29

AN INTERACTIVE DISPLAY FOR APPROXIMATION BY LINEAR PROGRAMMING

by

P. LaFata and J. B. Rosen

Technical Report #35

March 1969

Computer Sciences Department
1210 West Dayton Street
Madison, Wisconsin 53706

AN INTERACTIVE DISPLAY FOR APPROXIMATION BY LINEAR PROGRAMMING*

by

P. LaFata and J. B. Rosen

ABSTRACT

An interactive program with a graphical display has been developed for the approximation of data by means of a linear combination of functions selected by the user. The coefficients of the approximation are determined by linear programming so as to minimize the error in either the L_1 or L_∞ norm. Auxiliary conditions such as monotonicity or convexity of the approximation can also be imposed. This interactive system is described (including user instructions) and several examples of its use are given.

*This research was supported in part by the National Science Foundation under Grant NSF GP-6070, and in part by the University of Wisconsin Computing Center.

1. INTRODUCTION

The approximation of functions and the approximate solution of ordinary differential equations using linear programming (LP) has been discussed by a number of authors recently [10]. LP is also the computational technique used for the approximate solution of certain types of partial differential equations [12, 13, 3]. In all of these cases the approximation is obtained as a linear combination of selected functions. The efficiency and accuracy of these methods therefore depend, in many cases, on a suitable choice of approximating functions. Furthermore, error bounds are generally available only after the approximation has been computed. Once the choice of functions (and possibly other auxiliary conditions) has been made, the "best" coefficients are determined by LP. Information about the approximation thus obtained (including an error bound) will often enable the user to decide whether to add more functions, replace some functions, or make other changes so as to improve the approximation. Some of this desired information about an approximation is most conveniently presented by means of a graphical display. Thus an interactive graphical display coupled with an appropriate LP capability should be of great value in the rapid solution of this kind of generalized approximation problem.

In order to test the feasibility and gain experience in this area, an interactive program using a graphical display and LP has been developed for the approximation of data in the L_∞ and L_1 norms. This program is used with the Burroughs 5500 and its associated graphical display unit. By means of a keyboard input, the user specifies the data set (t_j, y_j), $j = 1, 2, \ldots, n$, with $t_j \in [t_1, t_n]$ and a set of selected functions $\varphi_i(t)$, $i = 1, \ldots, m$, with $m \leq n$. The program will then determine the m coefficients α_i, so as to minimize either $\max_j |v(\alpha, t_j) - y_j|$ (L_∞ norm), or $\sum_{j=1}^{n} |v(\alpha, t_j) - y_j|$ (L_1 norm), where $v(\alpha, t) = \sum_{i=1}^{m} \alpha_i \varphi_i(t)$. For $m = n$, and provided the matrix F defined by (2.5) is nonsingular, the approximation $v(\alpha, t)$ will interpolate the specified data.

At the user's option the minimization may be carried out subject to additional conditions on the approximation $v(\alpha, t)$. Any or all of the following conditions may be imposed: (1) lower and/or upper bounds on specified coefficients, (2) bounds on $v(\alpha, t)$ at any specified points $\xi_j \in [t_1, t_n]$, (3) bounds on the first derivative of $v(\alpha, t)$, (4) bounds on the second derivative of $v(\alpha, t)$. These last two conditions include as special cases the possibility of requiring that $v(\alpha, t)$ be monotone and/or convex (or concave) in t.

After specification by the user the optimum coefficients are determined by an LP solution, and the original data points and the approximating curve are then displayed on the graphical scope. Other information, such as the coefficient values and the maximum error, is also available. Based on this graphical and numerical information, the user may modify his choice of approximating functions, the number of functions or the auxiliary conditions, while still at the terminal. This interactive modification enables the user to rapidly explore the possibilities and to obtain a satisfactory approximation to his data with a minimum of time and effort. The use of either the L_1 or L_∞ norm may be preferred to the usual L_2 norm in certain cases. For example, approximation in the L_1 norm tends to ignore outliers, while the L_∞ norm often permits one to obtain the best bounds on the absolute error in the approximation [15, 2, 3]. Advantages of an interactive system with graphical display for other statistical applications and for certain types of optimization are discussed in several recent papers [6, 1, 9].

In Section 2 the approximation problem is shown to be equivalent to a primal LP problem. In Section 3 the interactive graphical program is described, and several examples of its use are given in Section 4. User instructions for this system are given in the Appendix.

This on-line interactive program has been designed and implemented so as to be convenient and accessible to a user with a minimum of computer experience. It allows great flexibility to make changes in the functions, the error norm, the auxiliary conditions and even the data, until a satisfactory approximation is obtained.

2. LINEAR PROGRAMMING FORMULATION

Given a set of n data points (t_j, y_j), $j = 1, \ldots, n$, with $t_j \in [t_1, t_n]$, it is desired to determine a continuous function $v(\alpha, t)$ for $t \in [t_1, t_n]$, which approximates the data so as to minimize the error in a specified norm. The approximation $v(\alpha, t)$ is given by a linear combination of m selected functions $\varphi_i(t)$, $i = 1, \ldots, m$, continuous on $[t_1, t_n]$, so that

$$v(\alpha, t) = \sum_{i=1}^{m} \alpha_i \varphi_i(t) \tag{2.1}$$

where $\alpha \in E^m$ denotes a vector with elements α_i.

The two error norms considered are the discrete L_1 norm

$$\| v(\alpha, t) - y \|_1 = \sum_{j=1}^{n} | v(\alpha, t_j) - y_j | \tag{2.2}$$

and the discrete L_∞ norm

$$\| v(\alpha, t) - y \|_\infty = \max_j | v(\alpha, t_j) - y_j | \tag{2.3}$$

We first consider the formulation of the L_∞ norm minimization as an LP problem. We assume that the m functions $\varphi_i(t)$ have been chosen and we introduce a new scalar variable γ, and consider the problem

$$\min_{\alpha, \gamma} \left\{ \gamma \;\middle|\; \begin{array}{c} -\gamma \le v(\alpha, t_j) - y_j \le \gamma \\ j = 1, \ldots, n \end{array} \right\} \tag{2.4}$$

It is easy to see that if γ^* is an optimum solution to (2.4), then $\gamma^* = \| v(\alpha, t) - y \|_\infty$, since otherwise a smaller value of γ could be obtained without violating any constraints.

In order to obtain a concise statement of the problem we define the $m \times n$ matrix

$$F = \begin{bmatrix} \varphi_1(t_1) & \varphi_1(t_2) & \cdots & \varphi_1(t_n) \\ \varphi_2(t_1) & \varphi_2(t_2) & \cdots & \varphi_2(t_n) \\ \vdots & & & \\ \varphi_m(t_1) & \varphi_m(t_2) & \cdots & \varphi_m(t_n) \end{bmatrix} \qquad (2.5)$$

We will consider vectors to be defined as column vectors and denote the transpose of a matrix or vector by a prime. Thus α is a column vector and α' is a row vector. We let $y \in E^n$ be the vector with elements y_j, and let $c' = (y', -y')$. Also let $w' = (\alpha', \gamma) \in E^{m+1}$, and $b' = (0, \ldots 0, 1) \in E^{m+1}$. Finally let $e \in E^n$ denote the sum vector, $e' = (1, 1, \ldots 1)$, and

$$A = \begin{bmatrix} F & -F \\ e' & e' \end{bmatrix} = (m+1) \times 2n \qquad (2.6)$$

Then we can write (2.4) in the concise form

$$\min_w \; \{b'w \mid A'w \geq c\} \qquad (2.7)$$

If we consider this as the unsymmetric dual LP problem [7], the equivalent primal problem in terms of a primal vector $x \in E^{2n}$ is given by

$$\max_x \left\{ c'x \;\middle|\; \begin{matrix} Ax = b \\ x \geq 0 \end{matrix} \right\} \qquad (2.8)$$

This primal problem has only $m+1$ rows, and is in the form suitable for the most efficient solution by the ALPS LP system [4]. The optimal solution to the primal problem also gives an optimal nonsingular $(m+1)\times(m+1)$ basis matrix B, and its inverse B^{-1}. The optimal dual vector is then given by $w = (B^{-1})'\bar{c}$, where $\bar{c} \in E^{m+1}$ consists of those elements of c which correspond to the optimal basis activities. The first m elements of w give the desired coefficient vector α, and the last element gives the minmax error γ.

The formulation of the L_1 norm minimization as an LP problem is similar. We now consider the problem

$$\min_{\alpha, \gamma} \left\{ \sum_{j=1}^{n} \gamma_j \;\middle|\; \begin{array}{c} -\gamma_j \le v(\alpha, t_j) - y_j \le \gamma_j \\ j = 1, \dots n \end{array} \right\} \qquad (2.9)$$

where $\gamma \in E^n$ is now a vector with elements γ_j. Again it can be seen that if γ^* is the optimal vector obtained by (2.9) we have

$$|v(\alpha, t_j) - y_j| = \gamma_j^*, \quad j = 1, \dots n$$

and therefore

$$\sum_{j=1}^{n} \gamma_j^* = \|v(\alpha, t) - y\|_1 \qquad (2.10)$$

We now define $b' = (0, \dots 0, e') \in E^{m+n}$, $w' = (\alpha', \gamma') \in E^{m+n}$, and

$$A = \begin{bmatrix} F & -F \\ I_n & I_n \end{bmatrix} = (m+n) \times 2n \qquad (2.11)$$

Then it follows that (2.9) is represented by (2.7). We again solve the equivalent primal given by (2.8) and obtain the optimal dual vector w as part of the optimal primal solution. It has been shown [15, 2] that this can be reduced to an upper-bounded variable LP problem in only m

constraints. Since the ALPS code does not have the upper-bounded variable capability this further reduction was not used.

In the remainder of this section we show how auxiliary conditions on the approximation can be imposed as part of the LP problem. Since this is done in essentially the same way for both the L_∞ and L_1 approximations, only the L_∞ case will be discussed here. There are four types of auxiliary conditions to be considered.

Lower and/or upper bounds may be placed on any of the coefficients. If ℓ_k and/or μ_k are given and it is desired that

$$\ell_k \le \alpha_k \le \mu_k \quad \text{for any } k \in \{1, 2, \ldots, m\},$$

then the following constraints are added to the dual LP problem:

$$\begin{bmatrix} e'_k & 0 \\ -e'_k & 0 \end{bmatrix} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix} \ge \begin{pmatrix} \ell_k \\ -\mu_k \end{pmatrix}$$

where e_k is a unit vector with unity as its k^{th} element.

The corresponding primal problem would have additional columns in A,

$$A = \begin{bmatrix} F & -F & e_k & -e_k \\ e' & e' & 0 & 0 \end{bmatrix} \qquad (2.12)$$

and additional elements in c',

$$c' = (y' \quad -y' \quad \ell_k \quad -\mu_k)$$

Each bound, lower or upper, adds a column to the primal problem (2.8).

Lower and/or upper bounds may be placed on the first derivative with respect to t, of the approximation $v(\alpha, t)$ at all the points $t_1, t_2, \ldots, t_n$. Let $\varphi_i^{(1)}(t)$ denote the first derivative of $\varphi_i(t)$, and let ℓ and/or μ be given. Suppose that it is desired that

$$\ell \leq \sum_{i=1}^{m} \alpha_i \varphi_i^{(1)}(t_j) \leq \mu \quad \text{for} \quad j = 1, 2, \ldots, n \; .$$

These derivative bounds are imposed by adding $2n$ columns in the primal problem. Define the $m \times n$ matrix $F^{(1)}$ as in (2.5) with $\varphi_i^{(1)}(t_j)$ replacing $\varphi_i(t_j)$. Then let

$$A = \begin{bmatrix} F & -F & F^{(1)} & -F^{(1)} \\ e' & e' & 0 & \ldots 0 \end{bmatrix} \tag{2.13}$$

and

$$c' = [y' \quad -y' \quad \underbrace{\ell \ldots \ell}_{n} \underbrace{\mu \ldots \mu}_{n}]$$

The optimal solution to (2.8) will now give $v(\alpha, t)$ satisfying the auxiliary derivative bounds. If only lower or upper bounds are desired it is necessary to add only n columns. For an appropriate choice of the φ_i, imposing these bounds at the points t_j will insure that $v(\alpha, t)$ satisfies similar derivative bounds for all $t \in [t_1, t_n]$. In particular $\ell > 0$ gives $v(\alpha, t)$ monotone increasing and $\mu < 0$ gives $v(\alpha, t)$ monotone decreasing.

Lower and/or upper bounds may be placed on the second derivative of the approximation $v(\alpha, t)$ at all the points $t_1, t_2, \ldots, t_n$, in the same manner. For such bounds $F^{(2)}$ replaces $F^{(1)}$ in (2.13), where the elements of $F^{(2)}$

are $\varphi_i^{(2)}(t_j)$, the second derivatives of the φ_i. Special cases are again obtained when $\ell \geq 0$ giving $v(\alpha, t)$ convex, and $\mu \leq 0$ giving $v(\alpha, t)$ concave.

Finally we can impose a lower and/or upper bound on the approximation $v(\alpha, t)$ at any set of points $t = \xi_j$. Thus if ℓ_{ξ_j} and/or μ_{ξ_j} are given and it is desired that

$$\ell_{\xi_j} \leq \sum_{i=1}^{m} \alpha_i \varphi_i(\xi_j) \leq \mu_{\xi_j},$$

then one or two columns are added to A and corresponding elements to c' in the primal problem for each point ξ_j. For example, if lower and upper bounds are imposed at one point ξ we get

$$A = \begin{bmatrix} F & -F & \varphi(\xi) & -\varphi(\xi) \\ e' & e' & 0 & 0 \end{bmatrix}$$

$$c' = (y' \quad -y' \quad \ell_\xi \quad -\mu_\xi)$$

where

$$\varphi(\xi) = \begin{pmatrix} \varphi_1(\xi) \\ \vdots \\ \varphi_m(\xi) \end{pmatrix} \in E^m$$

It should be noted that all four of these auxiliary conditions increase the number of columns in the primal problem, but do not change the number of rows. Therefore, these additional conditions will normally increase only slightly the number of iterations required to solve the primal LP problem.

3. INTERACTIVE PROGRAM

The program, which shall be referred to as CURVFIT, is written in ALGOL and can be executed from any of the teletype terminals linked to the Burroughs 5500. However, if the graphic display is to be used, a teletype which is located next to it must be selected.

When CURVFIT is executed it will request the user to type in the data to be fitted. When all of the data points have been received CURVFIT will both display them on the graphic unit and type them out on the teletype. This allows the user to check for possible errors. If any errors are found they may be corrected, and this process may be continued until the user is satisfied with the data.

The user is then requested to type in the functions $\varphi_i(t)$ that he wishes to use in approximating the data. The following class of functions, which will be called elementary functions, are accepted. Let an elementary function be defined to be a function which is obtained by addition, subtraction, multiplication, division, exponentiation, and composition starting with the real variable t, the rational numbers, and the functions e^v, $\sin(v)$, $\cos(v)$, $\arctan(v)$, $\ln(v)$, and $|v|$ where v may itself be an elementary function. The following is an example of an elementary function

$$\frac{1}{3} + \frac{e^{\sin(t)}}{(t^2 + \arctan(\ln(t)))} .$$

Note that quite a large class of functions can be represented by the above elementary functions. (For a more general definition of elementary functions see [8]. The user may also add to, delete, or change any of the functions he has previously typed in. If the user types in a function which is not recognized as an elementary function, CURVFIT will reject it and ask the user to try again.

A class of functions which are known to be important in approximation are spline functions [14, 5]. These are not now included as part of the system capability, but it is planned to add splines to the available class of functions.

Several options are available to the user. He may instruct CURVFIT to give him the best approximation to the data in the sense of the L_∞ norm or the L_1 norm. If he wishes to bound any of the coefficients α_i of the functions he can type in the desired bounds. If he wishes to constrain the first or second derivative of the resultant approximation $v(\alpha, t)$, he must type in the desired bounds and the first or second derivative of all of the functions $\varphi_i(t)$. As shown in Section 2, this will guarantee that the first or second derivative of the approximation lies between the specified bounds at the points $t_1, t_2, \ldots, t_n$. The derivative may or may not be within the specified bounds between the points. This depends on the number and position of the data points and the "smoothness" of the functions $\varphi_i(t)$. The user may also constrain the approximation $v(\alpha, t)$ at any other point ξ by typing in ξ and the desired bounds. The possibility of improving the approximation by imposing one or more of the above types of auxiliary conditions is illustrated in Section 4.

Given the above information (the data points, functions, the norm, and any additional conditions) CURVFIT generates the appropriate LP problem as input for a LP code called ALPS [4]. CURVFIT then puts this input on the magnetic disc, initiates the execution of ALPS, and puts itself "to sleep", checking intermittently to see if ALPS has finished. When ALPS has obtained an optimal solution, CURVFIT retrieves the results which ALPS has placed on the disc and presents them to the user.

In this manner, the "dirty work" of generating and solving the LP problem is accomplished without any intervention by the user. When the B5500 is not too busy, the user will seldom have to wait more than three minutes before receiving the results*. CURVFIT prints out the coefficients $\alpha_1, \alpha_2, \ldots, \alpha_m$ and the minimized value of the error norm. If the L_1 norm is being used, the user may have the absolute deviations γ_j, $j = 1, \ldots, n$ printed out. The graph of the "best" approximation $v(\alpha, t)$ is superimposed on the data points which are still being displayed on the graphic unit.

After examining the displayed points and the approximating curve, the user may change any of the functions or constraints (or even the data points) and have CURVFIT try again. This on-line interactive process may be repeated until the user is satisfied with the approximation obtained. If the user wishes to obtain a hard copy of the information displayed on the graphic

---

*A large number of batch jobs will not make this wait-time appreciably longer, but if there are five to ten remote jobs queued up waiting to be executed, it may take fifteen or twenty minutes before ALPS is executed. This can be avoided by running during non-peak hours.

unit he may instruct CURVFIT to write this information out on a magnetic tape, which can later be plotted with a Calcomp Plotter. In its present form, CURVFIT allows a maximum of fifty data points and twenty functions, but these limits can be easily expanded.

4. EXAMPLES

In this section we discuss three examples in order to show how the system operates and to illustrate the effect of imposing auxiliary conditions.

Example 1

The data shown in Table 1A represents the function $e^t \pm R \times 10^{-2}$ where R is a random integer, $0 \le R \le 9$. After receiving these data points, CURVFIT displayed the graph in Figure 1A. The functions $e^{.8t}$, $e^{.9t}$, e^t, $e^{1.1t}$, and $e^{1.2t}$ were typed in and CURVFIT was instructed to find the best fit in the L_∞ norm with no additional conditions. After 1.5 minutes the coefficients α_i shown in the third column of Table 1B were typed out and the graph in Figure 1B was displayed. This approximating curve, which has a maximum deviation of .0650, appears to be reasonable. However the large coefficient values are not reasonable, and reflect the fact that the φ_i chosen are almost linearly dependent. The same problem was then solved with the additional condition that all the coefficients must be non-negative. After 2.5 minutes CURVFIT displayed the graph in Figure 1C and typed out the coefficients shown in the last column of Table 1B. This approximating curve, which has a maximum deviation of .0771, has coefficients that are much more reasonable. Note that in this case only two of the five functions are being used. Furthermore, the sum $\alpha_1 + \alpha_5 = 1.0134$, so that $v(\alpha, t)$ is very close to e^t, the unperturbed function.

j	t_j	y_j	j	t_j	y_j
1	0.00	0.988	12	0.55	1.657
2	0.05	1.140	13	0.60	1.830
3	0.10	1.170	14	0.65	1.839
4	0.15	1.124	15	0.70	1.945
5	0.20	1.166	16	0.75	2.088
6	0.25	1.297	17	0.80	2.172
7	0.30	1.378	18	0.85	2.376
8	0.35	1.367	19	0.90	2.488
9	0.40	1.411	20	0.95	2.603
10	0.45	1.573	21	1.00	2.756
11	0.50	1.709			

Table 1A. Data for Example 1.

i	$\varphi_i(t)$	α_i	$\alpha_i \geq 0$
1	$e^{.8t}$	-4668.9107	.6265
2	$e^{.9t}$	17686.2941	0.0
3	e^{t}	-25109.2169	0.0
4	$e^{1.1t}$	15836.1280	0.0
5	$e^{1.2t}$	-3743.2838	.3869
max. error γ		.0650	.0771

Table 1B. Approximation for Example 1.

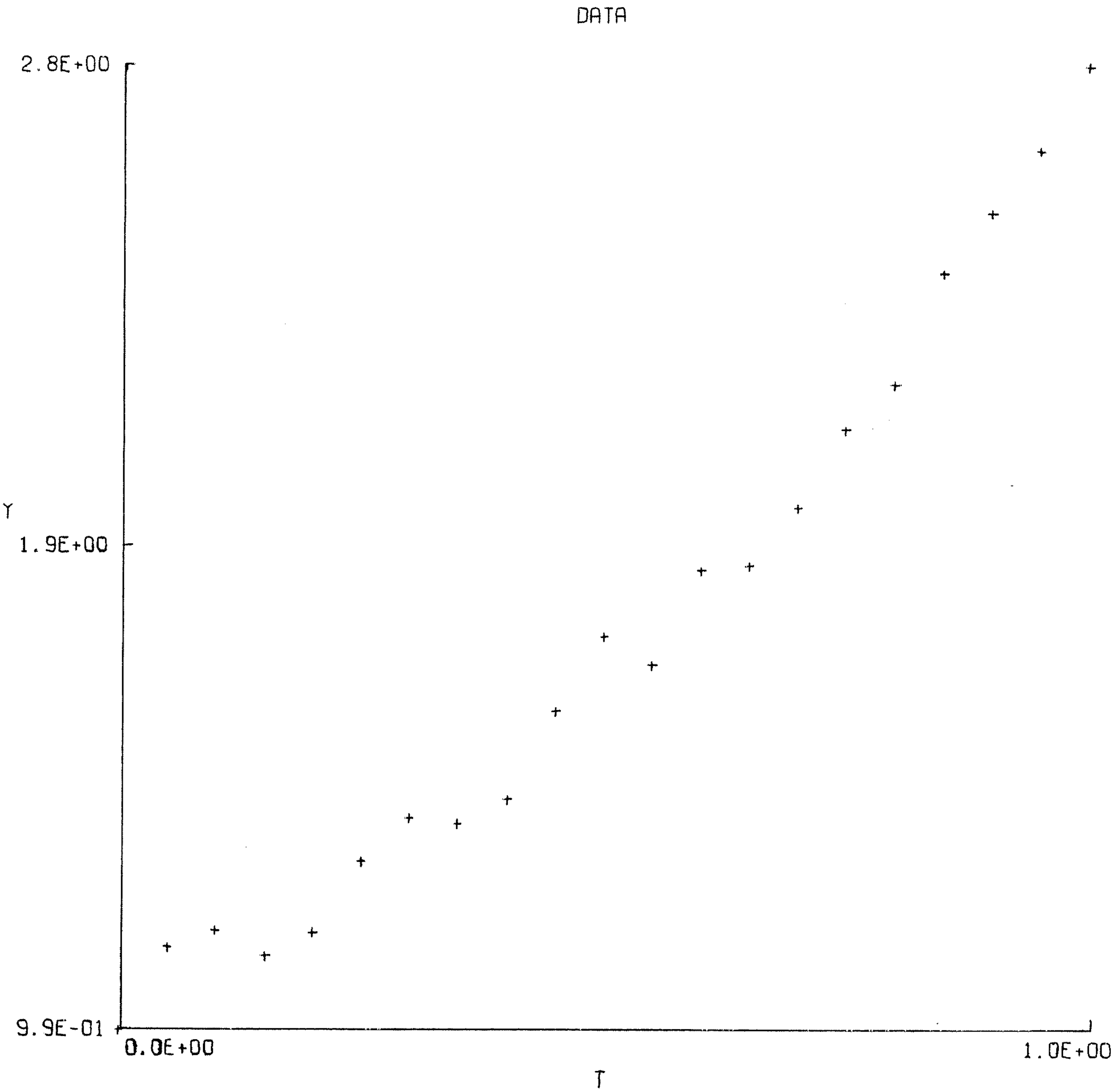

Figure 1A. Data for Example 1.

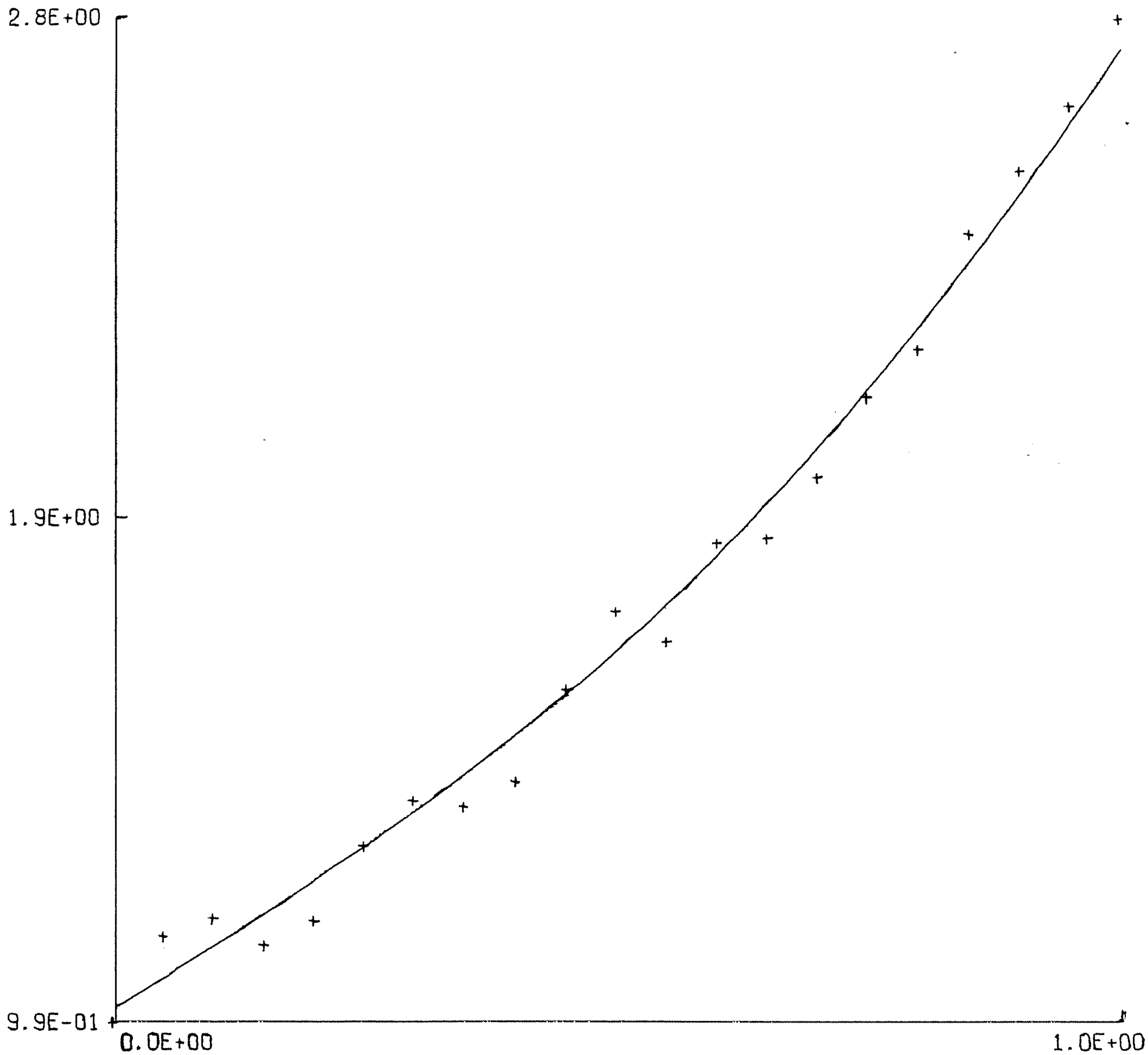

Figure 1B. L_∞ Approximation with no Conditions for Example 1.

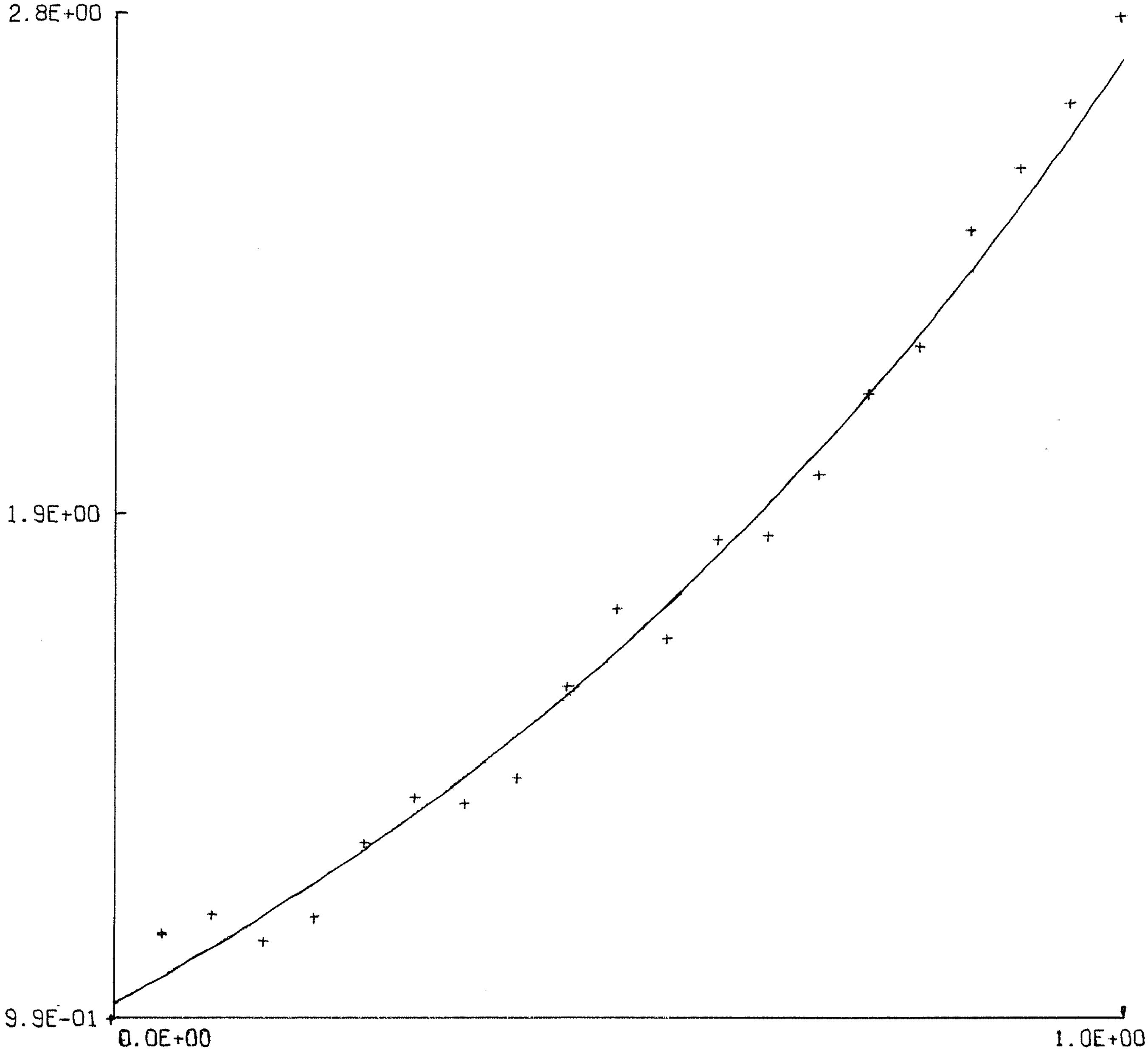

Figure 1C. L_∞ Approximation with $\alpha_i \geq 0$ for Example 1.

Example 2.

The data points shown in Table 2A were typed in. CURVFIT then displayed the graph shown in Figure 2A. CURVFIT was first instructed to find the best fit in the L_∞ norm using a polynomial of degree 7 with no additional conditions. After 2.7 minutes the coefficients in the third column of Table 2B were typed out and the approximation in Figure 2B, which has a maximum deviation of .3422, was displayed. It was desired to eliminate the "camel back" shape of this curve. Therefore a concavity requirement was imposed by means of an upper bound $\mu = 0$ on the second derivative of $v(\alpha, t)$, as discussed in Section 2. After 2.7 minutes the coefficients shown in the last column of Table 2B were printed out and the approximating curve in Figure 2C, with maximum deviation of .3789, was displayed. It is seen that the unwanted hump has disappeared.

j	t_j	y_j	j	t_j	y_j
1	0.00	0.0	12	0.55	6.2
2	0.05	2.4	13	0.60	6.3
3	0.10	2.7	14	0.65	6.4
4	0.15	3.5	15	0.70	5.8
5	0.20	4.5	16	0.75	6.3
6	0.25	5.2	17	0.80	5.9
7	0.30	5.5	18	0.85	6.0
8	0.35	6.0	19	0.90	5.5
9	0.40	5.6	20	0.95	3.6
10	0.45	6.2	21	1.00	0.5
11	0.50	5.9			

Table 2A. Data for Example 2.

i	$\varphi_i(t)$	α_i	α_i when $v''(\alpha, t) \le 0$
1	1	0.3422	0.3789
2	t	46.2386	41.0742
3	t^2	-298.2415	-193.5385
4	t^3	1360.5003	633.7902
5	t^4	-3450.9916	-1156.7482
6	t^5	4620.4595	1014.8607
7	t^6	-3045.7410	-295.4402
8	t^7	768.2757	-43.4981
max.	error γ	.3422	.3789

Table 2B. Approximation for Example 2.

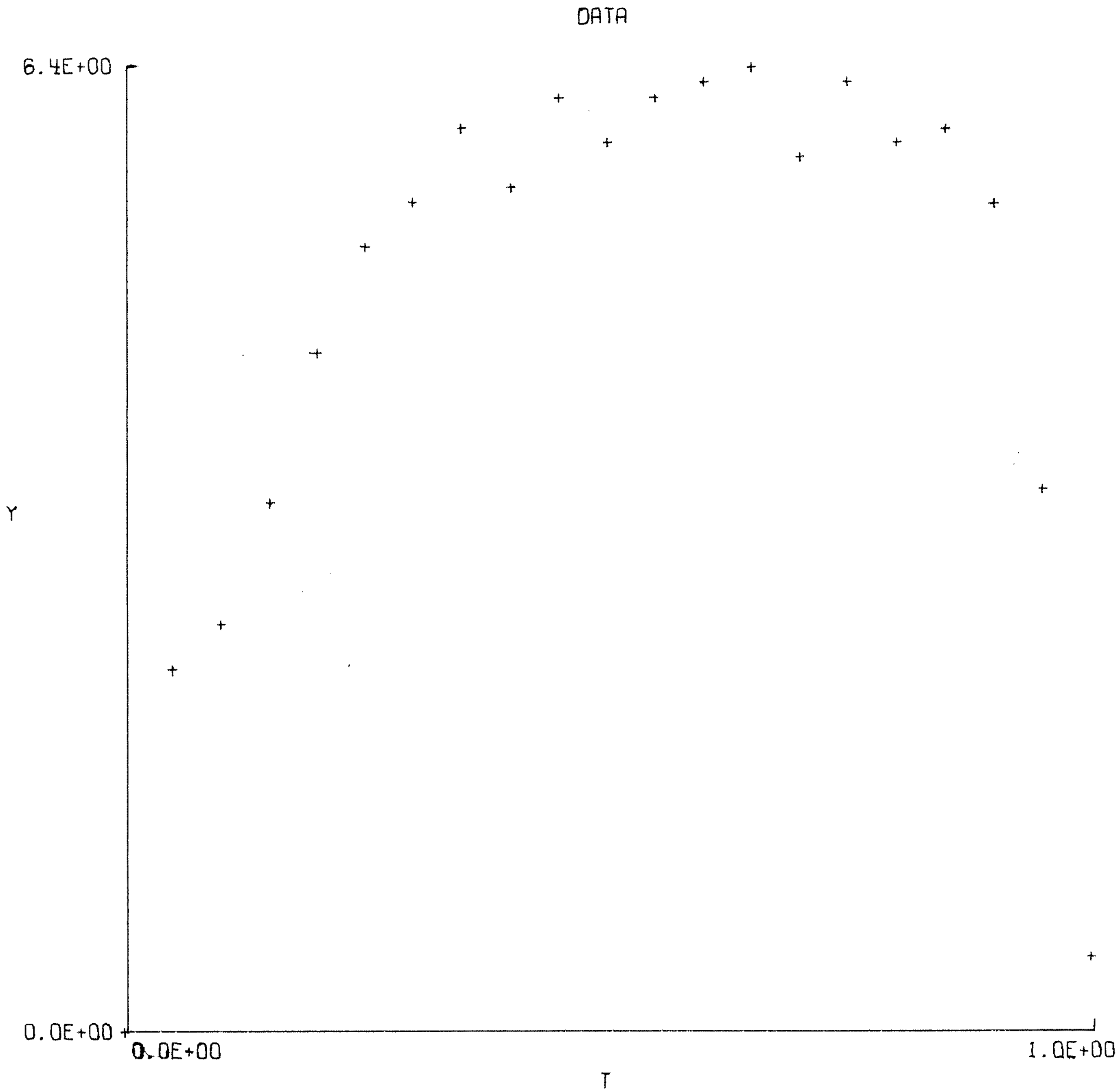

Figure 2A. Data for Example 2.

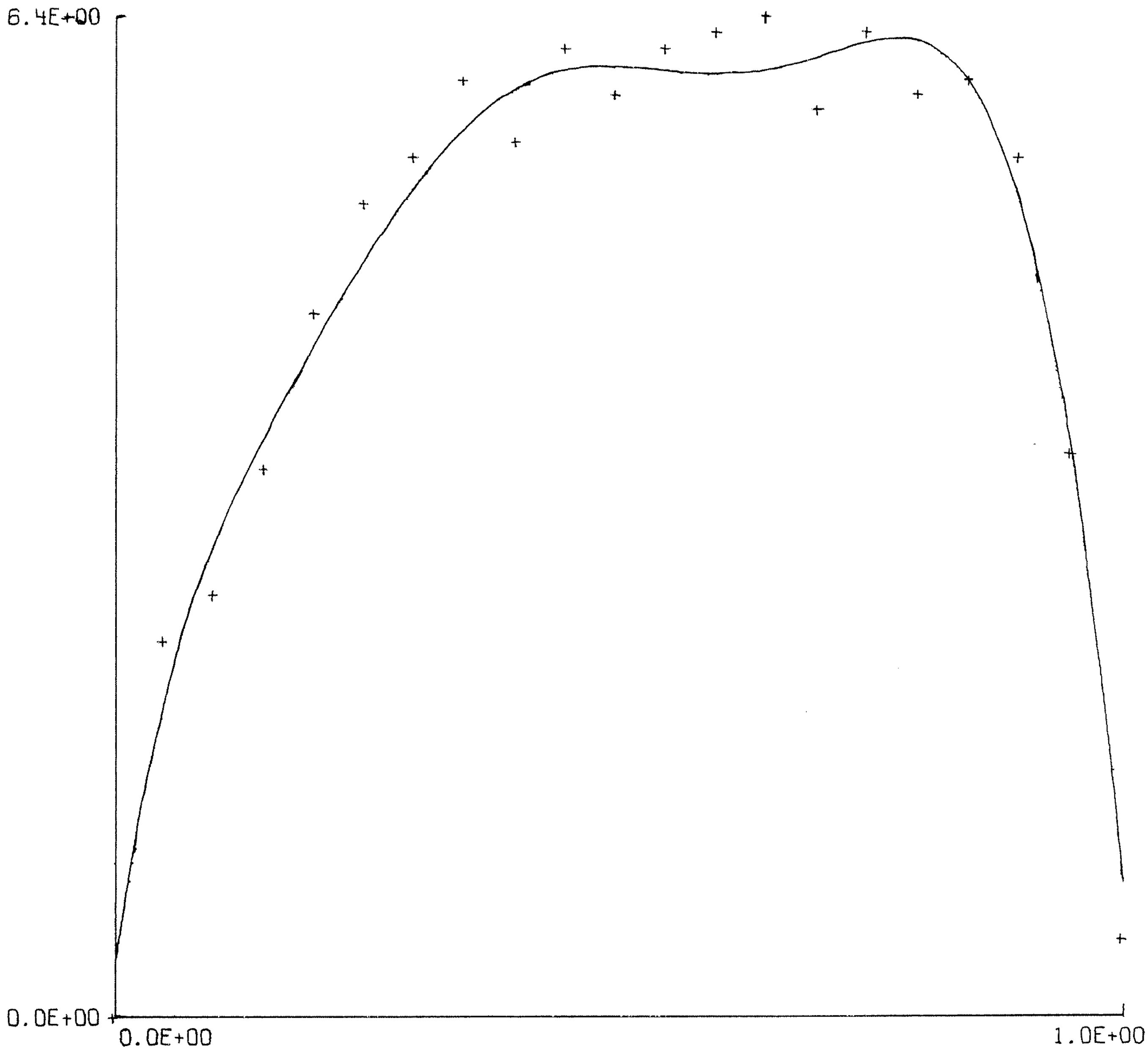

Figure 2B. L_∞ Approximation with no Conditions for Example 2.

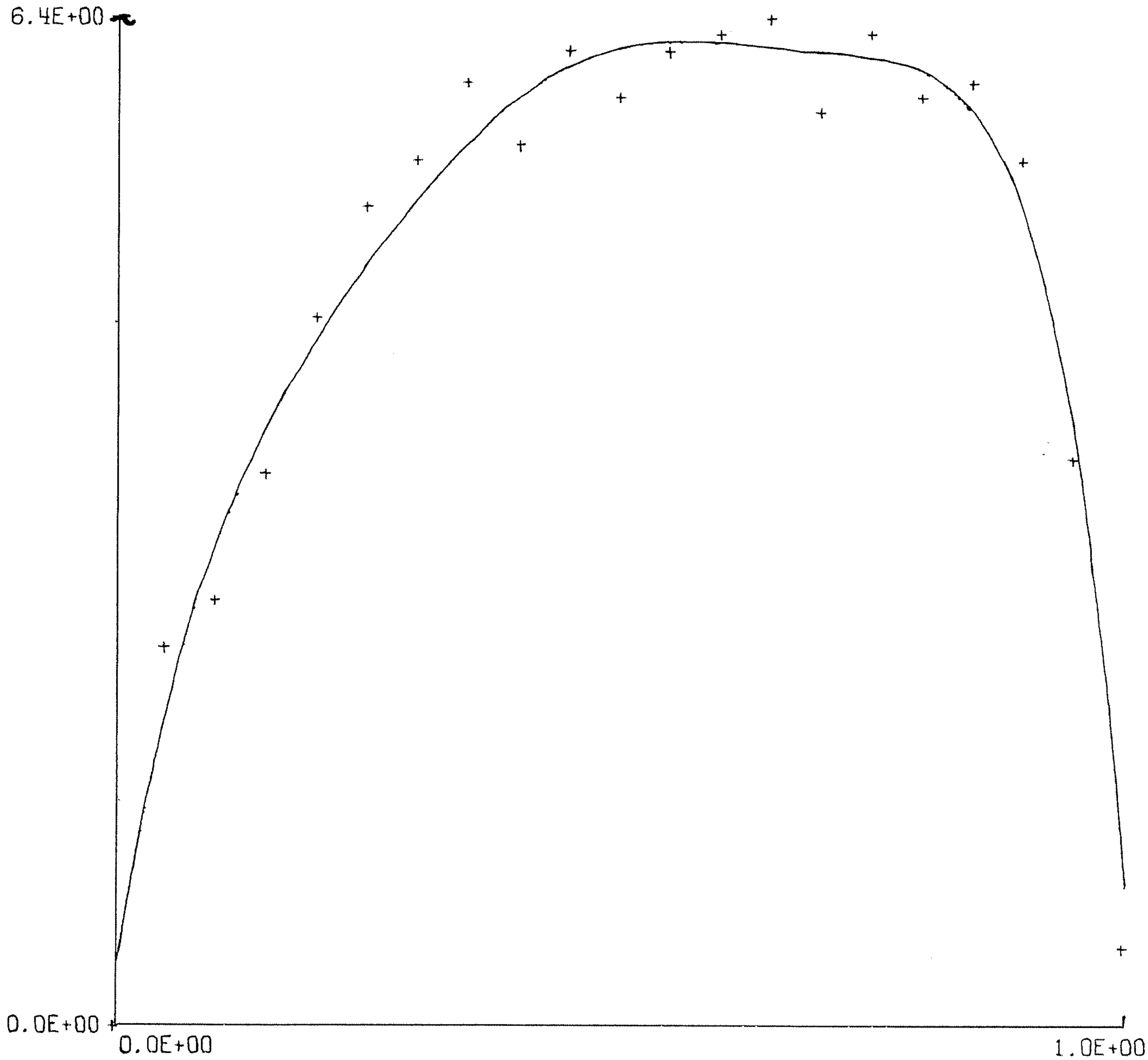

Figure 2C. L_∞ Approximation with Concavity Condition for Example 2.

Example 3.

The data points used for the example were obtained from a test problem proposed in [11], and are given in Table 3A. The data points were displayed by CURVFIT as shown in Figure 3A. CURVFIT was first instructed to find the best fit in L_∞ norm using a polynomial of degree 10 with no additional constraints. After 1.7 minutes the coefficients in the third column of Table 3B were types out and the approximation in Figure 3B, which has a maximum deviation of .0294, was displayed. In order to increase the "smoothness" of this curve a monotonicity requirement was imposed by means of a lower bound $\ell = 0$ on the first derivative of $v(\alpha, t)$, and CURVFIT was instructed to find the best fit in the L_1 norm using a polynomial of degree 10. After 2.6 minutes the coefficients shown in the last column of Table 3B were printed out and the approximating curve in Figure 3C, with maximum deviation of .0971 and L_1 error of .3607, was displayed.

j	t_j	y_j	j	t_j	y_j
1	0.00	0.431	14	0.52	0.669
2	0.04	0.409	15	0.56	0.746
3	0.08	0.429	16	0.60	0.760
4	0.12	0.422	17	0.64	0.778
5	0.16	0.530	18	0.68	0.828
6	0.20	0.505	19	0.72	0.846
7	0.24	0.459	20	0.76	0.836
8	0.28	0.499	21	0.80	0.916
9	0.32	0.526	22	0.84	0.956
10	0.36	0.563	23	0.88	1.014
11	0.40	0.587	24	0.92	1.076
12	0.44	0.595	25	0.96	1.134
13	0.48	0.647	26	1.00	1.124

Table 3A. Data for Example 3

i	$\varphi_i(t)$	α_i	α_i when $v'(\alpha, t) \geq 0$
1	1	0.4016	0.4300
2	t	-1.9340	0.0
3	t^2	43.9516	1.6323
4	t^3	-233.2457	-53.2872
5	t^4	-11.4299	613.8906
6	t^5	4073.3973	-3125.5384
7	t^6	-15799.1401	8672.5449
8	t^7	29007.0965	-14006.6647
9	t^8	-29117.4944	13128.6814
10	t^9	15423.7168	-6616.1851
11	t^{10}	-3384.2252	1385.6326
max. error γ		.0294	.0971

Table 3B. Approximation for Example 3.

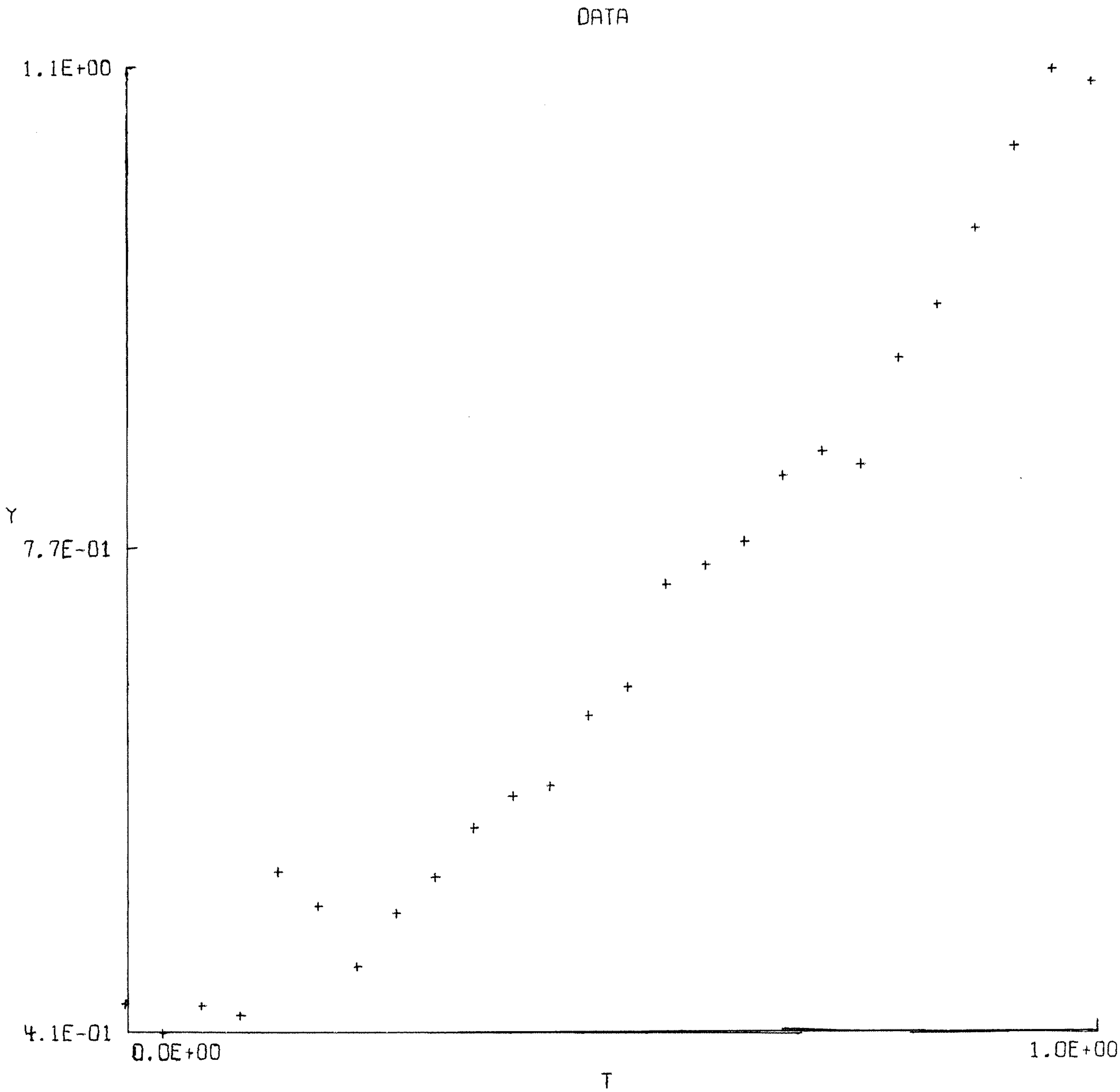

Figure 3A. Data for Example 3.

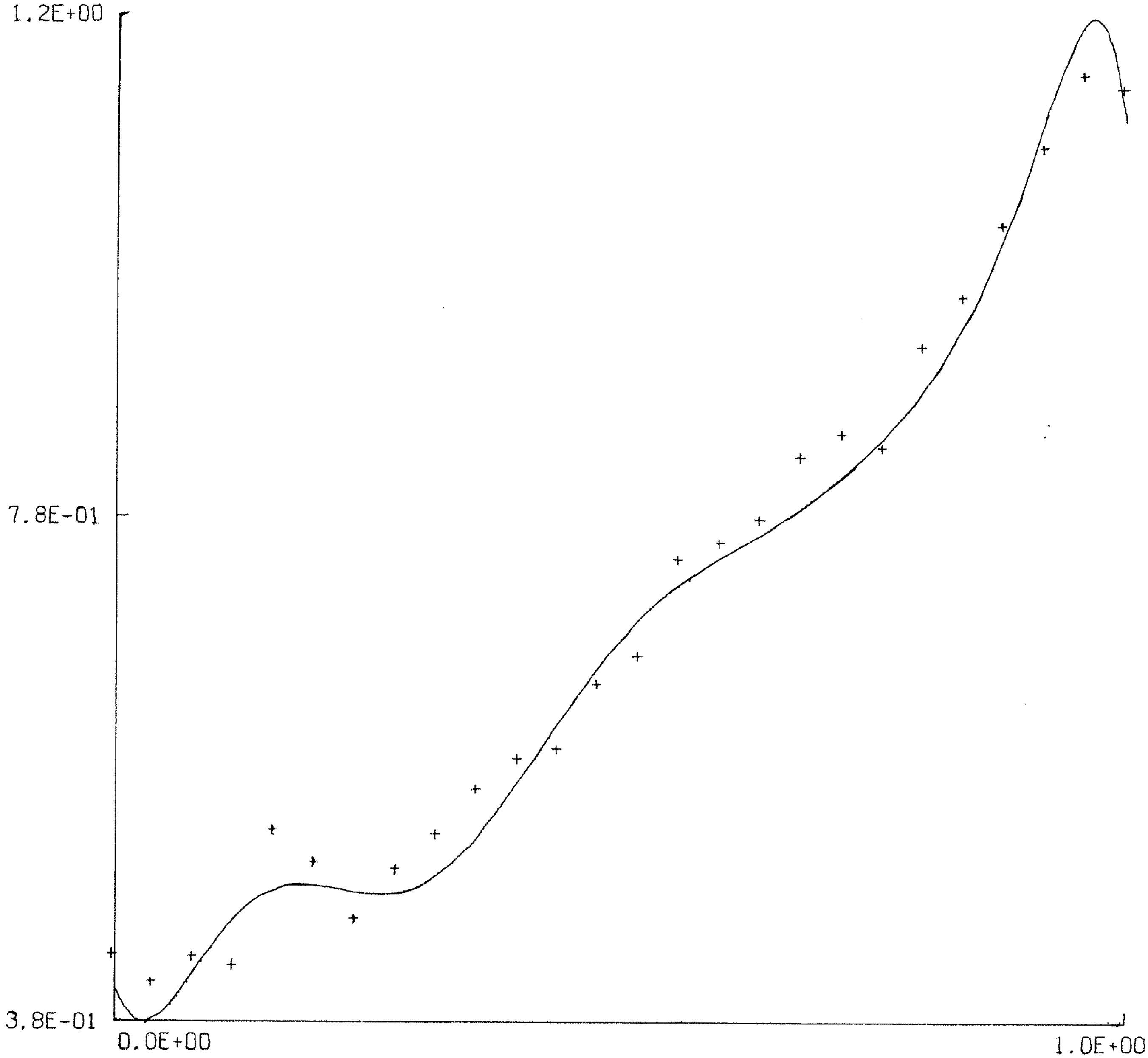

Figure 3B. L_∞ Approximation with no Conditions for Example 3.

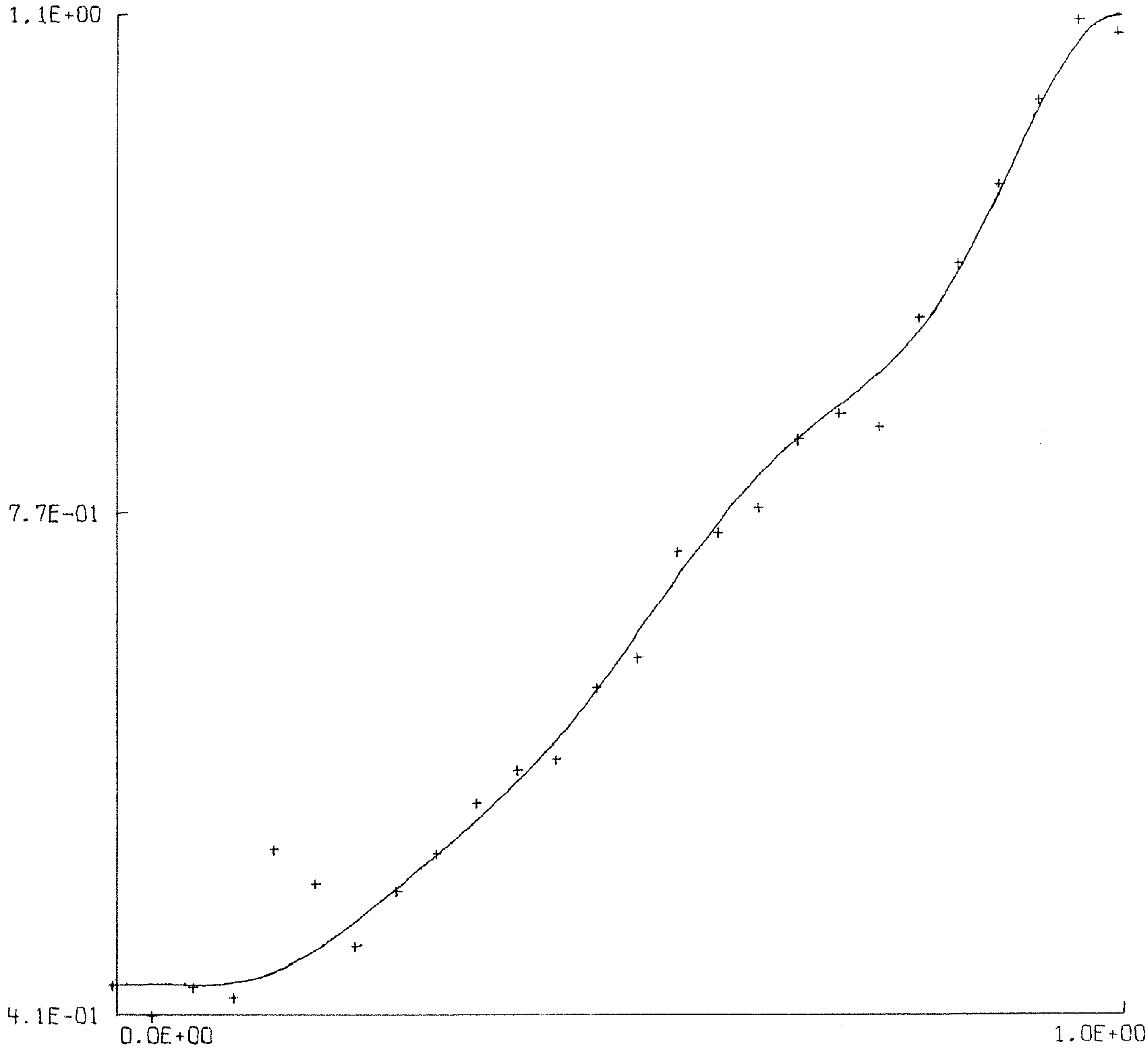

Figure 3C. L_1 Approximation with Monotonicity Condition for Example 3.

APPENDIX. USER INSTRUCTIONS

Since the program CURVFIT has been designed so that it would be easy to use, a User Manual is not really necessary. There are, however, a few things that should be mentioned.

1. Before executing CURVFIT make sure that CURVFIT/ALPS is on the disc. It is this program that CURVFIT uses to solve the LP problem.

2. To execute CURVFIT type:

?? EXECUTE CURVFIT/CURVFIT←

3. The first question CURVFIT will ask is IS THE GRAPHIC DOWN. If one replies NO, then the graphic will be used for output. If the reply is YES, then the user must give a magnetic tape to the operator with the instructions that it is a plotter tape and is to be mounted at a density of 200 BPI. In this case the graphic will not be used.

4. When requesting the function $\varphi_i(t)$, CURVFIT will type the index number i of that function. Next to this index number, on the same line, the user should type the desired function. Later on, if any function is to be deleted or changed, the user can indicate to CURVFIT which function is to be altered by typing its index number.

5. When the user is finished typing the set of data points or functions he should type DONE.

6. All questions, unless explicitly stated otherwise, should be answered with YES or NO.

7. If the user types a blank when CURVFIT requests the upper or lower bounds for the first or second derivative of the approximation, then no bound will be enforced.

8. CURVFIT will accept numbers of the following three types:

TYPE	EXAMPLES
INTEGER	216, -72
FLOATING POINT	0.001, -532.763
SCIENTIFIC	6.007E+07, -12.42E-22

The arithmetic operations which are accepted by CURVFIT and the symbols which must be used are listed in the following table. The operations are listed in increasing order of precedence.

SYMBOL	OPERATION
+, -	addition, subtraction
\, /	multiplication, division
-	negation (unary)
*	exponentiation
SIN, COS, ARCTAN, LN, FXP, ABS	sin( ), cos( ), arctan( ), ln( ), $e^{(\ )}$, absolute value

Parentheses may be, and should be used freely if there is any doubt about the precedence relationships of the operators. For example, the function on page 10 would be accepted by CURVFIT if it were typed in the following form:

```
1/3 + EXP(SIN(T) )/(T*2 + ARCTAN(LN(T) ) )
```

Note that the variable in all of the functions $\varphi_i(t)$ must be T .

REFERENCES

1. Ball, G. H. and D. J. Hall, "Promenade - an on-line pattern recognition system", Tech. Rept. No. RADC-TR-67-310, Stanford Research Inst., Sept. 1967.

2. Barrodale, I. and Young, A., "Algorithms for best L_1 and L_∞ linear approximations on a discrete set", Numerische Mathematik 8, 295-306 (1966).

3. Barrodale, I., "Approximation in the L_1 and L_∞ norms by linear programming", Ph.D. Thesis, Univ. of Liverpool, 1967.

4. Burroughs Corporation, "Burroughs algol linear programming system - ALPS-1", 1967.

5. Curry, H. B. and I. J. Schoenberg, "On Polya frequency functions IV: the fundamental spline functions and their limits", J. d'Analyse Math. 17, 71-107 (1966).

6. Dixon, W. J., "Use of displays with packaged statistical programs", AFIPS Conference Proceedings 31, 481-484, (1967).

7. Hadley, G., Linear Programming, Addison-Wesley, 1962. p. 237.

8. Hardy, G. H., "The integration of functions of a single variable", in Cambridge Tracts in Mathematics and Mathematical Physics (edited by P. Hall and F. Smithies), 1966.

9. Kirby, D. B., "Optimization using a remote control console for a digital computer", Simulation 7, 36-41 (1966).

10. Rabinowitz, P., "Applications of linear programming to numerical analysis", SIAM Review 10, 121-159 (1968).

11. Ralston, A., A First Course in Numerical Analysis, McGraw Hill, 1965. pp. 250-254.

12. Rosen, J. B., "Approximate computational solution of nonlinear parabolic partial differential equations by linear programming", Numerical Solutions of Nonlinear Differential Equations (D. Greenspan, Ed.), Wiley, 1966. pp. 265-296.

13. Rosen, J. B. "Approximate solution and error bounds for quasilinear elliptic boundary value problems", Comp. Sci. Dept. Tech. Rept. #30, Univ. of Wis., Nov. 1968.

14. Schoenberg, I. J., "On variation diminishing approximation methods", Proc. of MRC Symposium "On Numerical Approximation", Madison, Wis. (1958). pp. 249-274. See also M. Marsden and I. J. Schoenberg, Mathematica 8, 61-82 (1966).

15. Wagner, H. M., "Linear programming techniques for regression analysis", J. Amer. Statist. Assoc. 54, 206-212 (1959).