CR-133984

[redacted]

(NASA-CR-133984) DATA SMOOTHING AND ERROR DETECTION BASED ON LINEAR INTERPOLATION (Rice Univ.) 25 p HC $3.25 CSCL 05B

N73-31131

Unclas
G3/08 13561

ICSA

INSTITUTE FOR COMPUTER SERVICES AND APPLICATIONS

RICE UNIVERSITY

ICSA-275-025-006

DATA SMOOTHING AND ERROR DETECTION BASED ON LINEAR INTERPOLATION

by

V. M. Guerra and R. A. Tapia

Dept. of Mathematical Sciences
Rice University

ABSTRACT

In this paper we present a method, based on linear interpolation, for detecting and correcting bad data points in a set of data without contaminating the good data points. We are not concerned with the small random errors usually attributed to a noisy system and assume that the data points which are in error are relatively isolated from each other and that the number of such points is small compared to the total number of data points

Institute for Computer Services & Applications

Rice University

Houston, Texas 77001

September 1972

Research supported under NASA Contract #NAS9-12776

Data Smoothing and Error Detection Based on Linear Interpolation

V. M. Guerra* and R. A. Tapia*

1. Introduction. In the handling of large sets of data it is not uncommon to inadvertently introduce errors into the data. Typical causes for the introduction of error might be:

(a) Reading error;

(b) Keypunch error;

(c) Machine malfunction.

In this paper we consider the problem of detecting and removing these errors without contaminating the good data. We are not concerned with the small random errors usually attributed to a noisy system. Therefore it seems reasonable to expect that the data points which are in error are relatively isolated from each other and that the number of such points is small compared to the total number of data points; however the errors themselves will probably be quite large. This latter consideration alone forces us to reject the well-known averaging techniques for data smoothing [3]; since the bad data would significantly effect the good data.

---

* Department of Mathematical Sciences, Rice University, Houston, Texas 77001. This work was sponsored by NASA-MSC under contract NAS 9-12776.

If we consider removing the errors by smoothing the data using splines and least squares (see [4], [6], [7] and [8]), then it is well-known that the L_2 norm (least squares) is sensitive to outliers (hence, again our bad points would influence our good points). This observation immediately suggests the use of splines and the L_1 norm via linear programming with differential inequality constraints (see [2] and [5]). Our main reason for rejecting both L_1 and L_2 (as well as L_∞) approaches is both obvious and extremely realistic. Namely, for large data sets, such as the remote sensing data presently being analyzed at NASA Manned Spacecraft Center, the use of the L_1 or L_2 approach would require a prohibitive amount of computer time and computer storage and would undoubtedly lead to extreme numerical instabilities. The amount of work required to implement these two approaches is of the order of n^3 where n is the number of data points. The approach we are about to describe is of order n (i.e. the work increases linearly with the data). Moreover, while we acknowledge the fact that both the L_1 and L_2 approaches would probably give satisfactory results for small data sets we feel our approach will do as well.

In this paper we consider only the one-dimensional problem. In subsequent papers we will extend our approach to higher dimensions and also consider using methods of interpolation more sophisticated than linear interpolation.

2. The Linear Smoothing Algorithm. Consider a set of points in the plane with equally spaced abscissas, say

$$A = \{(x_i, y_i): i=1,\ldots,m\}.$$

Definition 1. By an anchor point of the set A we mean a point which is assumed to be correct and is not to be smoothed.

Remark. We shall assume that (x_1, y_1) and (x_m, y_m) are always anchor points of the set A.

Definition 2. By the point energy of the non-anchor point $(x_i, y_i) \in A$ we mean the (ordinate) distance from the line passing through the points (x_{i-1}, y_{i-1}) and (x_{i+1}, y_{i+1}) to the point (x_i, y_i). If (x_i, y_i) is an anchor point, then its point energy is zero.

Definition 3. By the total energy of the set A we mean the sum of the point energies of all points in A(i.e., the L_1-norm of the point energies).

Definition 4. By the smoothness of the set A we mean the largest point energy (i.e., the L_∞-norm of the point energies). Moreover, we say that A is ϵ-smooth if the smoothness of A is less than or equal to ϵ.

Proposition 1. The following are equivalent:

(a) The set A is 0-smooth;

(b) The set A has zero total energy;

(c) The set A lies on the piecewise-linear function which interpolates the anchor points of A.

Proof. The proof is straightforward.

Definition 4. By the normalized second difference at the point $(x_i,y_i) \in A$ we mean

$$r_i = \tfrac{1}{2}y_{i+1} - y_i + \tfrac{1}{2}y_{i-1}, \quad i=2,\ldots,m-1.$$

Proposition 2. If α_i denotes the point energy of the non-anchor point $(x_i,y_i) \in A$, then

$$\alpha_i = |r_i|.$$

Proof. The proof is not difficult.

Definition 5. By the linear smoothing approach we mean the transformation of the set A into an ϵ-smooth set by successive changes of the values of the ordinates of the points with the largest point energies. Specifically, if $1 \le k \le n$ is such that $\alpha_k = \max\{\alpha_i : 1 \le i \le n\}$ (if more than one such k exists then we choose the one of smallest index), then we change the point (x_k,y_k) to the point $(x_k,y_k+\theta_k r_k)$ for some $\frac{1}{2} \le \theta_k \le 1$ and repeat the procedure until (hopefully) the transformed set is ϵ-smooth (for some given $\epsilon \ge 0$).

Remark. If $\theta_k=0$, then the data is not modified. If $\theta_k=1$, then we are moving the point (x_k,y_k) onto the line interpolating its two neighbors; hence by requiring $\frac{1}{2} \le \theta_k \le 1$ we have guaranteed that the point energy at the k-th point

will decrease at least by a factor of $\frac{1}{2}$.

Remark. For simplicity we may choose θ_k always equal to a constant, e.g., $\frac{1}{2}$, $\frac{3}{4}$ or 1.

3. Convergence of the Linear Smoothing Algorithm. To distinguish between the values of the point energies and other quantities at different iterations a subscript, or a second subscript (whatever the case may be) will be added whenever necessary. For example A_n will denote the set A at the n-th iteration of the linear smoothing process. We also let A_0 denote A.

Proposition 3. If E_n denotes the total energy of A_n, then $\{E_n\}$ is a monotone nonincreasing sequence. Moreover $E_{n+1} \leq E_n - \frac{1}{2}\theta_k \alpha_{k,n}$ (k denotes the index of point in A_n which is to be modified) if either the (k-1)-th or (k+1)-th point is an anchor point. Finally we have $E_{n+1} = E_n$ if and only if $r_{k-1,n}$, $r_{k,n}$ and $r_{k+1,n}$ are of the same sign and the (k-1)-th and (k+1)-th points are not anchor points.

Proof. All the point energies except possibly α_{k-1}, α_k and α_{k+1} are the same at the n-th and (n+1)-th iteration. Moreover

$$y_{k,n+1} = y_{k,n} + \theta_k r_{k,n};$$

hence

$$r_{k+1,n+1} = r_{k+1,n} + \tfrac{1}{2}\theta_k r_{k,n}$$

$$r_{k-1,n+1} = r_{k-1,n} + \tfrac{1}{2}\theta_k r_{k,n} \tag{1}$$

$$r_{k,n+1} = (1-\theta_k) r_{k,n}.$$

Now since $0 < \theta_k \leq 1$ we have

$$|r_{k,n+1}| = (1-\theta_k)|r_{k,n}|.$$

Therefore taking absolute values, using the triangle inequality and adding in (1) we have that $E_{n+1} \leq E_n$. Clearly if the (k-1)-th or the (k+1)-th point is an anchor point we must have a decrease in the total energy of at least $\frac{1}{2}\theta_k |r_{k,n}|$. Again from (1) we will have a decrease if either $r_{k-1,n}$ or $r_{k+1,n}$ has a different sign than $r_{k,n}$. This proves the proposition.

Remark. Although the energy of A_{n+1} may be equal to the energy of A_n (i.e., no decrease) it may happen that A_{n+1} is significantly smoother than A_n. However a simple example can be constructed to show that the smoothness (in contrast to the energy) is not monotone nonincreasing; hence for certain purposes the natural criterion (norm) to use is the energy.

Proposition 4. If the total energy of the set A_n is not zero, then the maximum number of iterations that can occur without decreasing this energy is bounded above by

$$B = 2^m$$

(where m is the number of data points).

Proof. We will first show that if the energy is not decreased, then we can only modify a particular point twice before moving on to another point. Suppose we operate twice on the point

(x_k, y_k). The result of the first iteration is given by (1) and the result of the second is easily seen to be

$$r_{k+1,n+2} = r_{k+1,n} + (\tfrac{1}{2}\theta_k + \tfrac{1}{2}\theta_k(1-\theta_k)^2) r_{k,n}$$

$$r_{k-1,n+2} = r_{k-1,n} + (\tfrac{1}{2}\theta_k + \tfrac{1}{2}\theta_k(1-\theta_k)^2) r_{k,n}$$

$$r_{k,n+2} = (1-\theta_k)^2 r_{k,n}.$$

Now since the energy did not decrease we must have by Proposition 3, that $r_{k-1,n}$, $r_{k,n}$ and $r_{k+1,n}$ are all of the same sign. Also, since $\tfrac{1}{2} \leq \theta_k \leq 1$ we have

$$\tfrac{1}{2}\theta_k \geq (1-\theta_k)^2;$$

this shows that $|r_{k+1,n+2}| > |r_{k,n+2}|$. It follows that $(x_k, y_{k,n+2})$ will not be modified on the subsequent iteration. It is not difficult to show that we will move one point in at most 2 iterations, 2 points in at most $2+2^2$ iterations and in general K points in at most $\sum_{i=1}^{K} 2^i$ iterations. This proves the proposition.

Remark. The bound given in the previous proposition is far from being sharp. It merely demonstrates an important fact which will allow us to prove convergence.

Proposition 5. The sequence E_n giving the total energy at each iteration of the linear smoothing algorithm converges

to zero.

Proof. From Proposition 3 $\{E_n\}$ is a monotone nonincreasing sequence which is bounded below by zero; therefore it converges.

Suppose $E_n \to E \geq 0$. First note that for each $n=1,2,\ldots$ there exists an integer $1 \leq j(n) \leq m$ such that $\alpha_{j(n),n} \geq \frac{E}{m}$.

To see this suppose $\alpha_{i,n} < \frac{E}{m}$ for $1 \leq i \leq m$.

Then $E_n = \sum_{i=1}^{m} \alpha_{i,n} < m \frac{E}{m} = E$, which contradicts Proposition 3.

By Proposition 3 and 4 for some integer $n \leq J(n) \leq n+2^m$ we have that

$$E_{J(n)} \leq E_i - \tfrac{1}{2}\theta_k \alpha_{k,i}$$

$$\leq E_n - \tfrac{1}{2}\theta_k \alpha_{j(i),i}$$

$$\leq E_n - E/(4m). \qquad (i=J(n)-1)$$

Now, since $E_n \geq E$ we have $E_n - E = |E_n - E|$; therefore given $\epsilon > 0$ there exists $N > 0$ such that $E_n - E < \epsilon$ whenever $n > N$. We have

$$E_{J(n)} - E \leq E_n - E/(4m) - E$$

$$< \epsilon - E/(4m).$$

Now choosing $\epsilon < E/(4m)$ gives $E_{J(n)} < E$; which again contradicts Proposition 3. This proves the proposition.

Definition 6. Let $A_n = \{(x_i, y_i^n) : i=1,\ldots,m\}$ for $n=0,1,2,\ldots$.

We say that the sequence of sets $\{A_n\}$ converges to the set

$A^* = \{(x_i, y_i^*)\colon i=1,\ldots,m\}$ if $y_i^n \to y_i^*$ for $i=1,\ldots,m$.

Proposition 6. The total energy is a continuous functional, i.e. if $A_n \to A^*$, then $E(A_n) \to E(A^*)$.

Proof. If $A_n \to A^*$, then the sequence of vectors $y_n=(y_1^n,\ldots y_m^n)$ converges to the vector $y^*=(y_1^*,\ldots,y_m^*)$ pointwise; hence in any norm. Let $\alpha_{j,n}$ denote the point energy at the j-th point of A_n, with a similar definition for α_j^*. A simple construction should convince the reader that

$$|\alpha_{j,n}-\alpha_j^*| \leq 2\|y_n-y^*\|_\infty.$$

It follows that $\alpha_{j,n} \to \alpha_j^*$ and therefore $E(A_n) \to E(A^*)$. This proves the proposition.

Proposition 7. The linear smoothing algorithm converges, i.e., the sequence of sets $\{A_n\}$ converges to a set A^* with total energy zero.

Proof. We use the same notation as in the proof of Proposition 6. Clearly $\|y_n\|_\infty \leq \|y_0\|_\infty$, for $n=1,2,3,\ldots$. Hence $\{y_k\}$ must have a subsequence which is convergent, say to y^*. If A^* is the set corresponding to y^*, then by Proposition 5 and Proposition 6 $E(A^*) = 0$. If the entire sequence does not converge to y^*, then each neighborhood of y^* excludes infinitely many members of $\{y_n\}$. These excluded members must have a convergent subsequence. If y^{**} denotes this limit, then

$E(y^*) = E(y^{**}) = 0$; hence $y^* = y^{**}$; but this is a contradiction. This proves the proposition.

Remark. We have spent considerable time and effort proving that the linear smoothing algorithm converges to a solution which could have been immediately written down. Of course the complete philosophy of this approach is that we only allow a few iterations. Indeed, as our examples will show, this philosophy is quite natural and analogous to what would be done by an artist or a loftsman by hand. Namely, the algorithm converges very quickly to an acceptable solution and from then on the convergence is extremely slow. Our main reason for proving convergence was to demonstrate that the algorithm will not oscillate.

4. Examples. Consider the twenty points $A = \{(1,y(1)),\ldots, (20,y(20))\}$ taken from the graph of the cubic

$$y(x) = \frac{(x-7)(x-10)(x-13)}{910} .$$

As it stands the set A is .0297-smooth. In our first example we will introduce an error of .5 in the 11-th data point. We introduce errors of -.5 in the 9-th point and .5 in the 11-th point for our second example. Finally, the third example will consist of the points of A with an error of -.5 in the 10-th point and .5 in the 11-th point. Based on our theory we should expect the first two examples to behave better than the third. Indeed, we obtain a curve as smooth as the original curve in just one iteration for the first example and in just two iterations for the second example; however the third example requires five iterations. All the following calculations were performed using a value of one for θ_k.

Observe that all three examples show that the energy is monotone decreasing and that the smoothness is not monotone decreasing. However these examples imply that a reasonable stopping criterion (since we do not want to end up with a straight line) is to stop at the first iteration where the smoothness increases. In our examples this is the iteration at which the original smoothness is restored.

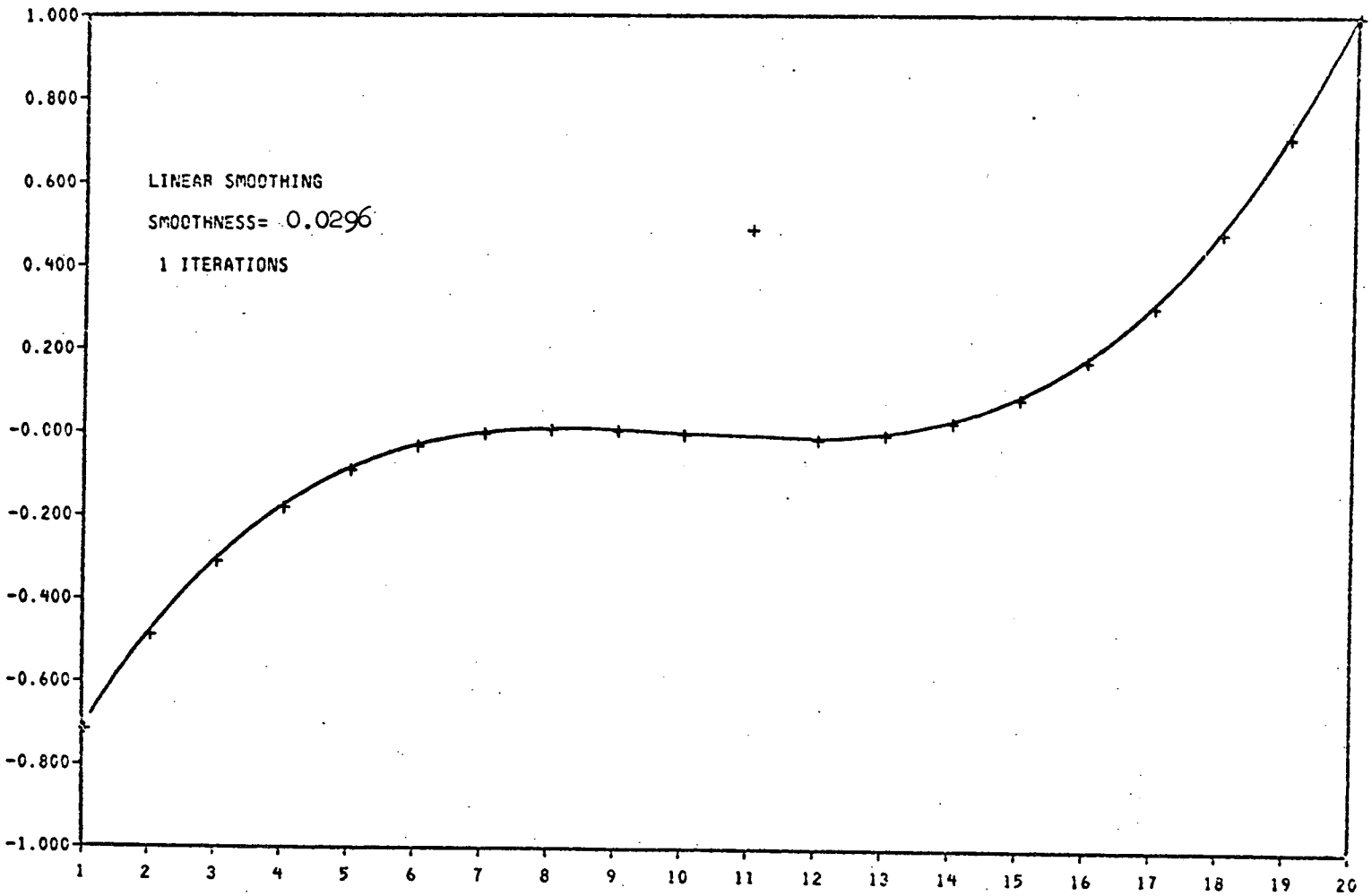
The following tables and graphs are reasonably self-explanatory; however we point out that the values for the energy and smoothness were calculated at the beginning of each iteration and not at the end.

EXAMPLE 1

	x	y	y	
1	-0.00000	-0.71209	-0.71209	
2	0.05000	-0.48352	-0.48352	
3	0.10000	-0.30769	-0.30769	
4	0.15000	-0.17802	-0.17802	
5	0.20000	-0.08791	-0.08791	
6	0.25000	-0.03077	-0.03077	
7	0.30000	0.0	0.0	
8	0.35000	0.01099	0.01099	
9	0.40000	0.00879	0.00879	
10	0.45000	0.0	0.0	
11	0.50000	-0.00879	0.49121	error
12	0.55000	-0.01099	-0.01099	
13	0.60000	0.0	0.0	
14	0.65000	0.03077	0.03077	
15	0.70000	0.08791	0.08791	
16	0.75000	0.17802	0.17802	
17	0.80000	0.30769	0.30769	
18	0.85000	0.48352	0.48352	
19	0.90000	0.71209	0.71209	
20	0.95000	1.00000	1.00000	

ITER	ENERGY	SMOOTHNESS	POINT MOVED
1	1.26044	0.49670	11
2	0.26703	0.02967	19
3	0.25220	0.04121	18
4	0.25220	0.04368	17
5	0.25220	0.04162	16
6	0.25220	0.03729	15
7	0.25220	0.03183	14
8	0.25220	0.02637	2
9	0.23901	0.03626	3
10	0.23901	0.03791	4

EXAMPLE 1

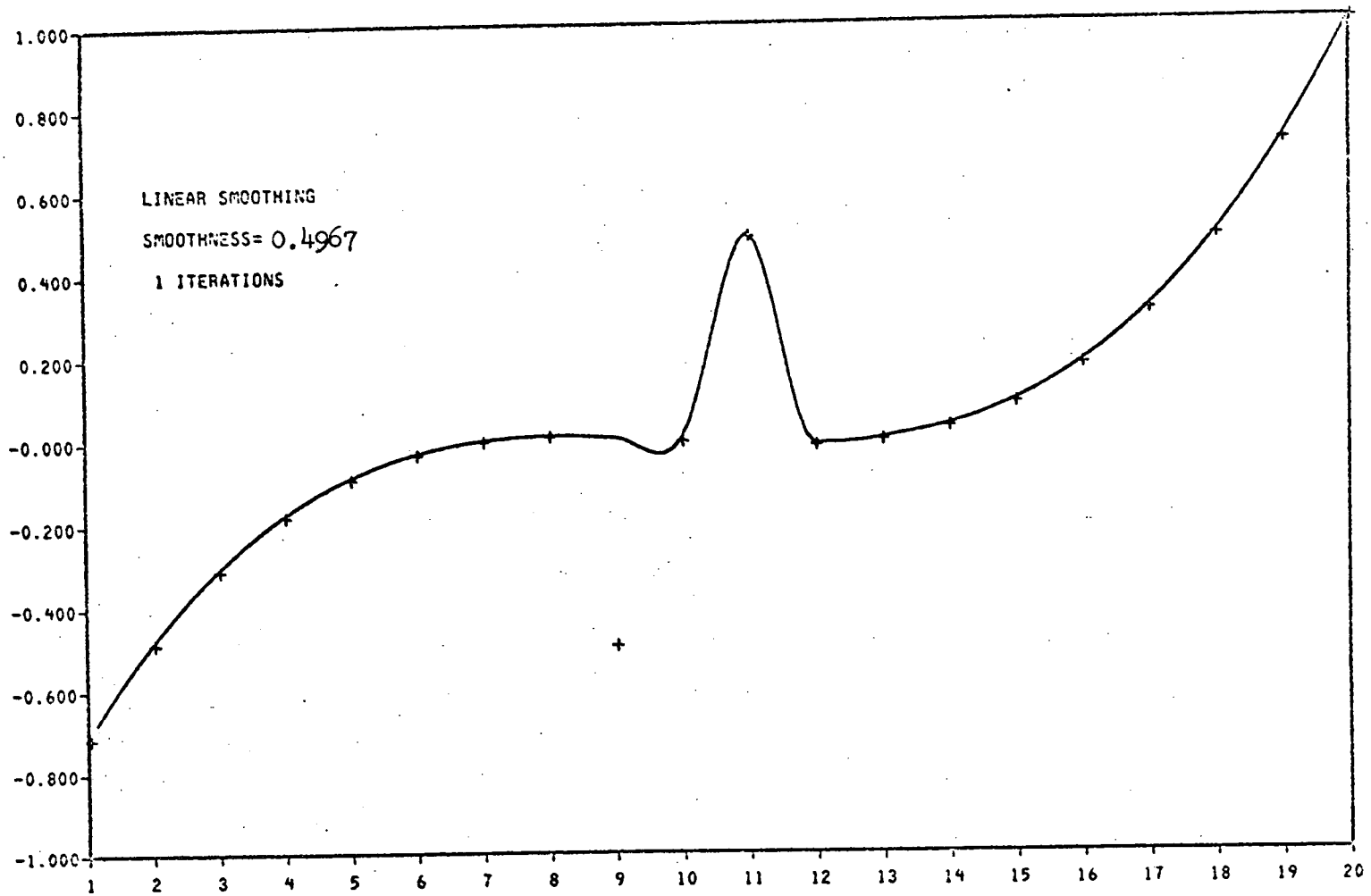

LINEAR SMOOTHING
SMOOTHNESS= 0.0296
1 ITERATIONS
1.000
0.800
0.600
0.400
0.200
-0.000
-0.200
-0.400
-0.600
-0.800
-1.000
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

EXAMPLE 2

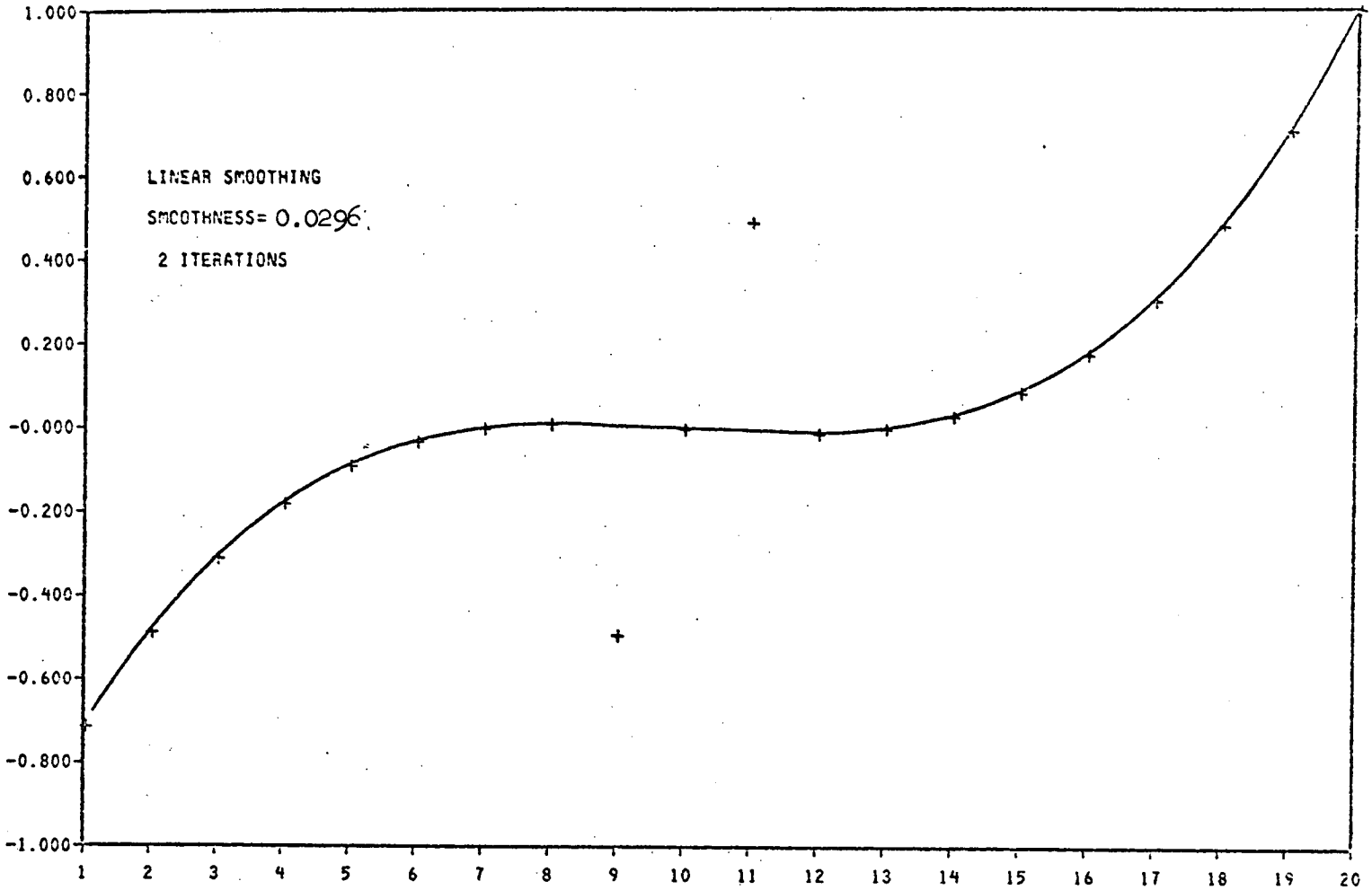
	x	y	y	
1	-0.00000	-0.71209	-0.71209	
2	0.05000	-0.48352	-0.48352	
3	0.10000	-0.30769	-0.30769	
4	0.15000	-0.17802	-0.17802	
5	0.20000	-0.08791	-0.08791	
6	0.25000	-0.03077	-0.03077	
7	0.30000	0.0	0.0	
8	0.35000	0.01099	0.01099	
9	0.40000	0.00879	-0.49121	error
10	0.45000	0.0	0.0	
11	0.50000	-0.00879	0.49121	error
12	0.55000	-0.01099	-0.01099	
13	0.60000	0.0	0.0	
14	0.65000	0.03077	0.03077	
15	0.70000	0.08791	0.08791	
16	0.75000	0.17802	0.17802	
17	0.80000	0.30769	0.30769	
18	0.85000	0.48352	0.48352	
19	0.90000	0.71209	0.71209	
20	0.95000	1.00000	1.00000	

ITER	ENERGY	SMOOTHNESS	POINT MOVED
1	1.75384	0.49670	9
2	1.25714	0.49670	11
3	0.26374	0.02967	19
4	0.24890	0.04121	18
5	0.24890	0.04368	17
6	0.24890	0.04162	16
7	0.24890	0.03729	15
8	0.24890	0.03183	14
9	0.24890	0.02637	2
10	0.23571	0.03626	3

EXAMPLE 2


LINEAR SMOOTHING
SMOOTHNESS= 0.4967
1 ITERATIONS
1.000
0.800
0.600
0.400
0.200
-0.000
-0.200
-0.400
-0.600
-0.800
-1.000
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

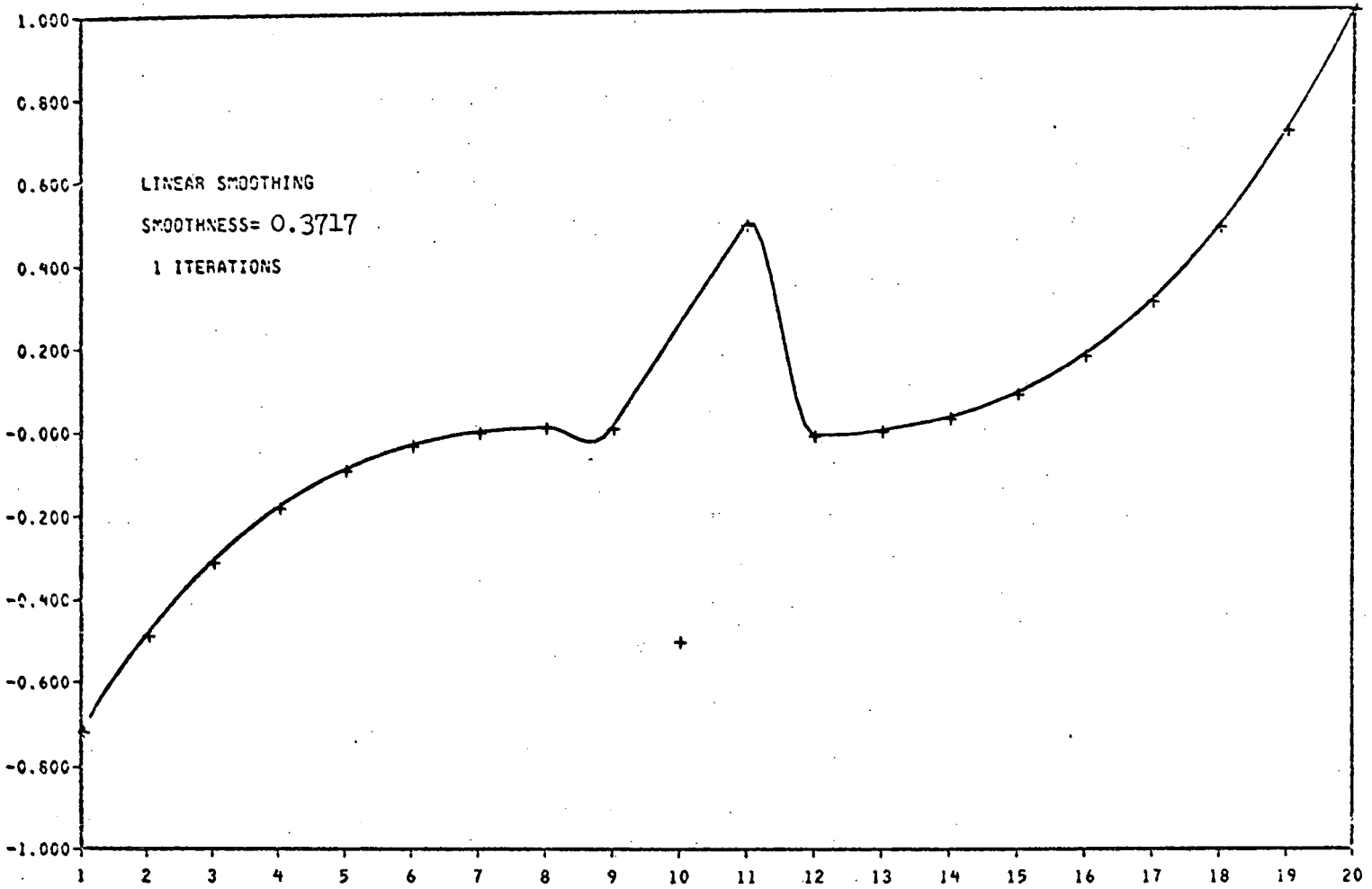
EXAMPLE 2


LINEAR SMOOTHING
SMOOTHNESS= 0.0296
2 ITERATIONS
1.000
0.800
0.600
0.400
0.200
-0.000
-0.200
-0.400
-0.600
-0.800
-1.000
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

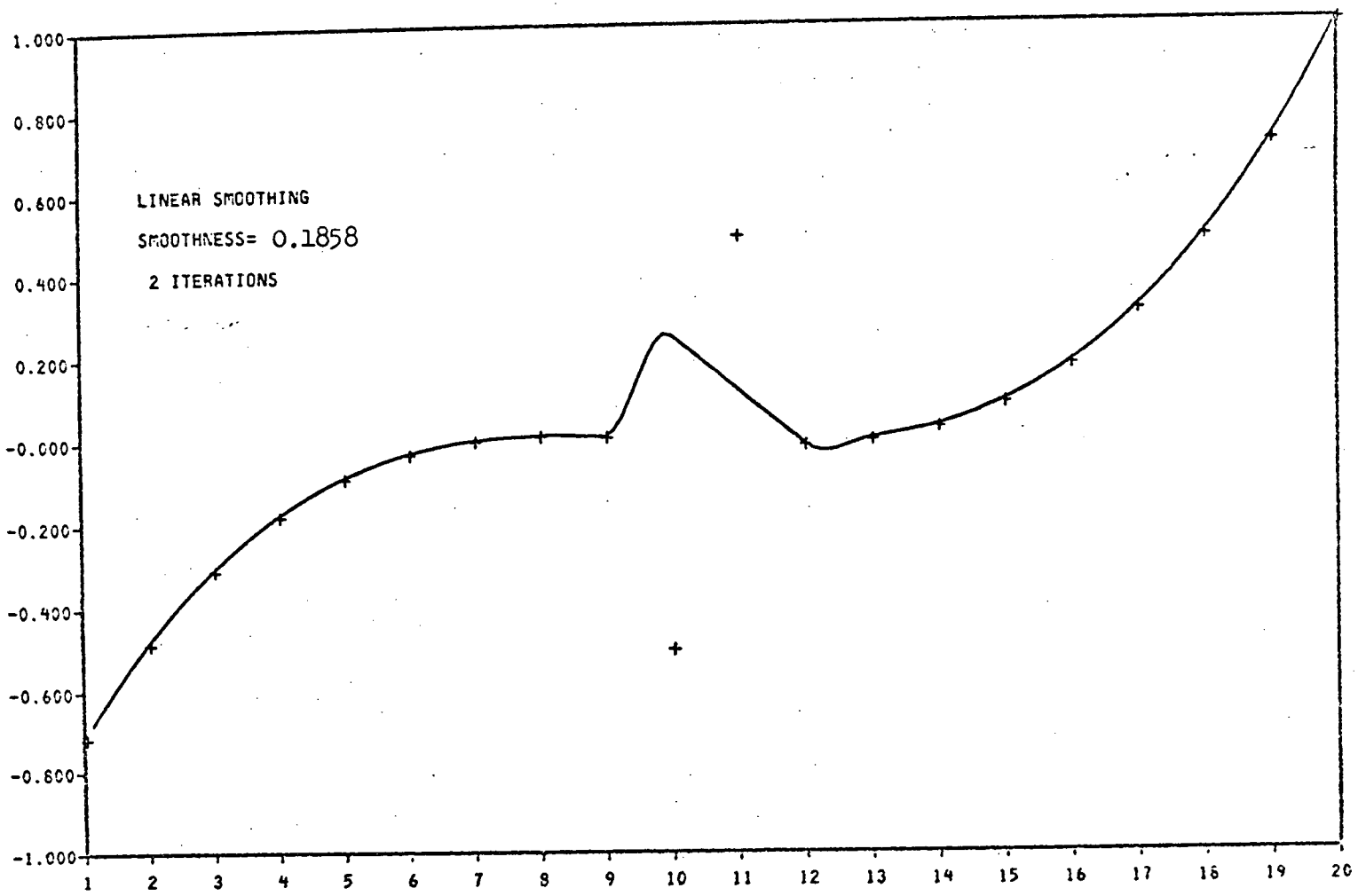
	x	y	y	
1	-0.00000	-0.71209	-0.71209	
2	0.05000	-0.48352	-0.48352	
3	0.10000	-0.30769	-0.30769	
4	0.15000	-0.17802	-0.17802	
5	0.20000	-0.08791	-0.08791	
6	0.25000	-0.03077	-0.03077	
7	0.30000	0.0	0.0	
8	0.35000	0.01099	0.01099	
9	0.40000	0.00879	0.00879	
10	0.45000	0.0	-0.50000	error
11	0.50000	-0.00879	0.49121	error
12	0.55000	-0.01099	-0.01099	
13	0.60000	0.0	0.0	
14	0.65000	0.03077	0.03077	
15	0.70000	0.08791	0.08791	
16	0.75000	0.17802	0.17802	
17	0.80000	0.30769	0.30769	
18	0.85000	0.48352	0.48352	
19	0.90000	0.71209	0.71209	
20	0.95000	1.00000	1.00000	

ITER	ENERGY	SMOOTHNESS	POINT MOVED
1	2.26044	0.75000	10
2	1.00385	0.37170	11
3	0.63214	0.18585	10
4	0.44629	0.09293	11
5	0.35337	0.04646	10
6	0.30690	0.02967	19
7	0.29207	0.04121	18
8	0.29207	0.04368	17
9	0.29207	0.04162	16
10	0.29207	0.03729	15

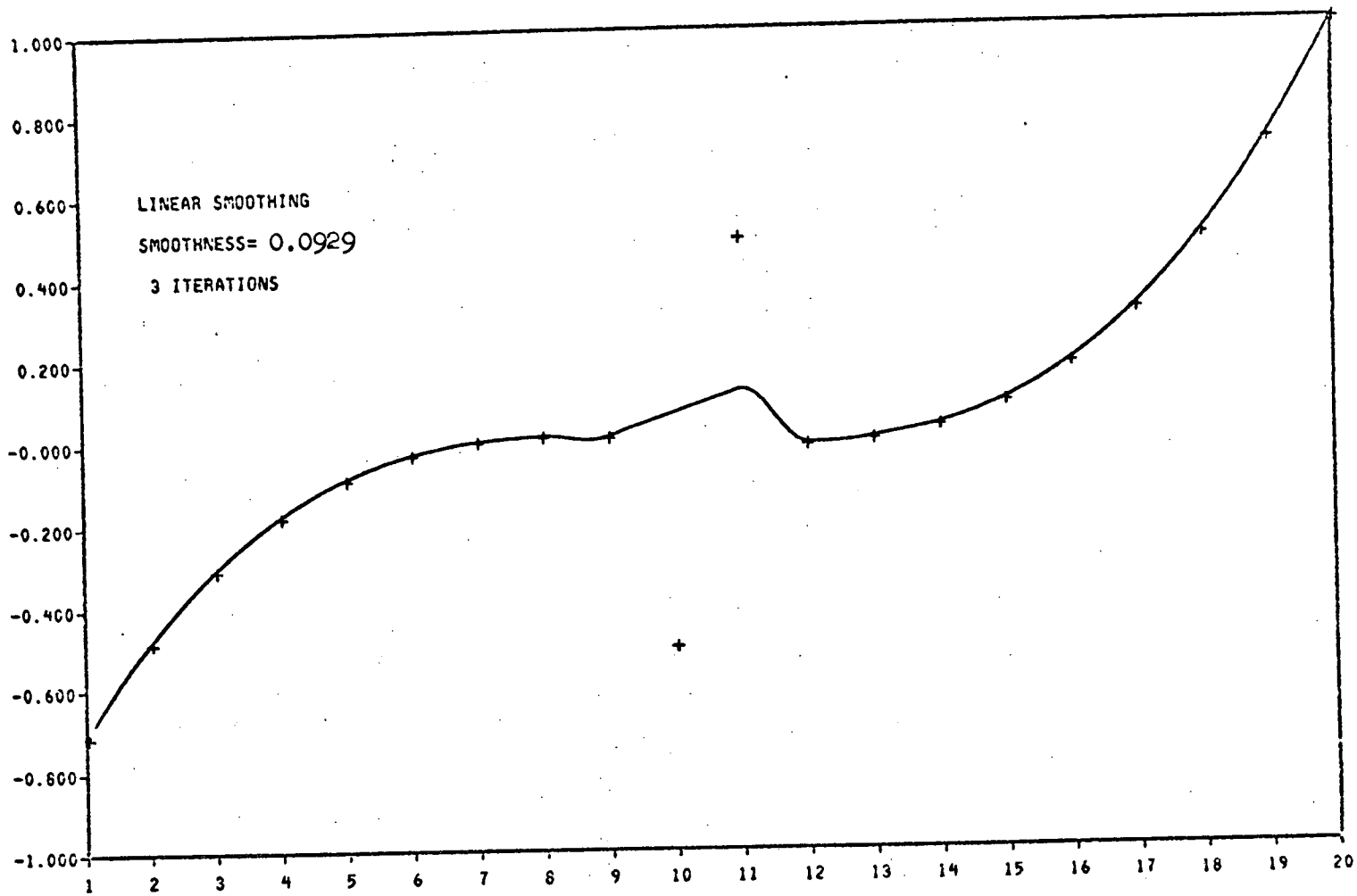
EXAMPLE 3


LINEAR SMOOTHING
SMOOTHNESS= 0.3717
1 ITERATIONS
1.000
0.800
0.600
0.400
0.200
-0.000
-0.200
-0.400
-0.600
-0.800
-1.000
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

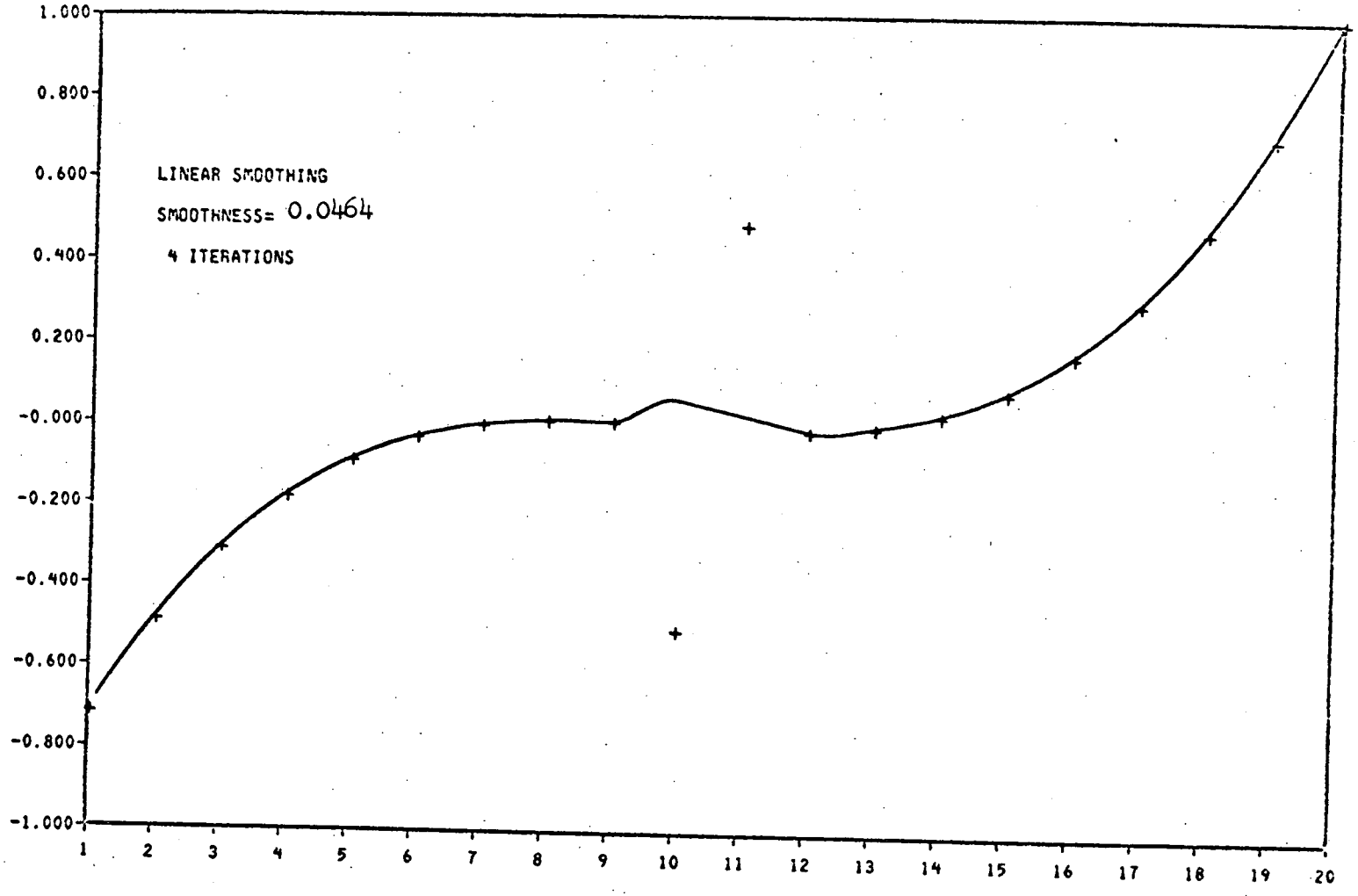
EXAMPLE 3


LINEAR SMOOTHING
SMOOTHNESS= 0.1858
2 ITERATIONS
1.000
0.800
0.600
0.400
0.200
-0.000
-0.200
-0.400
-0.600
-0.800
-1.000
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

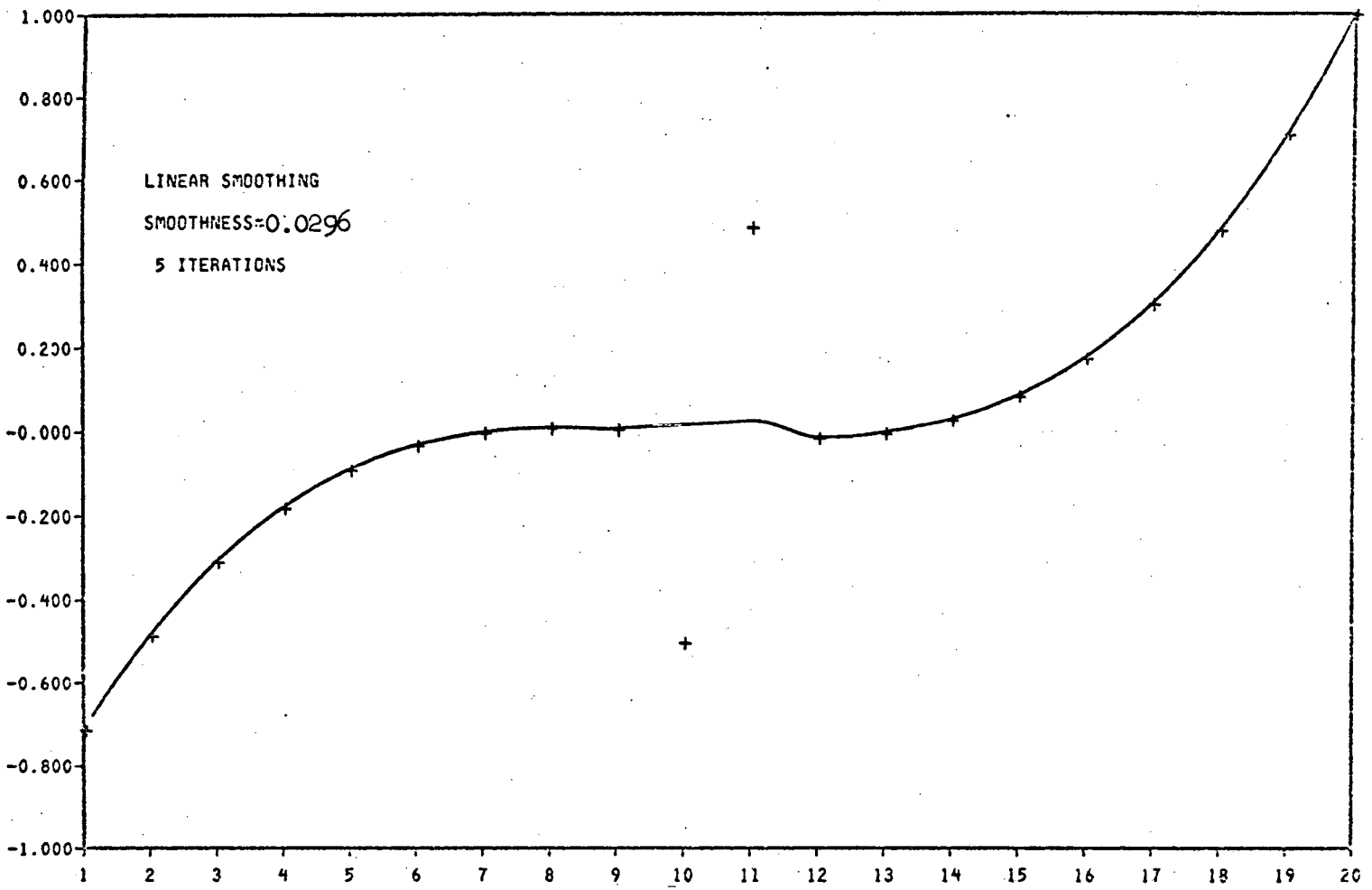
EXAMPLE 3


LINEAR SMOOTHING
SMOOTHNESS= 0.0929
3 ITERATIONS
1.000
0.800
0.600
0.400
0.200
-0.000
-0.200
-0.400
-0.600
-0.800
-1.000
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

EXAMPLE 3


LINEAR SMOOTHING
SMOOTHNESS= 0.0464
4 ITERATIONS
1.000
0.800
0.600
0.400
0.200
-0.000
-0.200
-0.400
-0.600
-0.800
-1.000
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

EXAMPLE 3


LINEAR SMOOTHING
SMOOTHNESS=0.0296
5 ITERATIONS
1.000
0.800
0.600
0.400
0.200
-0.000
-0.200
-0.400
-0.600
-0.800
-1.000
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

References

1. H. Akima, A new method of interpolation and smooth curve fitting based on local procedures, Jour. of A.C.M., 17 (1970), 589-602.

2. S.A. Berger, W.C. Webster, R.A. Tapia and D.A. Atkins, Mathematical ship lofting, Jour. of Ship Research, 10 (1966), 203-214.

3. G.E.P. Box and G.M. Jenkins, Time Series Analysis Forecasting and Control, Holden-Day, San Francisco, 1970.

4. T.N.E. Greville, Introduction to spline functions, In Theory and Applications of Spline Functions, T.N.E. Greville Editor, Academic Press, New York, 1969 pp. 1-35.

5. P. La Fata and J.B. Rosen, An interactive display for approximation by linear programming, Comm. of A.C.M., 13 (1970), 651-659.

6. C. Reinsch, Smoothing by spline functions, Numer. Math., 10 (1967), 177-183.

7. I.J. Schoenberg, Spline functions and the problem of graduation, Proc. Nat. Acad. Sci., U.S.A. 52 (1964), 947-950.

8. F. Theilheimer and W. Starkweather, The fairing of ship lines on a high-speed computer, David Taylor Model Basin, Applied Mathematics Laboratory Report 1474, January 1961.