	–0.2341	–0.2341	–0.1298	–0.1298	–1.0000	–1.0000	–1.0000	–1.0000
6	7.1108	–1.11	0.0034	–0.2341	–0.2341	–0.1298	–0.1298	$\underline{-1.1100}$	$\underline{-1.1100}$	$\underline{-1.1100}$	$\underline{-1.1100}$
7	8.2084	1.00	0.0077	–0.2341	–0.2341	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
8	8.7843	1.00	0.0147	–0.2341	–0.2341	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
9	9.2829	–1.00	0.0170	–0.2341	–0.2341	$\underline{-1.0000}$	$\underline{-1.0000}$	$\underline{-1.0000}$	$\underline{-1.0000}$	$\underline{-1.0000}$	$\underline{-1.0000}$
10	9.5207	–1.00	0.0356	–0.2341	–0.2341	–1.0000	–1.0000	–1.0000	–1.0000	–1.0000	–1.0000
11	9.6939	0.68	0.0489	0.5188	**0.6800**	0.5188	0.6800	0.5188	0.6800	0.6800	0.6800
12	11.1139	0.73	0.0060	0.5188	0.7300	0.5188	**0.7300**	0.5188	**0.7300**	**0.7300**	**0.7300**
13	11.4583	0.70	0.0246	0.5188	0.7000	0.5188	0.7000	0.5188	0.7000	0.7000	0.7000
14	11.4700	0.50	0.7244	0.5188	0.5000	0.5188	0.5000	0.5188	0.5000	0.5000	0.5000
Weighted sum of squares of residuals (1):				0.1085	0.1059	0.0422	0.0395	0.0026	1.03E–06	1.03E–06	0.0000

```
 ----------------------------------------
 RETURN FROM L2WPMA WITH MODE =     1
 NUMBER OF DATA (F)           =    14
 NUMBER OF MONOTONIC SECTIONS =     4
 NUMBER OF LOCAL MINIMA IN F  =     5
 NUMBER OF LOCAL MAXIMA IN F  =     4
 ----------------------------------------

        INDICES OF EXTREMA AT OPTIMUM
        Increment      Data index
        (J)            (ITHETA)
            0               1
            1               4
            2               6
            3               7
            4              14

        INDICES OF ACTIVE CONSTRAINTS AT OPTIMUM
        Increment      Active constraint      Lagrange mult
        (I)            (IACT)                 (PAR(IACT))
            1               3                   0.02
            2              10                  -2.20
            3              11                  -4.41
            4              12                  -3.25
            5              13                  -1.99
            6              14                  -0.80

        INDICES OF KNOTS AT OPTIMUM
        Increment      Knot index            Spline coeff
        (I)            (IAKN)                (Y(IAKN))
            1               1                 -0.1000
            2               2                  0.7000
            3               4                  0.8700
            4               5                 -1.0000
            5               6                 -1.1100
            6               7                  1.0000
            7               8                  1.0000
            8               9                  0.1017
            9              14                  0.1017

    Data points  Measurements  Data weights  Best appxmtn  Lagrange mult
    (X)          (F)           (WF)          (Y)           (PAR)

  1    1.0000      -0.1000       1.0000       -0.1000
  2    2.0000       0.7100       1.0000        0.7000        0.00
  3    3.0000       0.6900       1.0000        0.7000        0.02
  4    4.0000       0.8700       1.0000        0.8700        0.00
  5    5.0000      -1.0000       1.0000       -1.0000        0.00
  6    6.0000      -1.1100       1.0000       -1.1100        0.00
  7    7.0000       1.0000       1.0000        1.0000        0.00
  8    8.0000       1.0000       1.0000        1.0000        0.00
  9    9.0000      -1.0000       1.0000        0.1017        0.00
 10   10.0000      -1.0000       1.0000        0.1017       -2.20
 11   11.0000       0.6800       1.0000        0.1017       -4.41
 12   12.0000       0.7300       1.0000        0.1017       -3.25
 13   13.0000       0.7000       1.0000        0.1017       -1.99
 14   14.0000       0.5000       1.0000        0.1017       -0.80

Value of objective function at the optimum=    0.3673483610E+01
```

Fig. 1 Output of software package `L2WPMA` due to a driver program presenting the termination status, the number of data, the number of monotonic sections, the number of data extrema counting possibly the end point indices I1 and N (as follows from the definition of *L* and *U*), the optimal values of the integer variables (namely the positions of the turning points of the fit), the indices of active constraints and the corresponding Lagrange multipliers, the knot indices and the coefficients of a spline representation of the best fit and, Y, the best fit to the data, together with X, F, WF and PAR.

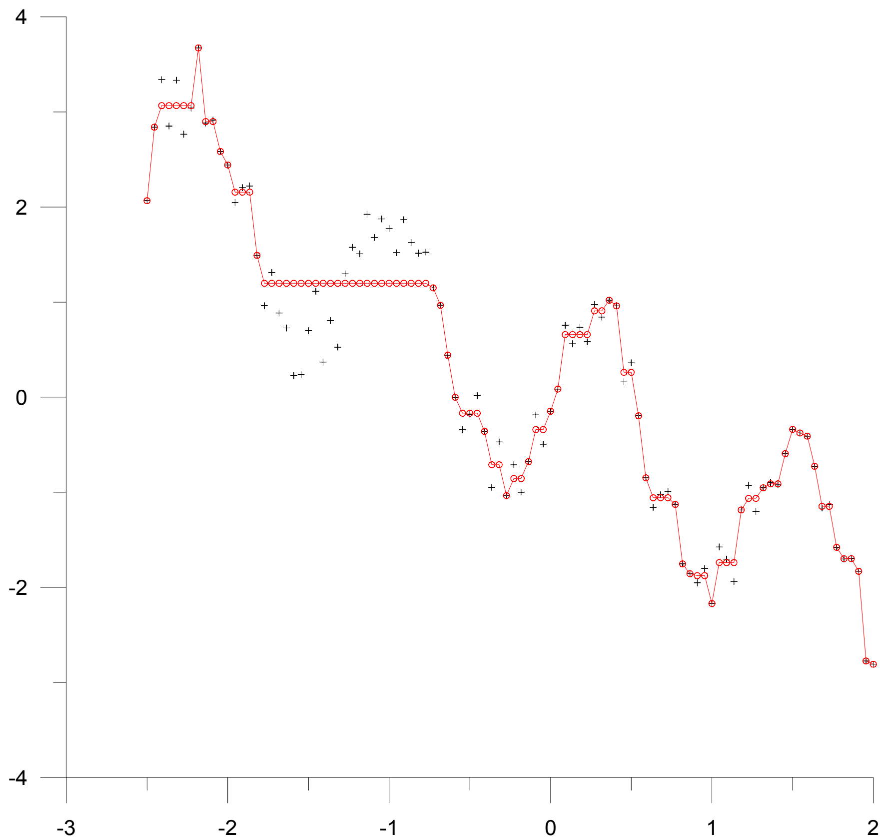

Fig. 2 Best least squares approximation with k=6 or k=7 monotonic sections to 100 data points generated by adding uniformly distributed random numbers from the interval (–0.5,0.5) to the measurements of $f(x)=\sin(5x)-x$ at equally spaced abscissae. The data are denoted by (+), the best approximation by (o) and the piecewise linear interpolant to the smoothed values illustrates the fit. The first decreasing section suggests that a better approximation is possible by increasing k.

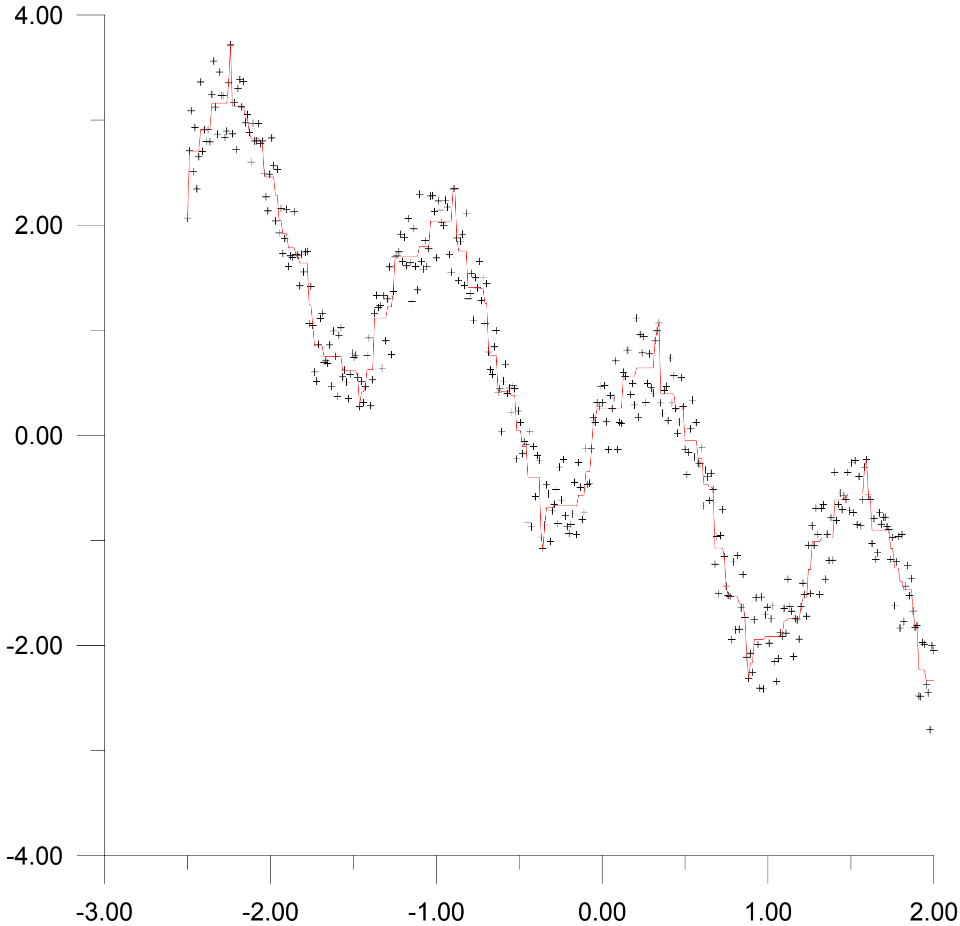

Fig. 3 Best least squares approximation with 8 monotonic sections to 200 data points generated as in Fig. 2. The data are denoted by (+) and the piecewise linear interpolant to the smoothed values illustrates the fit.