Int. J. Appl. Engng Ed. Vol. 7, No. 1, pp. 68–70, 1991
Printed in Great Britain.

0742–0269/91 $3.00+0.00

Computer Implementation of the State Variables Method*

E. MAHAJERIN†

Mechanical Engineering Department, Saginaw Valley State University, University Center, Michigan 48710, *U.S.A.*

The notion of converting an nth order system to n first-order systems by means of 'state variables' has been introduced in many control textbooks. However, the process requires a workable computer program which is not found in these textbooks. In this paper a simple and rather compact computer program is developed which is applicable to systems of any order. The idea is based on introducing arbitrary variables. These variables are combined in a suitable fashion and are eliminated by differentiation and substitution. The method provides practical advice on the analysis of control systems as well as many areas of engineering and applied mathematics. The work is supplemented by a listing of the program in BASIC along with an example to illustrate its application.

INTRODUCTION

AN nth order system can be represented by

$$a_n D_n y + a_{n-1} D_{n-1} y + a_{n-2} D_{n-2} y + \ldots + a_1 D_1 y + a_0 y = g(x) \qquad (1)$$

where D_K, $k = 1,2,..,n$ represents the k-th derivative of y with respect to x. The initial conditions consist of known values of y, D_1y, D_2y, ..., and $D_{n-1}y$ at $x = x_0$.

For dynamic systems (including control systems), the response y is a function of time (x) and $g(x)$ respresents the forcing function which is also known. The ease of obtaining analytical solutions evaporates quickly as n and/or the complexity of the model increases. Thus, in practice, computer-oriented methods are used. The state variable method [1,2] transfers an n-th order differential equation in one unknown into n first order equations in n unknowns. Before illustrating this technique we need to identify the first order model.

FIRST ORDER MODEL

Numerous methods for solving a first-order system have been unified [3]. Among these methods the Euler method enjoys simplicity and is therefore used for clarity purposes.

Consider the first-order system

$$a_1 \mathrm{d}y/\mathrm{d}x + a_0 y = g(x) \qquad x_0 \leqslant x \leqslant x_f \qquad (2)$$
$$y(x=x_0) = B_0$$

where the coefficients a_1 and a_0 are not necessarily constants.

The basis of Euler's method is to rewrite equation (2) as

$$y' = \mathrm{d}y/\mathrm{d}x = f(x,y) \qquad (3)$$

where

$$f(x,y) = [g(x) - a_0 y]/a_1.$$

By means of Taylor's Series, the expression for y at a point near x will be given approximately by

$$y(x+h) = y(x) + h\,y'(x). \qquad (4)$$

From Equation (2) this becomes

$$y(x+h) = y(x) + h\,f(x,y) \qquad (5)$$

where h is the x-increment. In computer equation (5) becomes

$$y_{k+1} = y_k + h\,f(x_k,y_k), k = 1,2,\ldots \qquad (6)$$
$$y_0 = B_0$$

Evaluation of (3) at the initial (x_0,B_0) permits one to compute y at an increment away from the initial point using (6). This process can be continued for as many steps as desired. Although this method has rather limited accuracy, it does, however, provide considerable insight into the understanding of the method in this paper as well as many other methods.

n-th ORDER MODEL—STATE VARIABLE METHOD

To obtain the nth-order version of equation (2), rewrite (1) as:

$$D_n y = f(x,y,D_1y,D_2y,D_3y,\ldots,D_{n-2}y, D_{n-1}y) \qquad (7)$$

* Paper accepted 1 June 1990.
† Enayat Mahajerin is Professor.

where

$$f(\ldots) = [g(x) - a_{n-1}D_{n-1}y - \ldots - a_1 D_1 y - a_0 y]/a_n,\ a_n \neq 0. \tag{8}$$

The initial conditions are expressed as

$$\begin{aligned} y(x_0) &= B_0 \\ D_1 y(x_0) &= B_1 \\ D_2 y(x_0) &= B_2 \\ D_3 y(x_0) &= B_3 \\ &\ \, . \\ &\ \, . \\ &\ \, . \\ D_{n-2} y(x_0) &= B_{n-2} \\ D_{n-1} y(x_0) &= B_{n-1}. \end{aligned} \tag{9}$$

To convert equation (1) into a system of first-order equations define the so-called 'state variables' y_1, $y_2, \ldots, y_n$

$$\begin{aligned} y_1 &= D_{n-1} y \\ y_2 &= D_{n-2} y \\ y_3 &= D_{n-3} y \\ &\ \, . \\ &\ \, . \\ &\ \, . \\ y_{n-1} &= D_1 y \\ y_n &= y \end{aligned} \tag{10}$$

Now, differentiate (10) with respect to x

$$\begin{aligned} D_1 y_1 &= D_n y \\ D_1 y_2 &= D_{n-1} y \\ D_1 y_3 &= D_{n-2} y \\ &\ \, . \\ &\ \, . \\ &\ \, . \\ D_1 y_{n-1} &= D_2 y \\ D_1 y_n &= D_1 y \end{aligned} \tag{11}$$

Finally, substituting (10) and (7) into (11) gives

$$\begin{aligned} D_1 y_1 &= f(x, y_n, y_{n-1}, y_{n-2}, \ldots, y_2, y_1) \\ D_1 y_2 &= y_1 \\ D_1 y_3 &= y_2 \\ &\ \, . \\ &\ \, . \\ &\ \, . \\ D_1 y_{n-1} &= y_{n-2} \\ D_1 y_n &= y_{n-1} \end{aligned} \tag{12}$$

which is a system of n first-order equations. The initial conditions (see equations 9 and 10) are:

$$\begin{aligned} y_1(x_0) &= B_{n-1} \\ y_2(x_0) &= B_{n-2} \\ y_3(x_0) &= B_{n-3} \\ &\ \, . \\ &\ \, . \\ &\ \, . \\ y_{n-1}(x_0) &= B_1 \\ y_n(x_0) &= B_0. \end{aligned} \tag{13}$$

We can solve this system by means of the method used for the first order model. Once the system is solved for state variables $y_1, y_2, \ldots, y_n$, we can find y and its derivatives according to (10).

THE COMPUTER PROGRAM

Table 1 shows a compact computer program for solving system (12) subjected to initial conditions (13). The INPUT values are initial position, x_0; order of the system, n; increment, h and the number of increments, k_{max} (see LINE 140). The user is required to type his/her function f (see Equation 7) in terms of state variables in line 120. All other state variables are automatically generated in the program (see function f_1 in LINES 130 and 200). Number of equations (assumed 10 in the program) is arbitrary and can be increased by the user. Arrays Y(10) and G(10) have been used to manipulate the state variables. (See the Loop in LINE 180.) The extension of the first order to handle a system of n first order equations is clearly shown in the Loop.

EXAMPLE

Consider the following 4-th order nonlinear system.

$$y^{(4)} + y^{(3)} + y^{(2)} + ye^{x}(y^{(1)} + y) = 1 + 2e^{-x}(x-2)$$
$$0 \leqslant x \leqslant 1$$

Table 1. The BASIC computer program for state variables method

```
100 'PROGRAM STATE-VARIABLES; E. M. 1/1/1990
110 DIM G(10),Y(10)
120 DEF FNF(X,Y1,Y2,Y3,Y4)=1+2*EXP(−X)*(X−2)−Y1−Y2−Y4*EXP(X)*(Y3+Y4)
130 DEF FNF1(Z)=Z
140 INPUT"Input X0,n,h,Kmax";X,N,H,KM
150 input"Input initials B0,B1,B2,B3";Y(4),Y(3),Y(2),Y(1)
160 FOR K=1 TO KM
170 PRINT X,Y(4),X*EXP(−X)
180 FOR J=1 TO N
190 G(1)=FNF(X,Y(1),Y(2),Y(3),Y(4))
200 if J>1 then G(J)=FNF1(Y(J−1))
210 Y(J)=Y(J)+H*G(J)
220 NEXT J
230 X=X+H
240 NEXT K
250 END
```

subject to the initial conditions

$y(0) = 0, y^{(1)}(0) = 1, y^{(2)}(0) = -2, y^{(3)}(0) = 3.$

The exact solution can be shown to be xe^{-x}. Using the notation in the paper this becomes

$D_4y + D_3y + D_2y + ye^x(D_1y + y) = 1 + 2e^{-x}(x-2).$

Consequently,

$D_4y = 1 + 2e^{-x}(x-2) - D_3y - D_2y - y\,e^x(D_1y + y).$

In terms of state variables this becomes

$D_4y = 1 + 2e^{-x}(x\text{-}2) - y_1 - y_2 - y_4\,e^x(y_3 + y_4).$

The right-hand side of this equation constructs the function f (see equation 7 and also line 120 of the program). Table 2 shows the computer results for two different increments. These results are directly compared to the exact values.

Table 2. Results for the example problem

x	y (h=.05)	y (h=.025)	y (Exact)
0	0	0	0
.05	.04535	.04643	.04756
.1	.08638	.08838	.09048
.15	.12339	.12619	.12911
.2	.15666	.16014	.16375
.25	.18648	.19025	.19470
.3	.21308	.21760	.22225
.35	.23670	.24162	.24664
.4	.25575	.26283	.26813
.45	.27590	.28144	.28693
.5	.29187	.29765	.30327
.55	.30568	.31167	.31732
.6	.31750	.32368	.32929
.65	.32750	.33386	.33933
.7	.33583	.34236	.34761
.75	.34266	.34936	.35427
.8	.34811	.35499	.35946
.85	.35233	.35941	.36330
.9	.35545	.36275	.36591
.95	.35760	.36513	.36740
1	.35889	.36669	.36788

CONCLUSIONS

Direct solution of higher-order differential equations can be very difficult and time consuming. Converting these equations to a system of first-order equations may be accomplished by introducing the state variables. However, this technique requires a workable computer program. These topics have been addressed in this paper. The proposed program is extremely compact and is easy to use. It offers the educator a powerful tool and can be integrated into a wide variety of areas such as dynamics and control.

REFERENCES

1. F. H. Raven, *Automatic Control Engineering*, 4th ed., McGraw-Hill (1987).
2. D. B. Miron, *Design of Feedback Control Systems*, Harcourt Brace Jovanovich (1989).
3. E. Mahajerin, A Unified computer program for the solution of ordinary differential equations, *Adv. Engng Software*, **9**(4), 222–224 (1987).