A PRINTER PLOTTER PROGRAM FOR DIGITAL SIMULATION STUDIES

MESSRS. K. L. PRIME AND C. S. BAUER
Department of Industrial Engineering and Management Systems
Florida Technological University
Orlando, Florida

Many computer simulation experiments involve the generation of large quantities of output data, resulting in extensive tables of numerical information. These tables are often difficult to interpret without considerable effort on the part of the reader, particularly with respect to the detection of variable trend perturbations in long strings of data. To alleviate this difficulty, a computer subroutine was developed to provide immediate printer plots of data arrays generated in simulation program runs. These plots allow the immediate examination of experimental run results, and provide the user with an easy-to-read tool for determining requirements for additional computer runs.

The program will plot up to five simultaneous data curves, with automatic plot variable scaling on each curve to achieve maximum output resolution in each instance. If the user wishes to plot any number of curves less than the five maximum allowed on a given set of axes, it is only necessary to fill the unused arrays appearing in the call statement with some common constant value, and the curve for this array or group of arrays will not be plotted. Similarly, simulation program outputs with more than five variables can easily be accommodated with multiple calls to the plotting program.

The plot routine was written in IBM 1130 FORTRAN, but should be acceptable to any FORTRAN compiler with an alphanumeric capability and provisions for a DATA statement. In fact, the authors use the same program deck on both IBM 1130 and IBM 360/65 computer runs, with the only change required being the appropriate selection for the FORTRAN logical unit number for the output printer. A complete listing of the program appears in figure 1.

The plotter is called from the data generation program with a statement of the following form:

```
CALL YPLOT(NPTS, A, B, C, D, E, X, IFLAG)
```

where:

NPTS = scaler integer variable giving the number of points to be plotted. The subroutine presently uses variable dimension array allocation to conserve memory requirements, and the reader should be aware that this feature may cause difficulties in other FORTRAN implementations.

A, B, C, D, E, = one dimensional FORTRAN arrays of length 'NPTS' containing the data points to be plotted.

X = one dimensional FORTRAN array of length 'NPTS' containing the values of the common independent variable values for each of the five data curves.

```
FORTRAN IV G LEVEL  19                YPLOT            DATE = 71177          14/34/24          PAGE 0001

            SUBROUTINE YPLOT  (M,AA,BB,CC,DD,EE,ZZ,KK)
            DIMENSION  X(100)
            DIMENSION  AA(2),BB(2),CC(2),DD(2),EE(2),ZZ(2)
            DATA  BL,A,B,C,D,E,DOT,DASH/' ','A','B','C','D','E','I','-'/
      C     FOLLOWING CARD SETS FORTRAN LOGICAL UNIT NUMBER FOR PRINTER...
            J = 6
            AMIN = AA(1)
            AMAX = AA(1)
            BMIN = BB(1)
            BMAX = BB(1)
            CMIN = CC(1)
            CMAX = CC(1)
            DMIN = DD(1)
            DMAX = DD(1)
            EMIN = EE(1)
            EMAX = EE(1)
            DO 555 N=1,100
        555 X(N) = BL
            DO 35 I=1,M
            IF ( AA(I) - AMAX) 10,11,11
         11 AMAX = AA(I)
         10 IF (AA(I) - AMIN) 12,12,13
         12 AMIN = AA(I)
         13 IF (BB(I) - BMAX) 20,21,21
         21 BMAX = BB(I)
         20 IF (BB(I) - BMIN) 22,22,23
         22 BMIN = BB(I)
         23 IF (CC(I) - CMAX) 24,25,25
         25 CMAX = CC(I)
         24 IF (CC(I) - CMIN) 26,26,27
         26 CMIN = CC(I)
         27 IF ( DD(I) - DMAX) 28,29,29
         29 DMAX = DD(I)
         28 IF (DD(I) - DMIN) 30,30,31
         30 DMIN = DD(I)
         31 IF (EE(I) - EMAX) 32,33,33
         33 EMAX = EE(I)
         32 IF (EE(I) - EMIN) 34,34,35
         34 EMIN = EE(I)
         35 CONTINUE
            AWID = AMAX - AMIN
            BWID = BMAX - BMIN
            CWID = CMAX - CMIN
            DWID = DMAX - DMIN
            EWID = EMAX - EMIN
            AUNIT = AWID / 100.
            BUNIT = BWID / 100.
            CUNIT = CWID / 100.
            DUNIT = DWID / 100.
            EUNIT = EWID / 100.
            AMID = AWID / 2. + AMIN
            BMID = BWID / 2. + BMIN
            CMID = CWID / 2. + CMIN
            DMID = DWID / 2. + DMIN
```

Figure 1. Plotter Program Listing (1 of 3)

```
FORTRAN IV G LEVEL  19                YPLOT            DATE = 71177          14/34/24          PAGE 0002

            EMID = EWIO / 2. + EMIN
            NK = 1
            DO 1000  L = 1,M
            Z = L
            IF ((L/10) - (Z/10.))  40,41,40
         41 DO 223  LL = 1,100
        223 X(LL) = DASH
         40 X(1)  =DOT
            X(25)  = DOT
            X(50) = DOT
            X(75) = DOT
            X(100)= DOT
            IF (AUNIT) 90,93,90
         90 KA = (AA(L) - AMIN) / AUNIT
            IF (KA)  91,91,92
         91 KA = 1
         92 X(KA) = A
         93 IF (BUNIT) 94,97,94
         94 KB = (BB(L) - BMIN) / BUNIT
            IF (KB) 95,95,96
         95 KB = 1
         96 X(KB) = B
         97 IF (CUNIT) 98,101,98
         98 KC = (CC(L) - CMIN) / CUNIT
            IF (KC)   99,99,100
         99 KC = 1
        100 X(KC) = C
        101 IF (DUNIT) 102,105,102
        102 KD = (DD(L) - DMIN) / DUNIT
            IF (KD) 103,103,104
        103 KD = 1
        104 X(KD) = D
        105 IF (EUNIT)106,109,106
        106 KE = (EE(L) - EMIN) / EUNIT
            IF (KE) 107,107,108
        107 KE = 1
        108 X(KE) = E
        109 IF (NK - 1)  151,152,151
        152 WRITE (J,111) AUNIT, BUNIT, CUNIT,DUNIT,EUNIT,
           1  AMIN, AMID, AMAX,
           2 BMIN, BMID, BMAX,
           3  CMIN, CMID, CMAX,
           4 DMIN, DMID, DMAX,
           5 EMIN, EMID, EMAX
        111 FORMAT('1'    ,'EACH UNIT ON THE Y -AXIS  =  ' ,
           1  F10.5,'  FOR EQ. A   ,   ',3X,F10.5,'  FOR EQ. B   ,   ',3X,
           1  F10.5,'  FOR EQ. C   ,   ',3X,/,T31,F10.5,'  FOR EQ. D   ,   ',3X,
           1  F10.5,'  FOR EQ. E ',
           25(/,F15.5,T45,F20.5,T95,F20.5),
           3/,T10,'+',T59,'+',T109,'+',/,T10,100('-'))
        151  IF  (KK)  155 , 156 , 155
        156 WRITE  (J,116) L,(X(MM),MM=1,100)
        116 FORMAT (1X, T6, I4, 100A1)
            GO TO  77
```

Figure 1. Plotter Program Listing (2 of 3)

```
FORTRAN IV G LEVEL  19                    YPLOT                DATE = 71177          14/34/24                PAGE 0003

         155 WRITE(J,117)  ZZ(L),(X(MM),MM=1,100)
         117 FORMAT (1X, F8.2,100A1)
          77 IF ((L/10) - (Z/10.))  45,46,45
          46 DO 47  KL = 1,100
          47 X(KL) = BL
          45 IF(NK - 50)  153,154,154
         154 NK = 0
         153 NK = NK + 1
             IF (AUNIT)    400,401,400
         400 X(KA) = BL
         401 IF (BUNIT)    402,403,402
         402 X(KB) = BL
         403 IF (CUNIT)    404,405,404
         404 X(KC) = BL
         405 IF (DUNIT)    406,407,406
         406 X(KD) = BL
         407 IF (EUNIT)    408,1000,408
         408 X(KE) = BL
        1000 CONTINUE
             RETURN
             END
```

Figure 1. Plotter Program Listing (3 of 3)

```
EACH UNIT ON THE Y -AXIS  =     0.85410  FOR EQ. A  ,       0.65424  FOR EQ. B  ,       0.20000  FOR EQ. C  ,
                                0.25631  FOR EQ. D  ,       0.0      FOR EQ. E
   314.59009                                        357.29492                                        400.00000
   200.00000                                        232.71179                                        265.42358
   100.00000                                        109.99994                                        119.99989
   100.00000                                        112.81557                                        125.63115
   120.00000                                        120.00000                                        120.00000
```

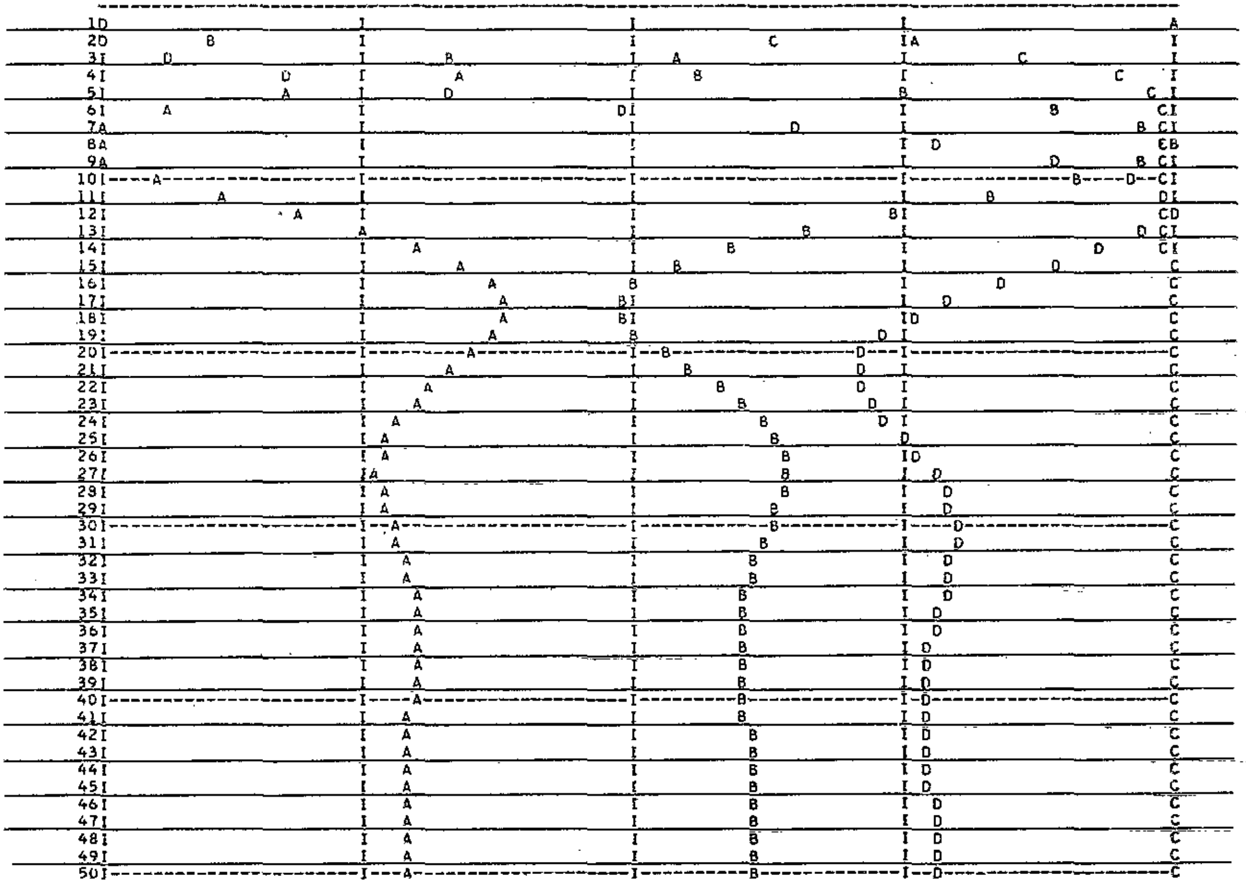

Figure 2. Plotter Output for Simulation Example

IFLAG = scaler integer control variable. If set equal to zero, the program will ignore the values in the X array and print an integer count of the data point numbers at the left of each row of the output plot. If set equal to one, the program will print the appropriate values from the X array at the left of each row of the output plot.

To illustrate the use of the program in a representative simulation study, consider the simple production-inventory system described in (1). The model equations are given in the reference in the DYNAMO Computer Language format, but may easily be converted to a state-variable differential equation following the procedure outlined in (2). Assuming a state vector definition of the form:

$$\underline{X} = \begin{bmatrix} X1 \\ X2 \\ X3 \\ X4 \\ X5 \end{bmatrix} = \begin{bmatrix} \text{Retail Inventory} \\ \text{Factory Order Backlog} \\ \text{Averaged Retail Sales} \\ \text{Production Ability} \\ \text{Retail Sales} \end{bmatrix}$$

the simulation model then becomes the fifth-order system:

$$\dot{\underline{X}} = \underline{AX}$$

with

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & .125 & 0 & -.25 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

A state-transition matrix data generation program was used to exercise the model with the initial conditions specified in the reference over a time frame of 0 to 49 units and a step size of 1 unit, giving a total of 50 state vector output points of 5 components each. The state vector values were incrementally loaded into the plot arrays as the simulation progressed. The plotter routine was called at the end of the numerical processing, with the results shown in figure 2. The first four curves, representing the first four components of the state vector, were plotted with the characters A, B, C, and D respectively. The minimum, maximum, and mid-point values for each curve appear at the top of each page of the output in ascending order. Note that the values for E are constant at a level of 120.000, indicating that the fifth state vector component remained at this constant value throughout the run. This aspect of program behavior is also used when plotting less than the full complement of five curves, as the data in a constant vector would plot as a straight vertical line, and are deleted from the plot output to maximize readability. Also note that the plot in figure 2 used IFLAG = 0, and the data point count generated by the plotter appears in the left hand margin.

A brief note about the scaling procedures employed by the program is in order. Under normal operating conditions, scaling is accomplished automatically for each curve without user intervention. If, however, a common scale for each of the five curves is required, the main program must be structured to supply the upper and lower limits of each curve's plot scale in two entries appended to each data array, and to set the point counter to a value equal to the number of the actual data points to be plotted plus two to force the scaling section to consider the supplied values. In order for this procedure to work properly, the lower limit specified must be less than or equal to the smaller expected data point, and similarly, the upper plot limit must be greater than or equal to the largest expected data point.