The University of Wisconsin
Computer Sciences Department
1210 West Dayton Street
Madison, Wisconsin 53706

Interactive Graphical Spline Approximation to Boundary Value Problems

by

J. B. Rosen and Paul LaFata

Technical Report #111

February 1971

Interactive Graphical Spline Approximation to Boundary Value Problems

by

J. B. Rosen and Paul LaFata

Abstract

Earlier work on interactive graphical approximation of data using linear programming has now been extended to ordinary differential equation multipoint boundary value problems. The approximation is obtained using a suitable spline basis where the degree and uniform knot size is specified by the user. The coefficients of the spline basis are determined so as to minimize the maximum error in the differential equation over a specified discrete grid.

This research was supported in part by the National Science Foundation under grant GJ-0362.

I. INTRODUCTION

Interactive graphical systems have proven to be powerful tools in the area of numerical analysis. This has been demonstrated by a number of general purpose mathematical analysis systems and also by some more specific systems for the approximation of data, functions, and differential equations [3,6,4]. For a fairly extensive survey of interactive graphical systems for mathematics, the reader is referred to [10]. In this paper we describe an interactive graphical system, DIFEQ, which allows us to approximate a specified function or obtain an approximate solution to a linear differential equation. One of the important characteristics of this system is that it allows the user to place various auxiliary conditions on the desired approximation. These conditions include lower and/or upper bounds on the approximation, or its first or second derivative, at specified points. A special case of this is the possibility of requiring that the approximation be monotone and/or convex (or concave).

Linear programming (LP) is the mathematical technique which allows us to find an approximation to the above types of problems while enforcing various auxiliary conditions. This has been discussed by a number of authors for the approximation of functions and approximate solution of ordinary differential equations [7,2] and also for the

approximate solution of certain types of partial differential equations [8,9]. In all of these cases the approximation $v(\alpha,x)$ is obtained as a linear combination of selected functions: $v(\alpha,x) = \sum_{i=1}^{m} \alpha_i \varphi_i(x)$.

Experience with a variety of possible choices indicates that spline functions offer an excellent compromise between complete generality and ease of use in this type of interactive program [1,5]. For this reason the system is limited to this class of functions. The user may however, very easily, specify the degree and number of basis splines he wishes to be used. The user also states the differential equation (or function) to be approximated, together with any auxiliary conditions he wishes to impose. The "best" coefficients α_i will then be obtained by the system using LP to minimize the maximum error in the differential equation. Appropriate graphical results are then displayed on the Adage scope. If the user feels, on the basis of this display, that the approximation has some undesirable characteristics, he can modify the auxiliary conditions, the degree or number of splines, or other parameters and try again. In this interactive way he can easily and rapidly explore various approximations until he has obtained a satisfactory approximation.

It should be remarked that the computing power of the 1108, or a comparable computer, is needed to solve the LP problem in a reasonable time (10 seconds or less). Because of the limited computing power of the graphics terminal, a high speed data channel connection between the terminal and the 1108 is essential for interactive use of this system.

In Section II we briefly give a mathematical description of the approximation problem, and the LP formulation using a suitable spline basis. Section III contains a description of the program DIFEQ and gives four examples of its use. The utility of an auxiliary condition (convexity in this case) for a differential equation is illustrated. The system is implemented on the ADAGE Graphics Terminal (AGT/10) and Univac 1108 computers.

II. MATHEMATICAL FORMULATION

In this section we will consider the class of problems on the interval $I = [a,b]$ given by

$$L[u] \equiv u^{(q)} + \sum_{k=0}^{q-1} a_k(x)\, u^{(k)} = f(x) \qquad (2.1)$$

where L is a linear differential operator and $q \in \{0,1,\ldots,4\}$. For $1 \le q \le 4$, this is a linear differential equation of order q. If $q = 0$, we understand that (2.1) becomes $u = f(x)$, so that we have the usual approximation problem. In the rest of this discussion we shall refer to (2.1) as a differential equation, with $q = 0$ as a special case. In general, when given a qth order differential equation, q initial value or boundary conditions are also required. These will be of the form

$$g_\ell[u(x_\ell)] = c_\ell, \quad \ell = 1,\ldots,q \qquad (2.2)$$

where g_ℓ, $\ell = 1,\ldots,q$ are linear boundary operators to be discussed later.

We wish to obtain a function $v(\alpha,x)$ which will approximate $u(x)$, the exact solution of (2.1), assuming the solution exists and is unique. The approximation

$$v(\alpha,x) = \sum_{i=1}^{m} \alpha_i \varphi_i(x) \qquad (2.3)$$

will be given by a linear combination of m selected functions $\varphi_i(x)$, $i = 1,\ldots,m$, where each φ_i has a continuous qth derivative on I. We will also let $\alpha \in E^m$ denote a vector with elements α_i.

In this system splines of degree $r \geq q + 1$ may be chosen, where $r \in \{2,3,4,5\}$. For computational purposes a convenient basis for an arbitrary spline of specified degree and uniform knot size is given by a slight modification of B-splines [5]. Details of this modification (denoted as β-splines) are contained in a previous publication [6]. Here we will just give a simple example. Suppose we wish to approximate a 2nd order differential equation on $[a,b]$ using six β-splines of degree three. Then $m = 6$, $r = 3$, and $v(\alpha,x) = \sum_{i=1}^{6} \alpha_i \varphi_i(x)$ where the $\varphi_i(x)$, $i = 1,\ldots,6$ are the β-splines. DIFEQ would then determine the uniform knot size Δ by

$$\Delta = \frac{b-a}{m-r} = \frac{b-a}{3} \tag{2.4}$$

and position the six β-splines as shown in Figure 1 where the sections of the functions lying outside of $[a,b]$ are indicated with dashed lines.

If r and m have been selected and Δ has been determined as in (2.4), then we can define the set of points

$$I_\Delta = \{x_j \mid x_j = a + j\Delta,\ j = 1,2,\ldots,m-r-1\},$$

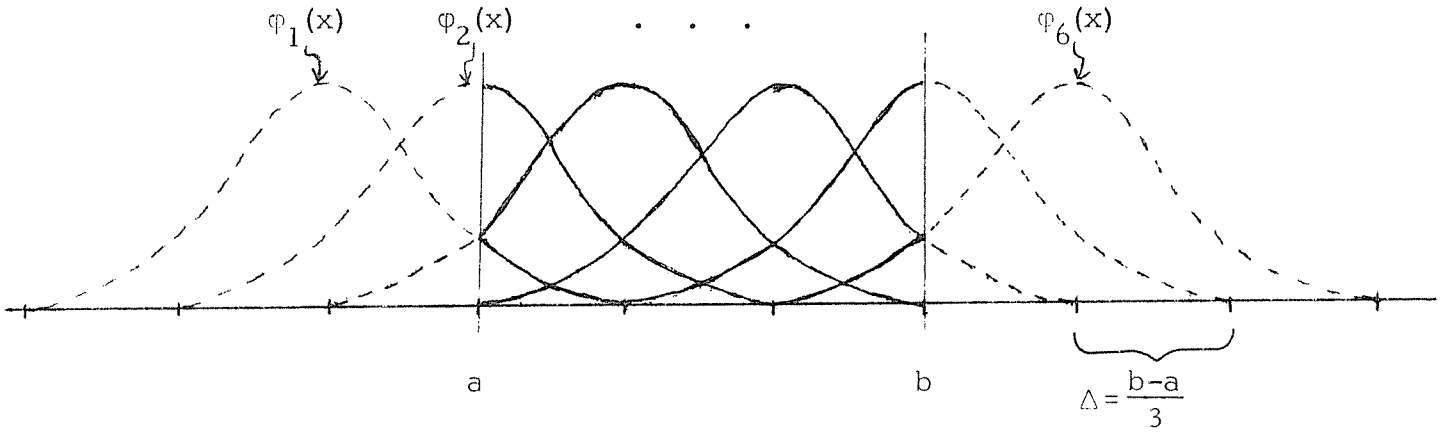

Fig. 1. Position of β-splines for m=6 and r=3.

consisting of the knots lying in the open interval (a,b). Let $T \in \{1,2,4,8,16\}$, and define

$$h = \frac{\Delta}{T} \quad \text{and} \quad I_h = \{x_j \mid x_j = a + jh,\ j = 1,2,\ldots,T(m-r)-1\},$$

so that $I_\Delta \subset I_h$.

We wish to determine a function $v(\alpha,x)$ for $x \in [a,b]$ which minimizes the maximum error over the discrete set of points I_h. That is, we want to find α so as to minimize

$$\begin{aligned} \psi(\alpha) &\equiv \|L[v(\alpha)] - f\|_{I_h} \\ &= \max_{x_j \in I_h} \left| v^{(q)}(\alpha,x_j) + \sum_{k=0}^{q-1} a_k(x_j) v^{(k)}(\alpha,x_j) - f(x_j) \right| \end{aligned} \tag{2.5}$$

If we let

$$\omega_i(x) \equiv L[\varphi_i(x)] \tag{2.6}$$

then (2.5) can be expressed as minimizing

$$\psi(\alpha) = \Big\| \sum_{i=1}^{m} \alpha_i \omega_i - f \Big\|_{I_h} \tag{2.7}$$

We also define

$$\sigma_{\ell i} = g_\ell[\varphi_i(x_\ell)], \quad \begin{matrix} i=1,\ldots,m \\ \ell=1,\ldots,q \end{matrix} . \tag{2.8}$$

Then our boundary conditions (2.2) become

$$\sum_{i=1}^{m} \alpha_i \sigma_{\ell i} = c_\ell \qquad \ell = 1,\ldots,q. \tag{2.9}$$

$$\text{Finally, let } A = \{\alpha_i \mid \sum_{i=1}^{m} \alpha_i \sigma_{\ell i} = c_\ell, \quad \ell = 1,\ldots,q\} \tag{2.10}$$

Now our minimization problem becomes:

Find a vector $\hat{\alpha} \in A$, such that

$$\psi(\hat{\alpha}) = \min_{\alpha \in A} \psi(\alpha). \tag{2.11}$$

For the details as to how a problem of this nature is expressed and solved using LP, see [6].

We now briefly mention the type of auxiliary conditions that the system DIFEQ can handle. We can place lower and/or upper bounds on the approximating functions $v(\alpha,x)$, $v'(\alpha,x)$, and $v''(\alpha,x)$ at specified points in $[a,b]$. In addition, we can place lower and/or upper bounds on the elements of the coefficient vector α. Additional points can also be added to the set I_h so our grid can become nonuniform. The manner in which these auxiliary conditions may be added to the LP problem is also discussed in [6].

After solving the LP problem and getting the coefficient vector $\hat{\alpha}$ and corresponding best approximation $v(\hat{\alpha},x)$, we would also like to see graphs of the error in the differential equation (also called "defect") given by

$$ED(x) = L[v(\hat{\alpha},x)] - f(x) \tag{2.12}$$

and of the solution error given by

$$E(x) = v(\hat{\alpha},x) - u(x), \tag{2.13}$$

if we happen to know the exact solution $u(x)$. When applicable, DIFEQ displays both of these graphs.

III. PROGRAM DESCRIPTION

The system DIFEQ is implemented on the Univac 1108 and Adage AGT/10 computers. The adage has a 16,000 30-bit word memory, two magnetic tape drives, a teletype, a graphical display with light pen, and various function switches and analogue inputs. As shown in Figure 2, DIFEQ actually consists of two programs, ADIFEQ and UDIFEQ which run on the adage and Univac respectively. These two programs communicate via a high speed data channel connecting the two computers. Almost all of the routines in ADIFEQ are written in ADEPT, the adage machine language, whereas UDIFEQ is written exclusively in FORTRAN.

To initiate the program UDIFEQ, a small deck of about five cards must be submitted at the 9300 input system which is adjacent to the Adage. This deck instructs the Univac to fetch from drum the program UDIFEQ and execute it. Usually UDIFEQ is active within five minutes. Assuming ADIFEQ has been initiated at the Adage, the display shown in Figure 3 would appear. In this frame we are requested to select the order q of the differential equation to be approximated, by tagging the appropriate line with the light pen.

Say we wish to approximate, as described in Section II, the function

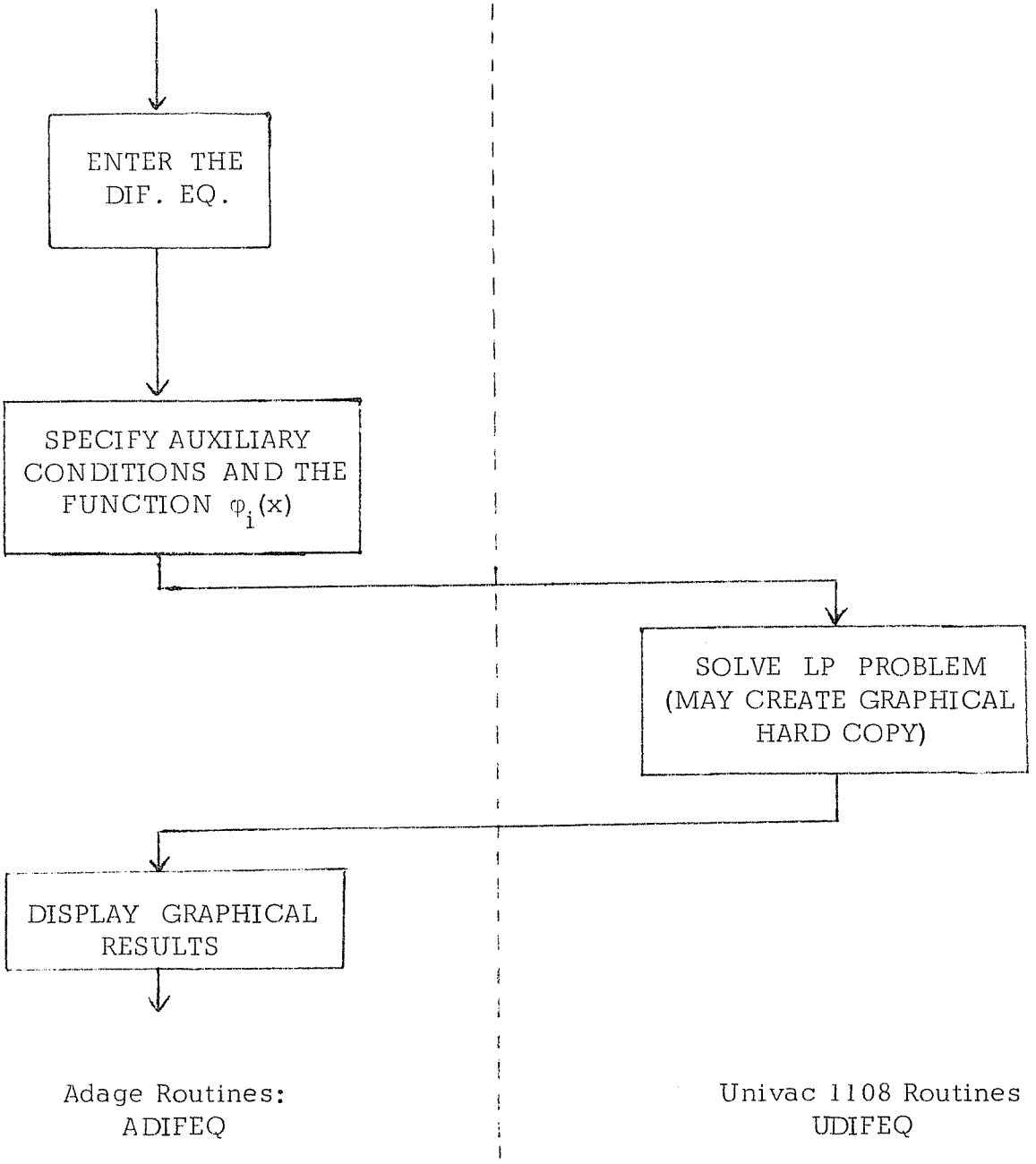

Fig. 2. Overview of DIFEQ

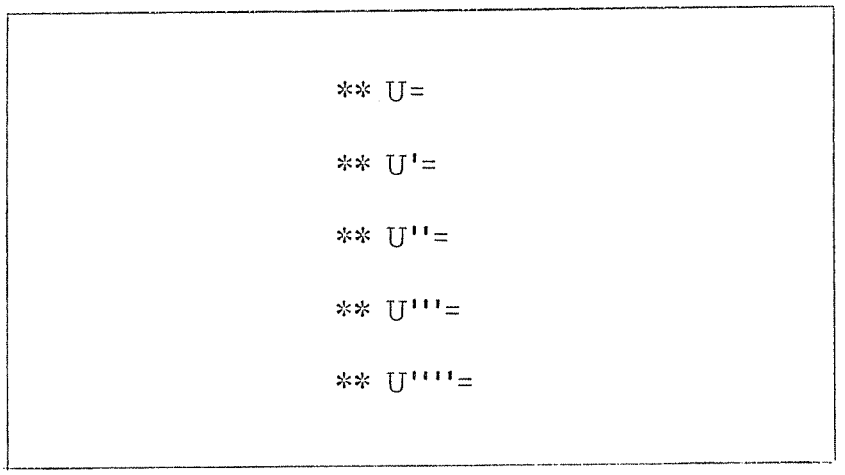

Fig. 3

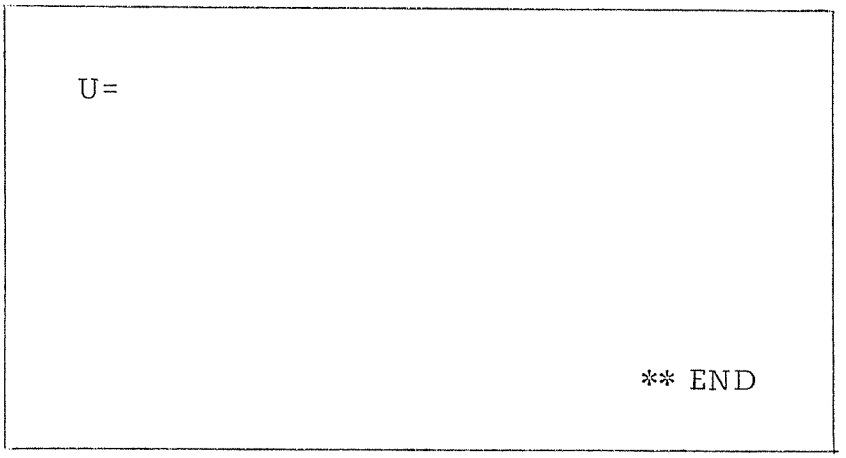

Fig. 4

$$u = 1 + 9 \sin (3x) + 2 \sin (9x) \qquad (3.1)$$

over the interval $[0,1]$. We wish to find the best approximation on a specified discrete grid I_h, with the additional conditions that our approximation be exact at the two end points; that is $u(0) = 1$ and $u(1) = 3.095$. After tagging the first line in Figure 3, the display in Figure 4 appears.

We have selected $q = 0$ and are now requested to enter the equation. This is done by using the light pen in conjunction with the sixteen function switches shown in Figure 5. The function switches programmatically represent the various syntactic units indicated below the switches. When a switch is pressed the appropriate symbol or symbols appear on the display to form the equation. There are not enough function switches to include one for each of the variables and functions in the set {x,u,u',u",u"'}, and rather than redefine the switches when one of the members of this set is to be selected, we decided to use the light pen in conjunction with switch 1 as illustrated in the following example. Assume we had indicated in Figure 3 that we wished to enter a second order differential equation. Then, whenever switch 1 was pressed, the following text would be added to the display

** X

** U

** U'

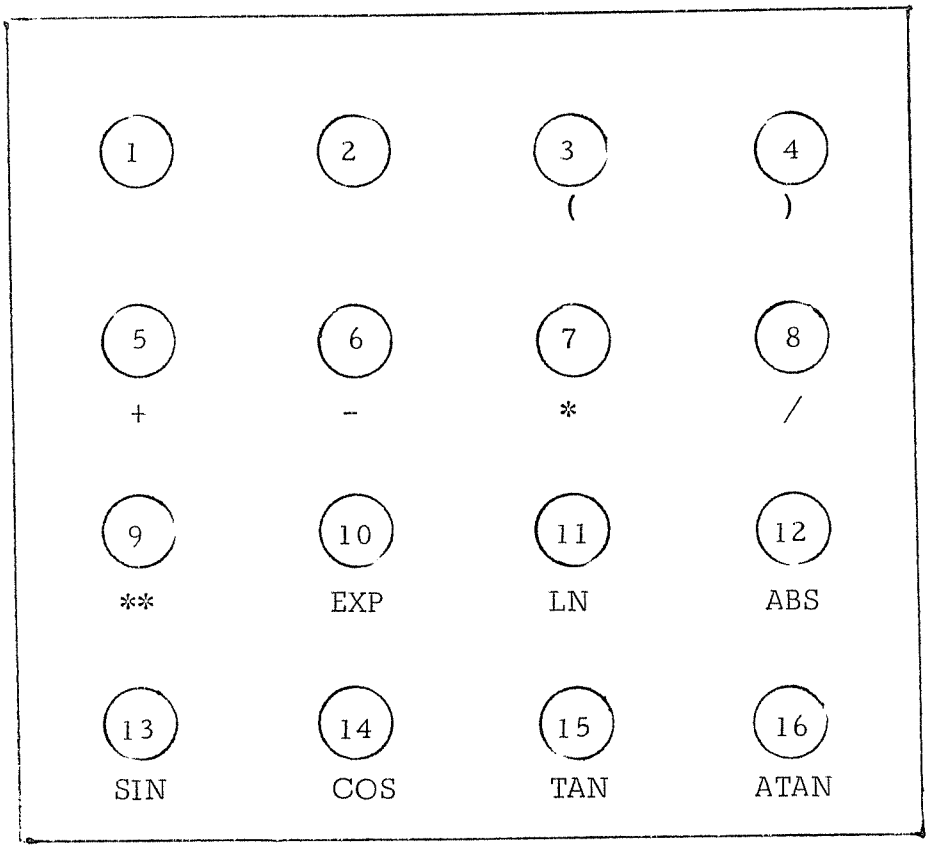

Fig. 5. Panel of sixteen function switches.

```
U= 1.ØØØØØ + 9.ØØØØØ*SIN( 3.ØØØØØ*X) + 2.ØØØØØ*SIN( 9.ØØØØØ*X)

                                                        ** END
```

Fig. 6

and the desired line could be selected with the light pen. Notice that the functions u" and u"' are not offered since they are not legitimate entries for the remaining terms of a second order differential equation. In our current problem, pressing switch 1 automatically sets the variable x into the equation.

Function switch 2 allows us to enter a constant. When it is pressed an array of numbers like the one in Figure 8 is added to the display, and the desired constant can be entered with the light pen. By operating the function switches with one hand and the light pen with the other, the equation as shown in Figure 6 can be generated quite rapidly. If an error has been made, the equation can be erased from right to left by pushing the foot pedal which is essentially another function switch. Each time the foot pedal is operated one syntactic unit in the equation is erased.

When the equation has been entered and "** END" is tagged, a syntax analyzer routine in ADIFEQ parses the equation and checks for syntax errors. If there are any errors we are taken back to Figure 3, otherwise we move on to Figure 7 where we may specify auxiliary conditions. In particular, we can place lower and/or upper bounds on the coefficients α_i of the β-splines, or bounds on the functions u(x), u'(x), or u"(x) at any specified points (as indicated

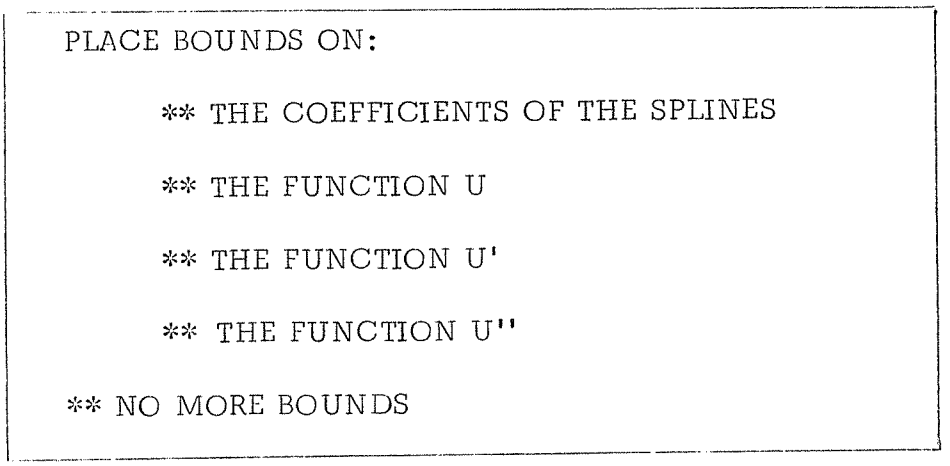
PLACE BOUNDS ON:
** THE COEFFICIENTS OF THE SPLINES
** THE FUNCTION U
** THE FUNCTION U'
** THE FUNCTION U''
** NO MORE BOUNDS

Fig. 7

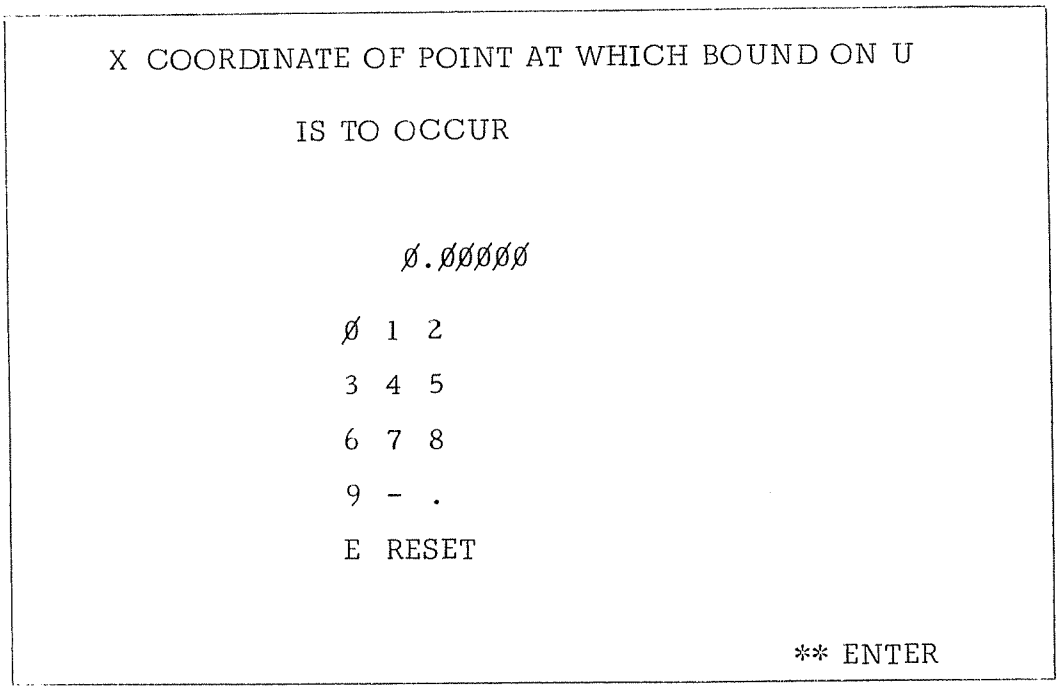
X COORDINATE OF POINT AT WHICH BOUND ON U
IS TO OCCUR
Ø.ØØØØØ
Ø 1 2
3 4 5
6 7 8
9 - .
E RESET
** ENTER

Fig. 8

in Section II, these bounds will actually be placed on the approximation $v(\alpha,x)$, $v'(\alpha,x)$, or $v''(\alpha,x)$.) Since we want to constrain $u(x)$ at the end points, we tag "** THE FCN U", and Figure 8 is shown allowing us to enter the point $x = 0$. Figure 9, now permits us to specify the lower bound, if any, for $u(0)$. After entering the lower bound $u(0) \geq 1$, the next display allows us to specify an upper bound in the same way. This sequence is then repeated for the condition $u(1) = 3.095$.

In Figure 10 we are requested to specify the interval $[a,b]$ and m, the number of β-splines to be used. Since we wish to use seven splines to approximate the equation over $[0,1]$, these numbers are entered, after forming them above the number array, by tagging the appropriate line "A =", "B =", or "M =". We could go back (* B) to Figure 7 and make changes in the auxiliary conditions or continue on (* c) to Figure 11 where the degree r of the splines is entered. After selecting the degree, the spline knot size is determined, using the equation $\Delta = (b - a)/(m - r)$, and this is displayed in Figure 12. If a different knot size is desired, we could go back and change m.

Now we are ready to specify the grid size for the discrete grid, I_h, over which we will minimize the error. We may choose a grid that is 2, 4, 8, or 16 times as fine as the knot size, or we can choose

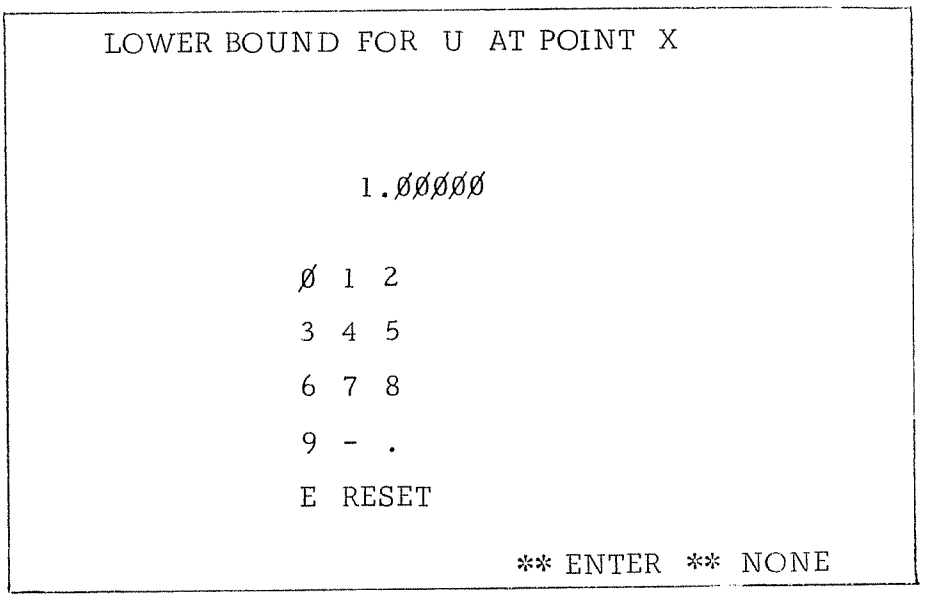
LOWER BOUND FOR U AT POINT X
1.ØØØØØ
Ø 1 2
3 4 5
6 7 8
9 - .
E RESET
** ENTER ** NONE

Fig. 9

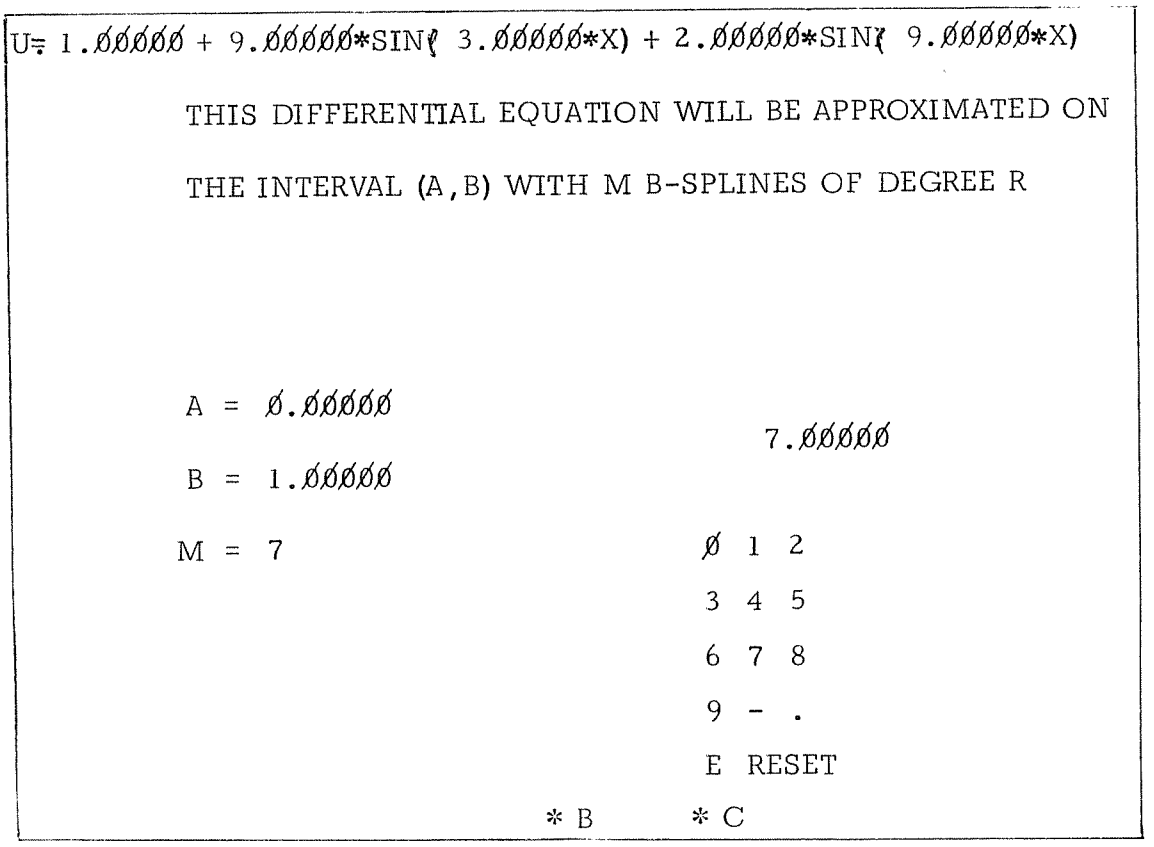
U= 1.ØØØØØ + 9.ØØØØØ*SIN(3.ØØØØØ*X) + 2.ØØØØØ*SIN(9.ØØØØØ*X)
THIS DIFFERENTIAL EQUATION WILL BE APPROXIMATED ON
THE INTERVAL (A,B) WITH M B-SPLINES OF DEGREE R
A = Ø.ØØØØØ
B = 1.ØØØØØ
M = 7
7.ØØØØØ
Ø 1 2
3 4 5
6 7 8
9 - .
E RESET
* B
* C

Fig. 10

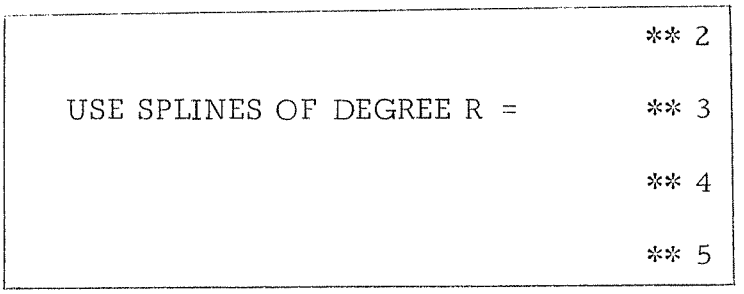

Fig. 11

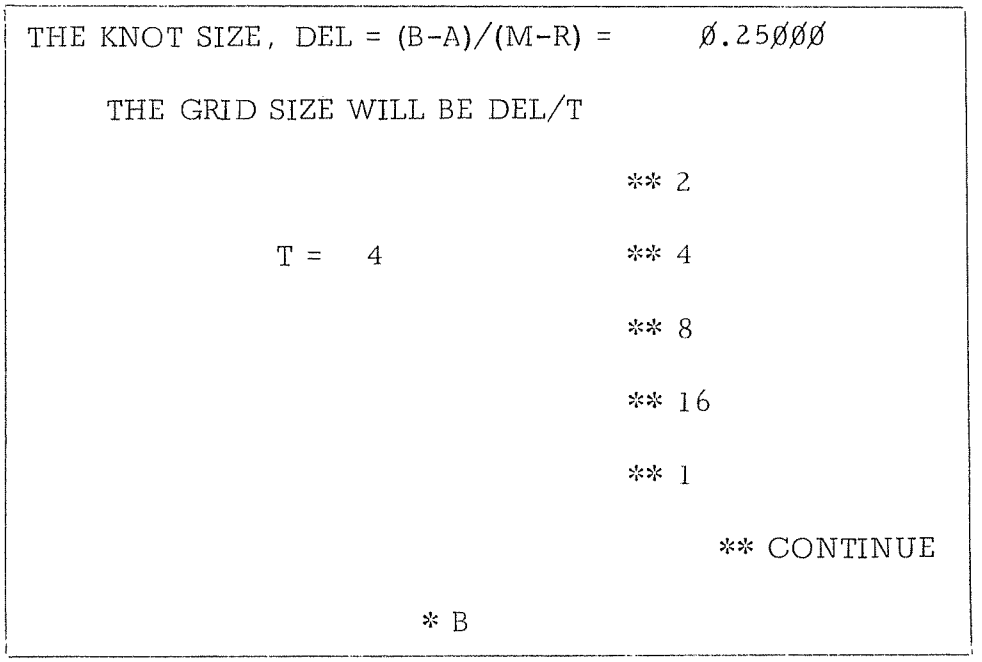

Fig. 12

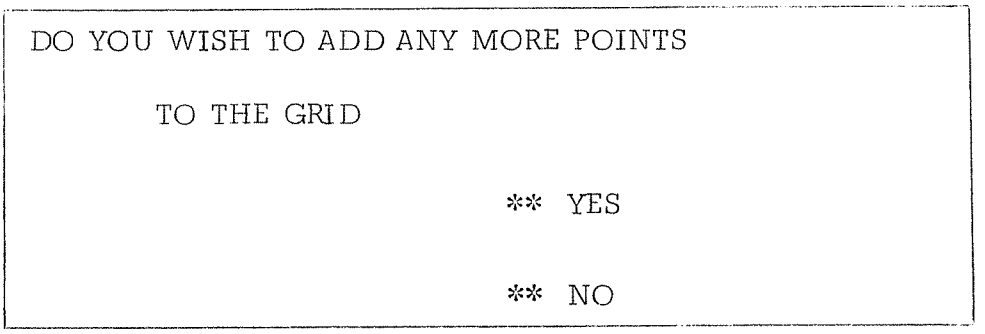

Fig. 13

$I_h = I_\Delta$. This last case allows us to satisfy the differential equation exactly at the knots in (a,b) and corresponds to the special case of interpolation at the knots. In Figure 12 we have selected $T = 4$ and therefore our grid will consist of the fifteen points $x_j = (.0625)j$ $j = 1,\ldots,15$. The end points a and b are not included in I_h, but may be added later, if desired.

In designing DIFEQ we have tried to make only the valid options available to the user. Thus in Figures 3, 7, 11, and 12 only legitimate choices are displayed. Whenever possible, the user's selections are immediately checked for consistency. For instance, in Figures 10 and 11 we would get no response if we tried to choose $b < a$ or $m \leq r$. Thus, to the extent possible, both syntax and consistency of input are automatically checked by the program.

After choosing the grid size, Figure 13 is displayed. If we wish, we may now place additional points in our grid. Since we don't wish to add points at this time, we tag "** NO".

Now that the problem has been defined, it is printed on the Adage teletype as shown in Figure 14 so the user has a hard copy of the input and has a final chance to look for mistakes. If he wishes to make some changes he can go back to Figures 3, 7, or 10. Otherwise, this data is sent to the 1108 where UDIFEQ generates the

```
U= 1.00000 + 9.00000*SIN( 3.00000*X) + 2.00000*SIN( 9.00000*X)

                7 B-SPLINES OF DEGREE 3 WILL BE USED
                 ON THE INTERVAL (  0.000,   1.000)

                 KNOT SIZE =  0.2500      T =  4

  FUNCTION       POINT        LOWER BND       UPPER BND
     U          0.00000         1.00000         1.00000
     U          1.00000         3.09500         3.09500
```

Fig. 14. Statement of problem (3.1) with no auxiliary conditions.

```
THE MAX GRID ERROR IS    0.77192E-01
THE MAX ERROR IS ABOUT   0.89143E-01

THE COEFFICIENTS ARE:          THE SPLINES ARE SUPPORTED ON:
      -8.09967                      ( -0.750,   0.250)
       1.18576                      ( -0.500,   0.500)
       7.35662                      ( -0.250,   0.750)
       4.38630                      (  0.000,   1.000)
       6.87899                      (  0.250,   1.250)
       3.64083                      (  0.500,   1.500)
      -9.06232                      (  0.750,   1.750)
```

Fig. 15. Teletype results from problem stated in Figure 14.

primal LP problem which is then solved by SIMDX, a double precision linear programming routine [11].

The coefficient vector $\hat{\alpha}$, obtained from SIMDX, is used by UDIFEQ to generate graphs of the approximation $v(\hat{\alpha},x)$ and the equation error $ED(x) = v(\hat{\alpha},x) - f(x)$, and this is then sent to ADIFEQ.

In this manner, the tedious job of generating and solving the LP problem is accomplished without any further intervention by the user. The waiting time between sending the data to the 1108 and the display of the graphical results on the Adage varies between one and sixty seconds. This time depends on the magnitude of the problem and the load on the 1108, with the majority of problems falling in the two to ten second range.

Figure 16 shows the results from our approximation. This display contains a graph of the approximation $v(\hat{\alpha},x)$ versus x and a graph of the equation error $ED(x) = v(\hat{\alpha},x) - f(x)$. These graphs are calculated using a fine grid of 101 points in $[a,b]$. While this is being displayed, the Adage teletype is printing out the results in Figure 15. It types out the coefficients $\hat{\alpha}_i$ of the splines, the interval over which they are non-zero, and the grid error of .077, which is the maximum equation error ED over the grid points I_h,

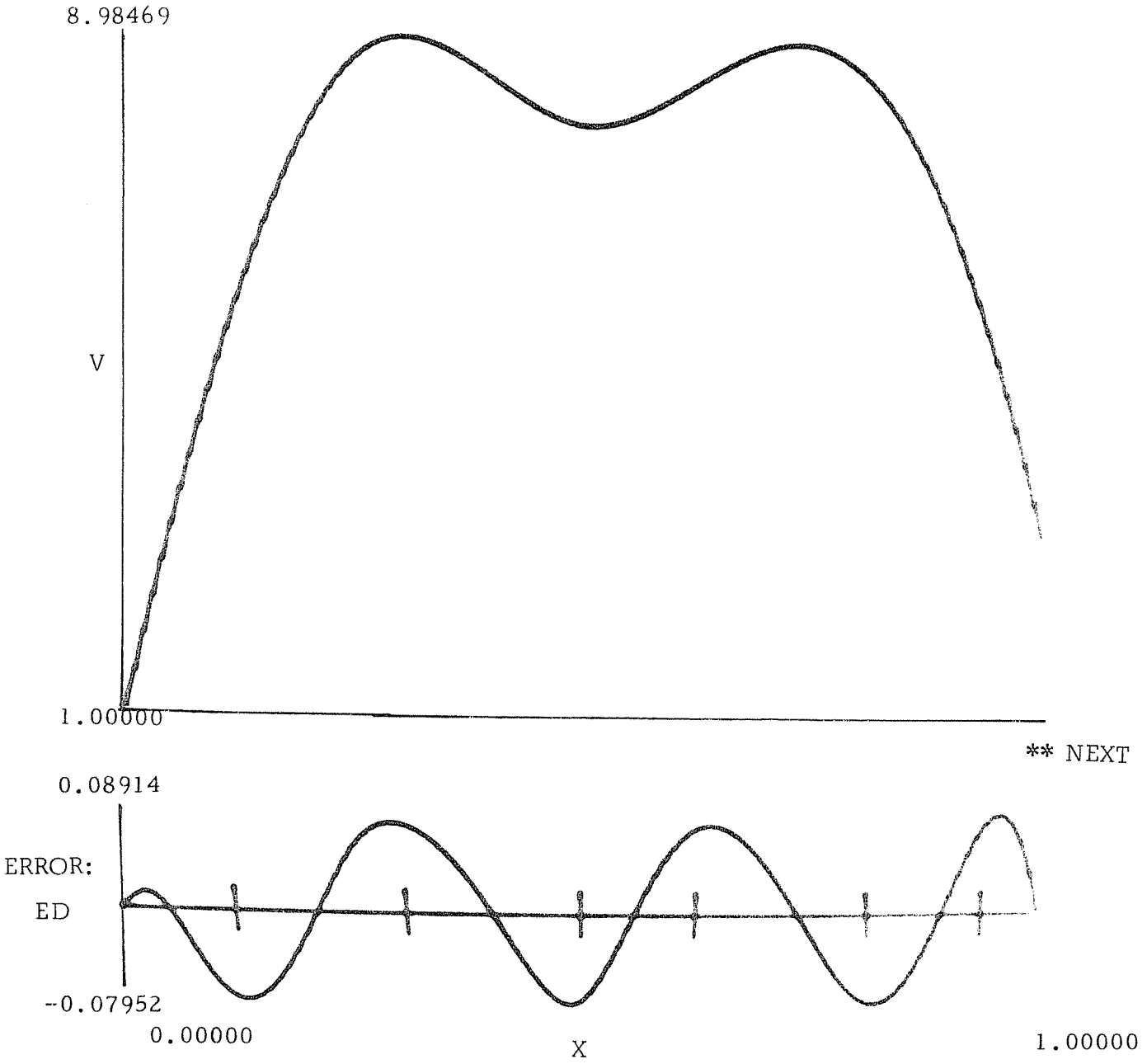

Fig. 16. Solution of problem stated in Figure 14.
Also see Figure 15 for teletype output.

and is attained at the points indicated by the vertical bars in Figure 16. The maximum error of .089 is the maximum equation error over the 101 points. Notice that, since we specified boundary conditions, the equation error is zero at $x = 0,1$.

We would now like to obtain an approximation which does not have the "camel-back" shape seen in Figure 16. We do this by adding the convexity constraints: $u''(x) \leq 0$ for $x \in \{.25, .375, .5, .625, .75\}$. It is not necessary for us to reenter the equation or boundary conditions, as these are retained by DIFEQ. The problem with the additional constraints is stated in Figure 17 along with the results from UDIFEQ. As seen in Figure 18, our approximation is now convex as desired. This convexity is obtained at the cost of an increase in the error to approximately 0.56.

To illustrate the approximate solution of a two-point boundary value problem, we now consider the following second order differential equation:

$$u'' + 9u = 9 - 144 \sin(9x)$$
$$u(0) = 1 \qquad u(1) = 3.095 \tag{3.2}$$

Here we are assuming that the right side of (3.2) is not known exactly. In practice this may arise for example, when the right side is given by a set of experimental data subject to error. In

```
U= 1.00000 + 9.00000*SIN( 3.00000*X) + 2.00000*SIN( 9.00000*X)

                7 B-SPLINES OF DEGREE 3 WILL BE USED
                 ON THE INTERVAL (  0.000,  1.000)

                KNOT SIZE =  0.2500     T =  4

   FUNCTION       POINT        LOWER BND        UPPER BND
       U          0.00000          1.00000          1.00000
       U          1.00000          3.09500          3.09500
       DDU        0.25000                           0.00000
       DDU        0.37500                           0.00000
       DDU        0.50000                           0.00000
       DDU        0.62500                           0.00000
       DDU        0.75000                           0.00000


 THE MAX GRID ERROR IS    0.54855E 00
 THE MAX ERROR IS ABOUT   0.55838E 00

THE COEFFICIENTS ARE:          THE SPLINES ARE SUPPORTED ON:
      -15.54486                     ( -0.750,  0.250)
        3.42160                     ( -0.500,  0.500)
        5.85846                     ( -0.250,  0.750)
        5.71396                     (  0.000,  1.000)
        5.56947                     (  0.250,  1.250)
        5.42497                     (  0.500,  1.500)
      -14.88935                     (  0.750,  1.750)
```

Fig. 17. Statement and teletype results of problem (3.1) with convexity conditions added.

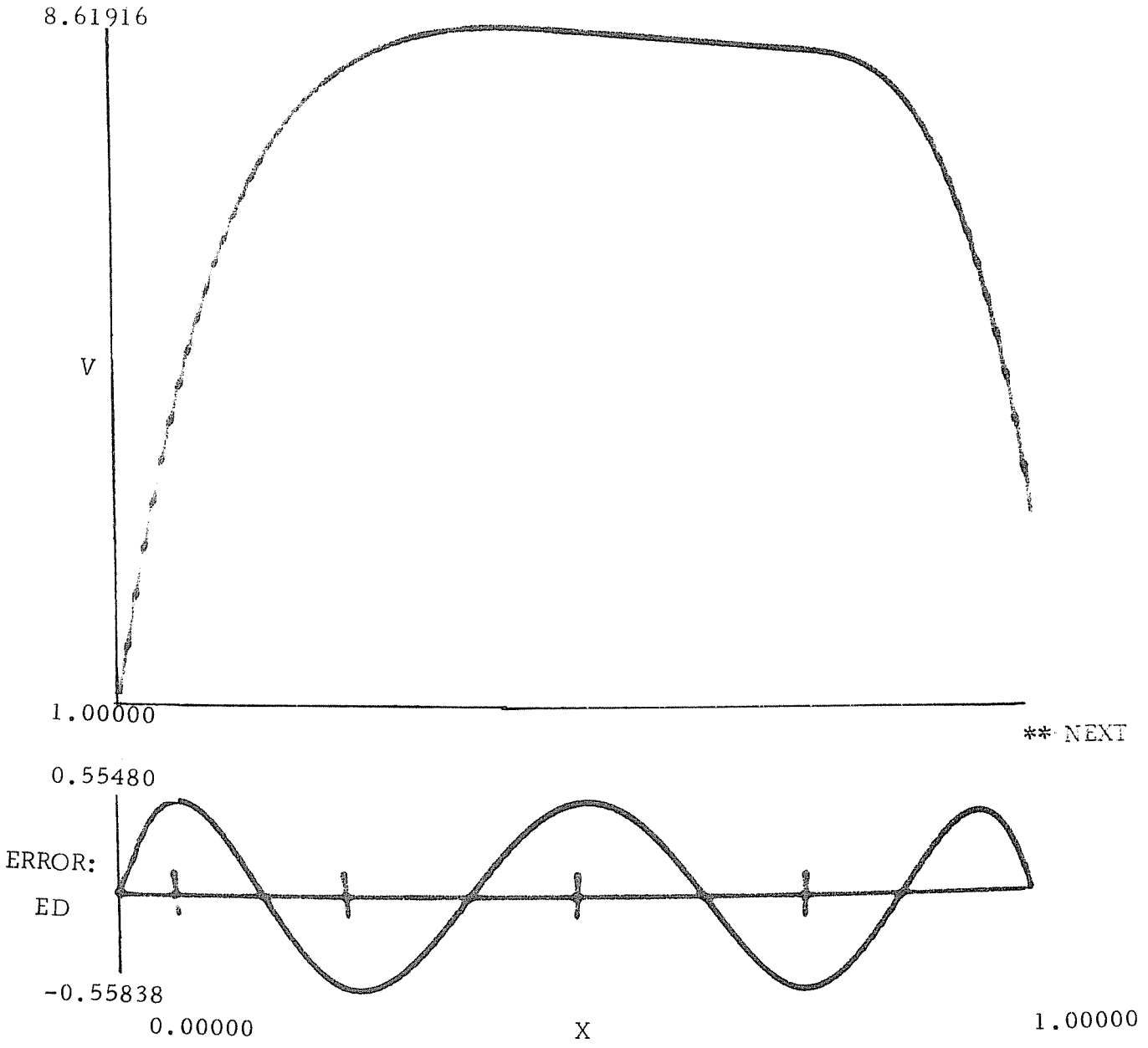

Fig. 18. Solution of problem stated in Figure 17.

addition to satisfying the boundary conditions, we will require that our approximate solution be convex and bounded above by 9.0 on the interval [0,1]. These auxiliary requirements are based on assumed prior knowledge of the behavior of the solution to the problem. The differential equation (3.2) is entered in the same manner used to enter the function (3.1). As shown in Figure 19, this problem will be approximated using 13 cubic β-splines. For the purpose of illustration we have not included the convexity requirement but have entered the constraints:

$$u(x) \leq 9 \quad \text{for} \quad x \in \{.4, .55, .7\} \tag{3.3}$$

The graphical results are given in Figure 21 where the approximate solution $v(\hat{\alpha},x)$ and the differential equation error $ED(x) = L[v(\hat{\alpha},x)] - f(x)$ are shown. The teletype output is shown in Figure 20. In this problem the constraints (3.3) are not active, that is, the solution would have remained the same even if the constraints had not been included.

In order to obtain a convex solution we shall now add the constraints:

$$u''(x) \leq 0 \quad \text{for} \quad x \in \{.25, .375, .5, .625, .75\}.$$

This new problem is shown in Figure 22 and the results are given in Figures 23 and 24. Here we see that the approximate solution has the desired characteristics, but the differential equation error is

```
U''=-9.00000*U+ 9.00000-144.00000*SIN( 9.00000*X)

                13 B-SPLINES OF DEGREE 3 WILL BE USED
                  ON THE INTERVAL (  0.000,   1.000)

                  KNOT SIZE =  0.1000      T =   4

 FUNCTION        POINT          LOWER BND        UPPER BND
    U           0.00000          1.00000          1.00000
    U           1.00000          3.09500          3.09500
    U           0.40000                           9.00000
    U           0.55000                           9.00000
    U           0.70000                           9.00000
```

Fig. 19. Statement of problem (3.2) with upper bound conditions added.

```
 THE MAX GRID ERROR IS    0.80474E 01
 THE MAX ERROR IS ABOUT   0.80834E 01

THE COEFFICIENTS ARE:          THE SPLINES ARE SUPPORTED ON:
      -1.63070                      ( -0.300,   0.100)
       0.65843                      ( -0.200,   0.200)
       2.99697                      ( -0.100,   0.300)
       4.33735                      ( -0.000,   0.400)
       4.37833                      (  0.100,   0.500)
       3.67536                      (  0.200,   0.600)
       3.17701                      (  0.300,   0.700)
       3.43349                      (  0.400,   0.800)
       4.22838                      (  0.500,   0.900)
       4.73655                      (  0.600,   1.000)
       4.08032                      (  0.700,   1.100)
       2.15122                      (  0.800,   1.200)
      -0.30520                      (  0.900,   1.300)
```

Fig. 20. Teletype results from problem stated in Figure 19.

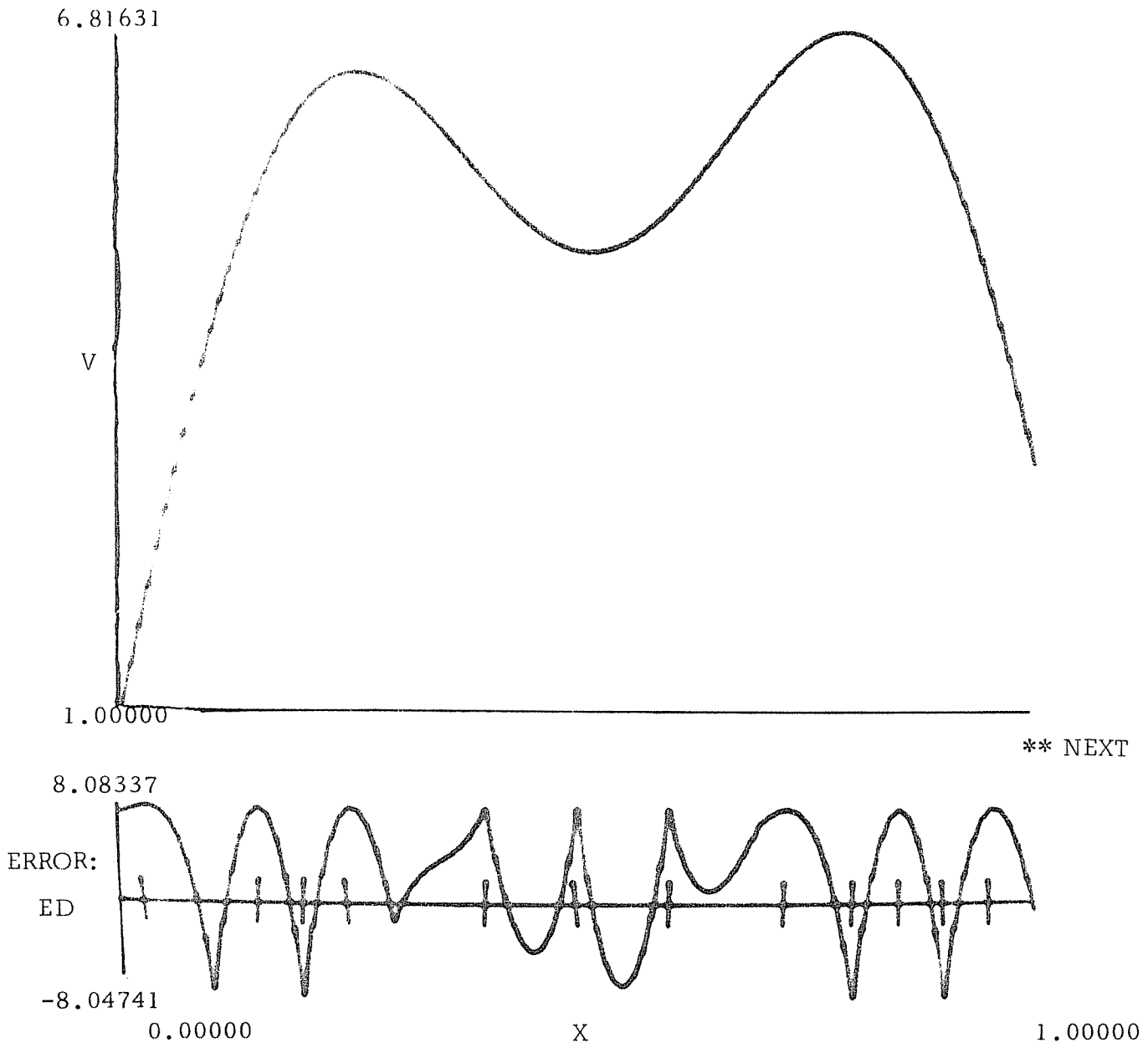

Fig. 21. Solution of problem stated in Figure 19.
Also see Figure 20 for teletype output.

```
U''=-9.00000*U+ 9.00000-144.00000*SIN( 9.00000*X)

              13 B-SPLINES OF DEGREE  3 WILL BE USED
                ON THE INTERVAL (  0.000,   1.000)

                KNOT SIZE =  0.1000     T =   4

 FUNCTION      POINT        LOWER BND        UPPER BND
     U         0.00000        1.00000          1.00000
     U         1.00000        3.09500          3.09500
     U         0.40000                         9.00000
     U         0.55000                         9.00000
     U         0.70000                         9.00000
     DDU       0.25000                         0.00000
     DDU       0.37500                         0.00000
     DDU       0.50000                         0.00000
     DDU       0.62500                         0.00000
     DDU       0.75000                         0.00000
```

Fig. 22. Statement of problem (3.2) with upper bound and convexity conditions added.

```
 THE MAX GRID ERROR IS    0.68573E 02
 THE MAX ERROR IS ABOUT   0.69490E 02

 THE COEFFICIENTS ARE:          THE SPLINES ARE SUPPORTED ON:
       -1.52409                      ( -0.300,   0.100)
        0.61534                      ( -0.200,   0.200)
        3.06273                      ( -0.100,   0.300)
        5.00717                      ( -0.000,   0.400)
        5.80057                      (  0.100,   0.500)
        6.04631                      (  0.200,   0.600)
        6.01420                      (  0.300,   0.700)
        5.98210                      (  0.400,   0.800)
        6.03877                      (  0.500,   0.900)
        5.55391                      (  0.600,   1.000)
        4.28751                      (  0.700,   1.100)
        2.07749                      (  0.800,   1.200)
       -0.21746                      (  0.900,   1.300)
```

Fig. 23. Teletype results from problem stated in Figure 22.

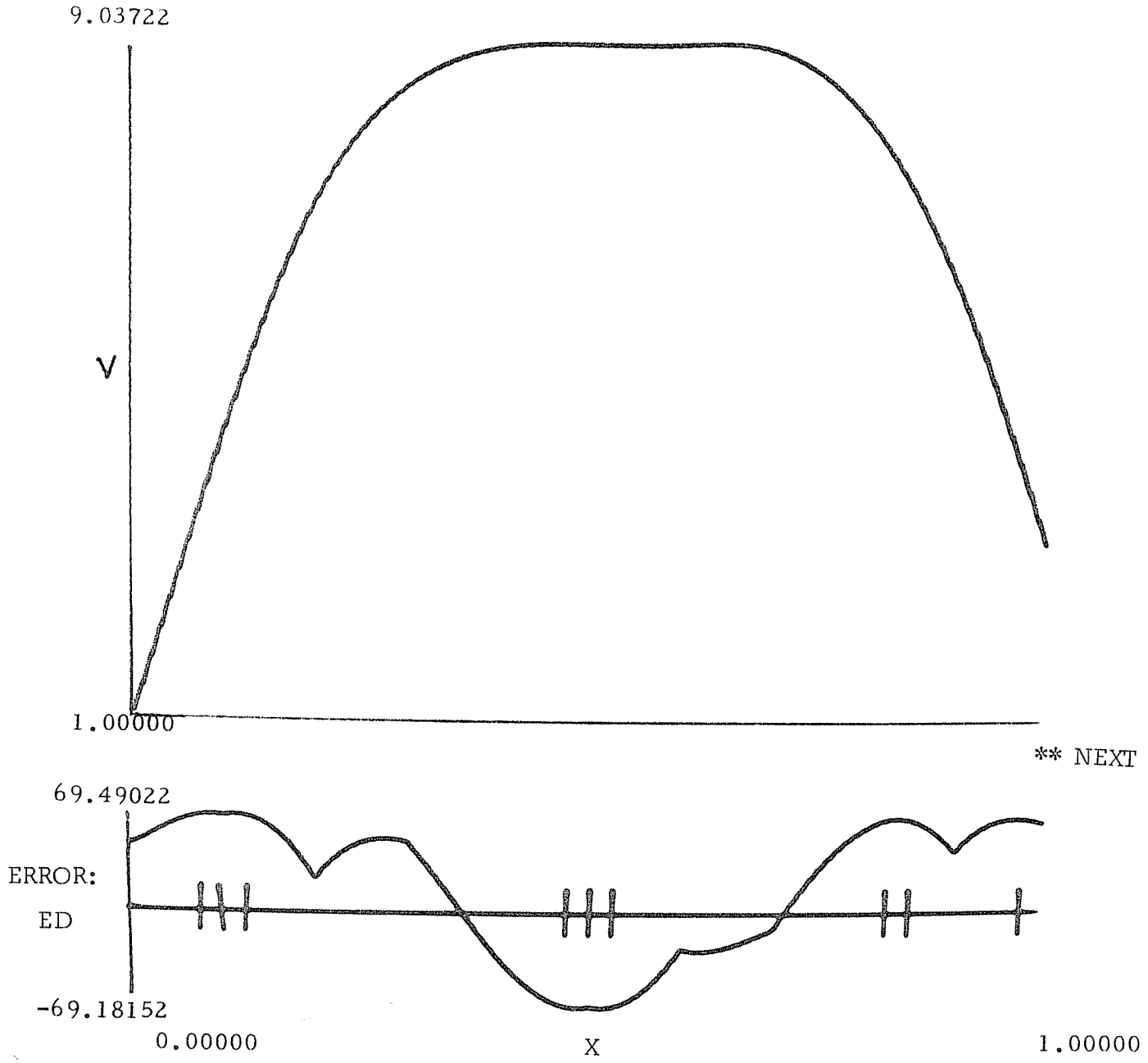

Fig. 24. Solution of problem stated in Figure 22.
Also see Figure 23 for teletype output.

of course significantly larger. The constraints (3.3) are active, for $v(.4) = v(.55) = 9$. Notice however that the upper bound of 9.0 over $[0,1]$ is not strictly enforced since $v(.45) = 9.037$. If desired, this could be avoided simply by including more points in the constraints (3.3).

If the right side of (3.2) is exact, then this differential equation has as its exact solution the function (3.1). For equations of this type (when the differential equation is exact and we happen to know the exact solution) we may wish to see a comparison between the exact and approximate solutions. If this is the case the exact solution must be entered, and a graph such as the one in Figure 25 will be shown. The top graph in this figure gives the solution error $E(x) = v(\hat{\alpha},x) - u(x)$ where the approximate solution $v(\hat{\alpha},x)$ is the one shown in Figure 24, and the bottom graph is the differential equation error which is also shown in Figure 24.

In this section we have presented four examples and would now like to indicate the amount of human and computer effort involved in defining and approximating these problems. A rough estimate of this is the amount of time required on the computer. At the Adage a total of 19 minutes was required to generate and approximate the four examples (this is from log-in to log-out), and during this time

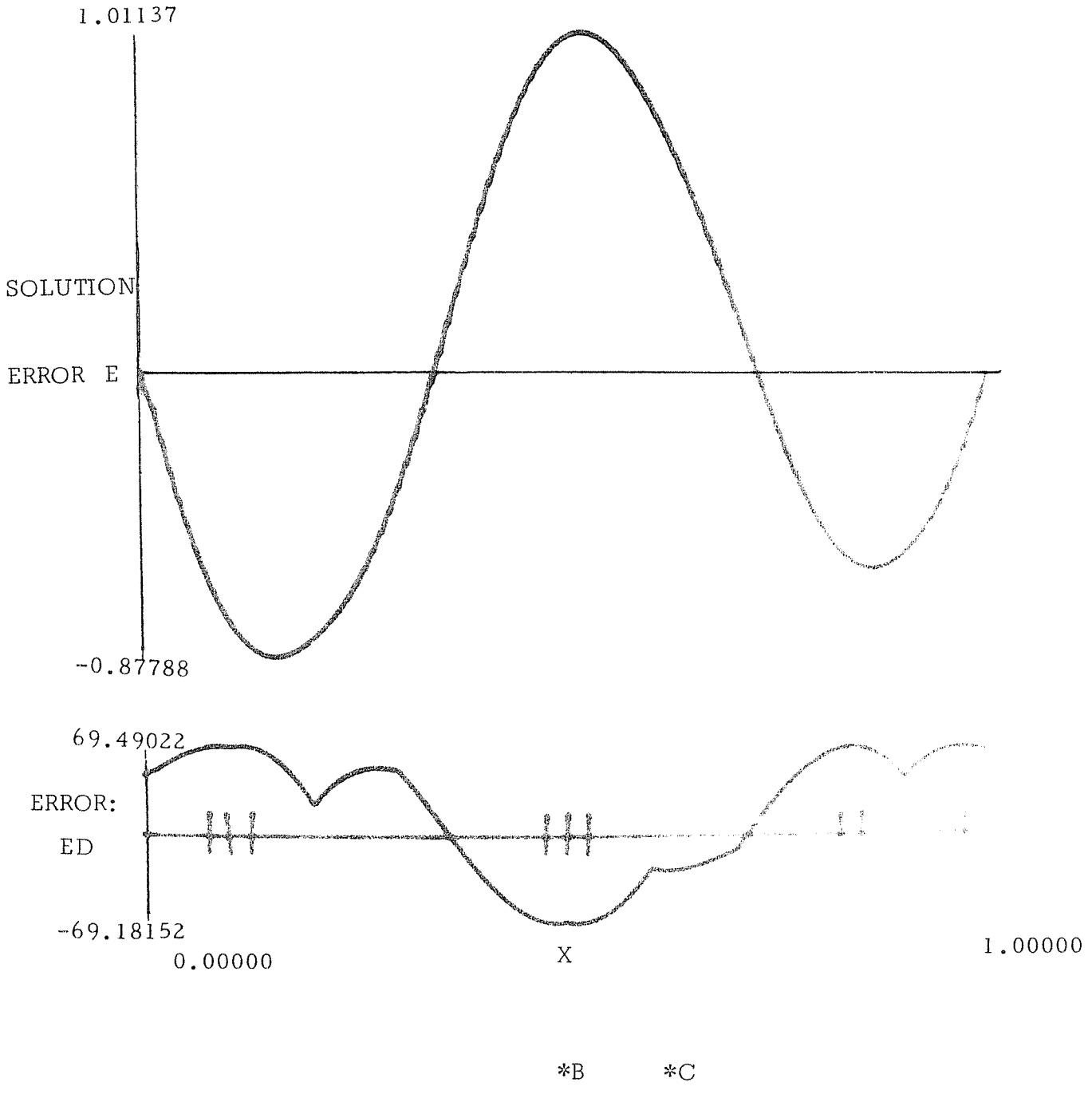

Fig. 25. Graph of solution error for problem indicated in Figures 22, 23, and 24.

the 1108 used only 19 seconds of CPU time. In an effort to make DIFEQ an easy system to use, it has been designed so that information does not have to be repeated during successive iterations of a problem. When it is desired to change some auxiliary conditions or parameters and recalculate the approximation to an equation, it is only necessary to specify the parameters to be changed, as DIFEQ retains the original values. In Figure 26 the flow of control or sequence of displays is illustrated, with the numbers in the boxes refering to the figures in this section. The normal flow of control, which is indicated with double lines, can be modified by the user, as indicated, to avoid reentering any parameters which are not to be changed. Thus for example, when we approximated the differential equation (3.2) the second time with additional side conditions, it was not necessary to redefine the equation or any of the parameters since they remained the same; only the auxiliary conditions had to be defined. In Figure 26, the line from box 7 to box 13 is dashed to indicate that even though the intervening displays (10, 11, and 12) will appear, it is not necessary to redefine the parameters at these displays and one can very rapidly pass through them to box 13.

In this manner we can approximate functions and solutions to differential equations, and can rapidly explore the effects which

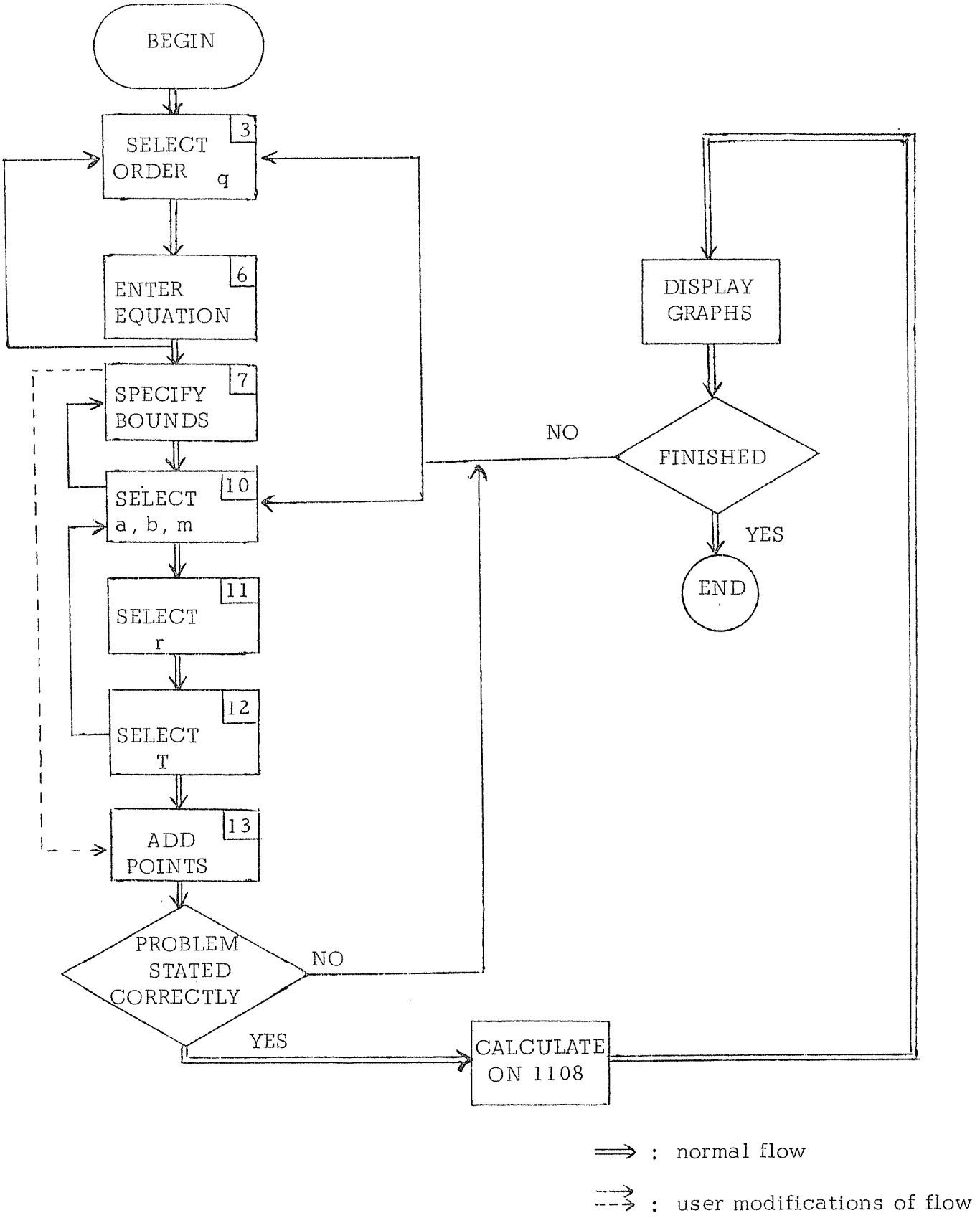

Fig. 26: Flow of control within DIFEQ

auxiliary conditions and chosen parameter values have on these approximations.

REFERENCES

1. Ahlberg, J. H., Spline approximation and computer-aided design. In Advances in Computers Volume 10, F. L. Alt and M. Rubinoff (Eds.), Academic Press, New York, 1970, pp. 275-289.

2. Barradale, I., and Young, A. Computational experience in solving linear operator equations using the Chebyshev norm. In Numerical Approximation to Functions and Data, J. G. Hays (Ed.), Athlone Press, London, 1970, pp. 115-142.

3. Culler, G. J., and Fried, B. D. The TRW two-station on-line scientific computer. In Computer Augmentation of Human Reasoning, M. A. Sass and W. D. Wilkenson (Eds.), Spartan Books, Inc., Washington, D. C., 1963, pp. 67-87.

4. Gallaher, L. J., and Perlin, I. E. A learning program for the integration of systems of ordinary differential equations. In Interactive Systems for Experimental Applied Mathematics, M. Klerer and J Reinfelds (Eds.), Academic Press, New York, 1968, pp. 335-340.

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

6. LaFata, P., and Rosen, J. B. An interactive display for approximation by linear programming. Comm. ACM 13, 11 (Nov. 1970) 651-659.

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

8. Rosen, J. B. Approximate solution and error bounds for quasilinear elliptic boundary value problem. SIAM J. Numer. Anal. 7 (1970), 80-103.

9. Rosen, J. B. Minimum error bounds for multidimentional spline approximation. Tech. Rept. No. 100, Computer Sciences Dept., U. of Wis., Madison, Oct. 1970.

10. Smith, L. B. A survey of interactive graphical systems for mathematics. Computing Surveys 2, 4 (Dec. 1970) 261-301.

11. U. of Wis. Computing Center. SIMPDX/SIMPLX linear programming subroutines. Madison, 1970.