# AN AUTOMATED METHOD OF CREATING PIECEWISE LINEAR CUMULATIVE PROBABILITY DISTRIBUTIONS

Thomas M. Kisko

University of Florida
Gainesville, Florida 32610

## INTRODUCTION

One of the most fundamental aspects of computer simulation is the generation of stochastic variates. The basic objective is to replicate the underlying stochastic process as accurately as possible. In sumulation languages like GPSS and SIMSCRIPT this is generally done using a pseudo random number generator in conjunction with an "estimate" of the inverted cumulative probability distribution function of the stochastic variate.

A common way of representing the inverted cumulative probability distribution is through the use of a piecewise linear approximation. The topic of this paper deals with the transformation of sample data into piecewise linear cumulative probability distributions.

## DEFINITION OF THE PROBLEM

Given a set of n sample observations from a population, construct a piecewise linear approximation of the cumulative probability distribution function of the observed population. The only assumption regarding the distribution is that it is of the continuous type. The method of producing this piecewise linear approximation should be fast and accurate within a specified error criterion.

## PRESENT METHODS

Several existing methods are used to attack this problem. One technique is to assume that the population has a certain type of distribution (e.g., normal). The sample data are then used to estimate certain parameters of the assumed distribution (e.g., $\sigma$ and $\mu$). The resulting estimated cumulative distribution is then simplified with a piecewise linear estimation. The major drawback of this method is that the assumption of distribution type is not always desirable or possible. A second drawback of the method is that an approximation of the assumed distribution is used.

Another popular method is to generate a frequency histogram of the sample data and then construct a piecewise linear cumulative probability distribution directly from the histogram. Before the frequency histogram can be generated, a decision must be made as to the width of the frequency classes. The shortcoming of this method is that there is no decision that will satisfy all distribution types.

A third alternative is to generate an exact empirical cumulative graph of the sample observations and hand fit a piecewise linear estimation of the graph. This method is both time-consuming and error-prone.

These methods each violate one or more of the problem's constraints. The following proposed method of attacking this problem takes the form of an algorithm. The algorithm essentially automates the hand-fitting process utilizing linear regression. A listing of the 27-step algorithm follows the discussion of the algorithm logic.

## ALGORITHM LOGIC

The algorithm initially generates a set of coordinates that represents estimated points that lie on the cumulative probability distribution function of the population. These points are considered sequentially to extend a regression line until the maximum deviation from any point to the regression line exceeds some limit. When the limit is exceeded the algorithm starts a new regression line of the remaining points and continues as above until the limit is again exceeded. This process continues until all points have been used.

The logic of the algorithm proceeds as follows: let $X_1, X_2, \ldots, X_n$ denote a random sample from a random variable X whose distribution is of the continuous type. Let $Y_1$ be the smallest of these $X_i$, let $Y_2$ be the next $X_i$ in order of magnitude, ..., and let $Y_n$ be the largest $X_i$. That is, $Y_1 < Y_2 < \ldots < Y_n$ represent $X_1, X_2, \ldots, X_n$ when the latter are arranged in ascending order of magnitude. Then the statistic $Y_i$ can be used in the following probability statement:

$$P(X < Y_i) = F_i \text{ where } F_i = \frac{i}{n+1}, \quad 1 \le i \le n \qquad (1)$$

This important statement can be used to show n estimated intersection points of the cumulative distribution of X. These intersection points (see Figure 1) are defined as:

$$P_i = (Y_i, \frac{i}{m+1}),\ i = 1,2,3,\ldots,n \qquad (2)$$

The first regression line starts using $P_1$ and $P_2$. It is then extended to include each successive $P_i$ until a user-controlled error criterion is exceeded. The regression equation (Equation 3) that is used minimizes the vertical distance to the line for all points being considered.

$$F = sY + b \qquad (3)$$

where,

$$s = \frac{(k-j+1)\left\{\sum_{i=j}^{k} Y_i F_i\right\} - \left\{\sum_{i=j}^{k} Y_i\right\}\left\{\sum_{i=j}^{k} F_i\right\}}{(k-j+1)\left\{\sum_{i=j}^{k} Y_i^2\right\} - \left\{\sum_{i=j}^{k} Y_i\right\}^2}$$

and,

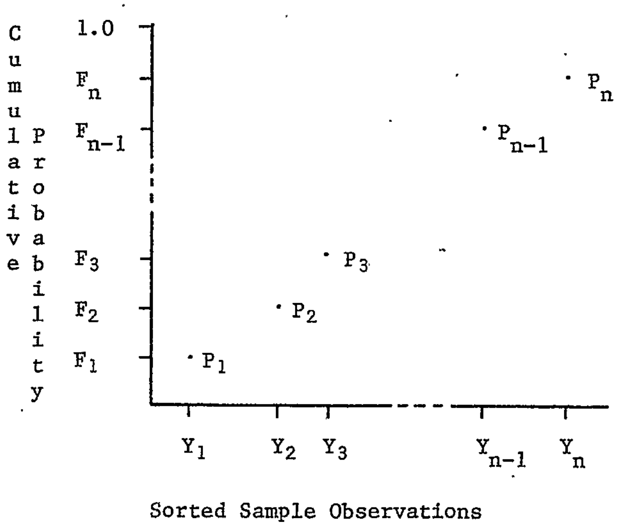
$$b = \frac{\left\{\sum_{i=j}^{k} F_i\right\} - s\left\{\sum_{i=j}^{k} Y_i\right\}}{\{k - j + 1\}}$$

In testing the error, the Y-axis is temporarily normalized to $Y_n = 1.0$ (i.e., $X_{max} = 1.0$). On this temporarily normalized scale, d (Equation 4) is the distance from the current regression line to the most distant $P_i$ associated with the current regression line.

$$d = \underset{j \le i \le k}{\mathrm{MAX}} \left[\frac{F_i - sY_i - b}{\sqrt{1 + Y_i^2 s^2}}\right] \qquad (4)$$

FIGURE 1

Points Used for the Piecewise Regression



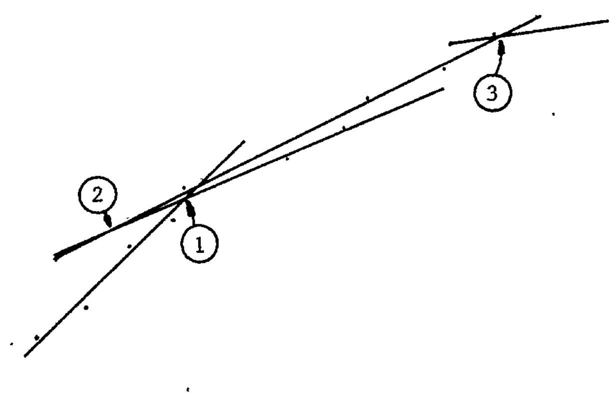

Epsilon, ε, is the user-specified limit of d. If d > ε, the last point considered becomes the first point of a new regression line. This regression line is extended point by point subject to the same constraints as above. Successive regression lines are computed in this manner until they include $P_n$.

A routine has been incorporated in the algorithm that checks to see if the series of intersections of the approximation distribution is monotonic nondecreasing. A "standard fix-up" is initiated to delete the point in error.

From Figure 2 it can be seen that the location of 2 is impossible in a cumulative distribution because it is out of sequence with 1 and 3. This point 2 is simply deleted by the algorithm while still leaving a good approximation of the data points with 1 and 3. It is obvious that 2 can be deleted without a drastic increase in error because this "out of sequence" condition will only result when two curves K and K+1 have slopes nearly the same. Thus, they would define approximately the same line.

FIGURE 2

Sequence Error




## THE ALGORITHM

1. Let $X_1, X_2, X_3, \ldots, X_n$ represent n sample observations of a random variable X whose distribution is of the continuous type.
2. Sort the sample observations $[X_i]$ into ascending order and call the sorted sample set $[Y_i]$, such that $Y_1 < Y_2 < Y_3 \ldots < Y_{n-1} < Y_n$.
3. For each i, $1 \le i \le n$, let $F_i = \frac{i}{n+1}$ and define the point $P_i$ as $(Y_i, F_i)$.
4. Let j=1, k=2, ℓ=1.
5. If k=n+1, then go to step 11.
6. Find the regression line, $R_\ell$, of points $P_j$ through $P_k$.
7. With the Y axis temporarily normalized to $Y_n = 1.0$, let d = the maximum of the

perpendicular distances from the regression line to $P_i$, for each i, $j \leq i \leq k$.

8. If $d \leq \varepsilon$, a predetermined error limit, then let $k = k+1$ and $L_\ell = R_\ell$ and go to step 5.
9. Let $j = k-1$ and $\ell = \ell+1$.
10. Go to step 6.
11. Find the intersection points, $I_i$, of regression lines $L_i$ and $L_{i+1}$ for $1 \leq i \leq \ell-1$.
12. Define $I_0$ as the intersection of regression line $L_1$ and the Y-axis.
13. Define $I_\ell$ as the intersection of regression line $L_\ell$ and the F=I line.
14. If Y of point $I_0$ is negative, then redefine point $I_0$ as the intersection of regression $L_1$ and the F-axis.
15. Let $i = 0$.
16. If point $I_{i+1}$ is not monotonic increasing from point $I_i$, then go to step 19.
17. Let $i = i+1$.
18. If $i = \ell$, then go to step 26, otherwise go to step 16.
19. If $i = \ell-1$, then go to step 24.
20. If point $I_{i+2}$ is not monotonic increasing from point $I_i$, then go to step 24.
21. For each j, $i+1 \leq j \leq \ell-1$, redefine point $I_j$ as point $I_{j+1}$.
22. Let $\ell = \ell-1$ and $i = 0$.
23. Go to step 17.
24. For each j, $i \leq j \leq \ell-1$, redefine point $I_j$ as point $I_{j+1}$.
25. Go to step 22.
26. Points $I_j$, $0 \leq j \leq \ell$, define the intersection points of a piecewise linear approximation of the estimated population cumulative probability distribution of the input data.
27. End.

DISCUSSION OF THE ALGORITHM

Step numbers 1 through 4 initialize variables and sort the sample data. The method of sorting is left to the user. Step numbers 5 through 10 are used to find the regression lines of the points. Each line is sequentially determined through evaluation as described previously. Step numbers 11 through 14 define the intersection points of the piecewise linear polygon to be used as the approximation distribution. Step number 14 was included by this user but should be considered optional. With the step included, the algorithm will not tolerate negative values of sample observations, nor will the resulting distribution contain any negative values. This is useful in generating time distributions that are not allowed to go negative. Step numbers 15 through 25 represent a "standard fix-up" routine. This routine checks to see if the series of intersections of the approximation distribution is monotonic nondecreasing. It should be emphasized that the routine performs a "standard fix-up" and its accuracy cannot be guaranteed. It is essential that a visual inspection be made when a sequence error is noted. It will be left to the user to determine whether or not a "good fit" has been made.

IMPLEMENTATION

The algorithm was implemented as a FORTRAN IV program. The computer program reads sorted sample data according to a user-specified format. After executing, the program lists the intersection points of the approximation distribution and punches, in GPSS format, FUNCTION cards ready for direct input into a GPSS simulation program.

A listing of the computer program along with an example printout follows the text of this paper. It is self-documented with comment cards.

VALIDATION

The initial validation effort was subjective in nature. A visual verification along with a comparison of sample means and expected values was performed. Many sets of sample data with a variety of distributional shapes were tested.

It was shown the algorithm works best with symmetrical distributions. If the distribution had a long tail above the median, the algorithm tended to overestimate the mean. Likewise, if the distribution had a longer tail below the median, the resulting expected value was slightly smaller than the sample mean. In both cases, the difference in means grew smaller as the number of observations increased and as $\varepsilon$ decreased. The magnitude of the difference was typically less than one percent of the sample mean with $n > 100$ and $0.015 \leq \varepsilon \leq 0.002$.

To further validate the algorithm, samples from a known distribution were fed to the computer program. The resulting GPSS FUNCTION was used to create samples that were subjected to a chi-square test.

The exponential distribution was used for the exercise. Samples were generated in a separate FORTRAN program utilizing a uniform random number generator and an inverse cumulative probability function. The samples were then sorted and passed to the main program. Two sample sizes were tested: $n = 100$ and $n = 1000$.

For each of the two sample sizes, four values of $\varepsilon$ were tested: $\varepsilon = 0.015, 0.010, 0.005, 0.002$. The GPSS chi-square test program sampled 1000 values from each distribution. Ten frequency classes were used in the test (i.e., $\nu = 9$). Table 1 summarizes the results of the experiments.

Several interesting observations can be made from these results. First, as $\varepsilon$ decreased, $\bar{d}$, the average distance to the line for all sample points, also decreased proportionately. In the above tests $\bar{d}$ is approximately one-third the value of $\varepsilon$ regardless of the sample size.

A second observation is that as $\varepsilon$ decreased, the number of points on the approximation distribution increased. The smaller sample size tended to increase the number of points faster than the larger sample as $\varepsilon$ decreased.

The chi-square test shows that only two of the experiments resulted in a "good fit". Values for $\varepsilon$ of 0.005 and 0.002 of the one thousand sample experiments resulted in chi-squares within acceptable ranges. This demonstrates two intuitively

TABLE 1

Results of Validation Exercise

| $n$ | $\varepsilon$ | $\bar{d}$ | $\ell$ | $\chi^2$ |
|---|---|---|---|---|
| 100 | 0.015 | 0.0043 | 5 | 58.6 |
| 100 | 0.010 | 0.0031 | 7 | 58.3 |
| 100 | 0.005 | 0.0018 | 14 | 68.8 |
| 100 | 0.002 | 0.0016 | 34 | 73.0 |
| 1000 | 0.015 | 0.0043 | 5 | 45.9 |
| 1000 | 0.010 | 0.0038 | 6 | 29.4 |
| 1000 | 0.005 | 0.0014 | 9 | 11.6 |
| 1000 | 0.002 | 0.0006 | 17 | 11.0 |

Where,
- $n$ - sample size
- $\varepsilon$ - epsilon error limit
- $\bar{d}$ - average error d
- $\ell$ - number of points in GPSS FUNCTION
- $\chi^2$ - chi-square statistic

For $\nu = 9$, $\chi^2_{.95} = 16.9$, $\chi^2_{.05} = 3.33$

obvious points. The larger the sample size the better and the algorithm behaves best when $\varepsilon$ is small.

The value of $\varepsilon$ seems to have no positive effect on the chi-square statistic for the small sample size experiments. This is partially due to the biased nature of the 100 samples. The sample mean was more than 5 % above the expected value and the sample standard deviation was almost 2 % below expectations.

It should be pointed out that the program did do an excellent job in approximating the sample cumulative probability distribution function. If the exact sample cumulative probability distribution function generated from the 100 samples were subjected to the chi-square test, it would have also resulted in a bad fit. However, in cases where no assumption can be made about the shape of the distribution of the population, the sample distribution is the best estimate of the population's distribution regardless of the sample size.

A piecewise linear approximation of the sample distribution is just a condensed form for making the sample distribution more useable in simulation experiments. This validation exercise shows that the program can be an excellent method of converting sample data into simulation-oriented cumulative probability functions.

## CONCLUSION

This author has used the program to create over fifty distributions that actually have been used in simulation models. Where sample sizes were over one hundred, the program generated estimated cumulative probability distributions that accurately replicated the characteristics of the observed sample. In some cases, sample sizes of less than one hundred were used. Due to the peculiar nature of the underlying process being observed, this method proved to be the only way of estimating the distribution of the population. The algorithm has proved itself to be not only reliable and accurate, but also a great time-saver in the simulation process.

## BIBLIOGRAPHY

1. Burington, Richard Stevens, May, Donald Curtis, Jr. Handbook of Probability and Statistics With Tables. McGraw-Hill Book Company, Inc., New York, 1970.
2. Naylor, Thomas H., Balintfy, Joseph L., Burdick, Donald S., Kong Chu. Computer Simulation Techniques. John Wiley and Sons, Inc., New York, 1966.
3. Phillips, Don T. Applied Goodness of Fit Testing. American Institute of Industrial Engineers, Inc., Norcross, Ga., 1972.
4. Shannon, Robert E. Systems Simulation; The Art and Science. Prentice-Hall, Inc., Englewood Cliffs, N. J., 1975.
5. Spiegel, Murray R. Theory and Problems of Statistics. McGraw-Hill Book Company, Inc., New York, 1961.

## COMPUTER PROGRAM LISTING

```
C        AN AUTOMATED METHOD OF CREATING PIECEWISE
C        LINEAR CUMULATIVE PROBABILITY DISTRIBUTIONS
C
C                          THOMAS KISKO
C                    UNIVERSITY OF FLORIDA
C
C
C        INPUT REQUIREMENTS:
C        UNIT 5-PROGRAM SPECIFICATIONS
C              (ONE CARD FOR EACH FUNCTION)
C
C             COL  FORMAT  DEFINITION
C             1-5     A5   FUNCTION NAME
C             6-9     I4   NUMBER OF OBSERVATIONS
C            10-15  F6.3   LARGEST OBSERVED VALUE
C            16-21  F6.4   EPSILON ERROR LIMIT
C            22-61  10A4   FORMAT OF OBSERVATIONS,EG. (2X,F5.2)
C
C        UNIT 2 - SAMPLE DATA
C                 ONE RECORD FOR EACH OBSERVATION
C                 ONE SET OF OBSERVATIONS FOR EACH FUNCTION
C                 EACH SET OF OBSERVATIONS MUST BE IN ASCENDING ORDER
C                 FORMATED ACCORDING TO PROGRAM SPECIFICATION CARD
C
C        NOTES: 1. ALTHOUGH THE USER MAY SPECIFY THE FORMAT OF THE
C                  OBSERVATION DATA AT INPUT SOME PRINT,PUNCH AND INPUT
C                  FORMATS MAY HAVE TO BE MODIFIED BY THE USER TO AVOID A
C                  LOSS OF DATA
C
C                2. THE PROGRAM WILL ALWAYS HANDLE UP TO 1000 OBSERVATIONS
C                  PER FUNCTION. THE CONSTRAINT ON THE NUMBER OF
C                  OBSERVATIONS IS THAT THE PROGRAM CAN ONLY WORK ON
C                  UP TO 1000 UNIQUE VALUES FOR EACH REGRESSION LINE.
C
C        VARIABLE DEFINITIONS:
C
C        AERR-AVERAGE OF ALL PERR'S
C        AREA - AREA UNDER CUMULATIVE DISTRIBUTIONS
C        B - Y INTERCEPT OF CURVE
C        BB(100)-Y-INTERCEPT OF MCTH LINE
C        BP - PREVIOUS DETERMINED Y INTERCEPT
C        CMEAN-MEAN OF OBSERVED DATA
C        ERR-SUM OF PERR'S FOR THIS LINE
C        ERRP - PREVIOUS ERROR
C        FMT(10)-OBJECT TIME FORMAT FOR UNIT 2 FILE
C        I-POINTER FOR X(1000)
C        ILAG= 1 NEW CURVE STARTING (NEW FUNCTION)
C        ILAG = 2 OLD CURVE
C        ITIME-PREVIOUS VALUE FOR TIME
C        ITIME - PREVIOUSLY READ TIME IN CUM (OLD TIME)
C        JC-COUNTER FOR # OF POINTS READ IN (TIME)
C        K - KTH INTERVAL IE KTH XDIV
C        KC-VALUE OF J AT START OF SET OF IDENTICAL VALUES OF TIME
C        MC - # APPROXIMATE CURVES
C        N= NUMBER POINTS / DISTRIBUTION
C        NAME - FUNCTION NAME
C        NOP-NUMBER OF APPROXIMATION POINTS=FINAL VALUE OF MC + 1
C        PERR-SHORTEST DISTANCE TO LINE WITH AXIS NORMALIZED
C        PN-2*N+2 COMPUTATION SAVER
C        RMEAN - MEAN OF APPROX CURVE
C        S - SLOPE OF CURVE
C        SP - PREVIOUS DETERMINED SLOPE
C        SS(100)-SLOPE OF MCTH LINE
C        SUMX -USED TO CALCULATE S AND B
C        SUMXQ-USED TO CALCULATE S
C        SUMXY-USED TO CALCULATE S
C        SUMY -USED TO CALCULATE S AND B
C        TERR-SUM OF ERR'S UP TO THIS POINT
C        TIME-VALUE FROM UNIT 2 FILE
C        X(1000)-SET OF ALL UNIQUE TIME VALUES (FOR THIS LINE ONLY)
C        XINCPT(100)-INTERSECTIONS OF APPROXIMATION CURVES
C        XMAX - MAX TIME OF DISTRIBUTION
C        Y-VALUE OF CUMULATIVE PROB. UP TO THIS POINT
C        YINCPT(100)-INTERSECTIONS OF APPROXIMATION CURVES
C        YP(1000) - MIDPOINT OF YS
C        YT-PREVIOUS VALUE OF Y
C
         REAL*8 SUMX,SUMY,SUMXQ,SUMXY,NAME
```

COMPUTER PROGRAM LISTING (CONTINUED)

```
         REAL YP(1000),SS(100),BB(100),ITIME,FMT(10),
        * X(1000),XINCPT(100),YINCPT(100)
   28    TERR=0.
         ERR=0.
         AREA=0.
         YT=0.
         MC=1
         I=1
   91    READ(5,97,END=99) NAME,N,XMAX,EPS,FMT
   97    FORMAT(A5,I4,F6.2,F6.4,10A4)
          WRITE(6,42) NAME
   42     FORMAT('1'////' FUNCTION NAME ------------------ ',A5/)
          WRITE(6,40) N
   40     FORMAT(' NUMBER OF SAMPLE OBSERVATIONS -- ',I6/)
          WRITE(6,41) XMAX
   41     FORMAT(' MAXIMUM OBSERVED VALUE --------- ',F6.2/)
          WRITE(6,43) EPS
   43     FORMAT(' EPISLON ----------------------- ',F6.4/)
          WRITE(6,44) FMT
   44     FORMAT(' FORMAT OF OBSERVATIONS --------- ',10A4)
         ILAG=2
         Y=0.
         JC=1
         READ(2,FMT) TIME
         KL=1
   94    ITIME=TIME
         KC=JC
   93    JC=JC+1
         IF(JC.GT.N) GO TO 96
         READ(2,FMT) TIME
          IF (TIME.LT.ITIME) GO TO 60
         IF(TIME.EQ.ITIME)GO TO 93
         Y=Y+(JC-KC)/(N*1.0)
         X(I)=ITIME
         KL=KL+1
   96    IF(KC.NE.1)GO TO 27
         GO TO 24
   96    Y=1.
         X(I)=ITIME
         ILAG=1
         IF(JC.NE.2) GO TO 27
         WRITE(6,95) NAME,X(1)
   95    FORMAT('1','******FUNCTION ',A5,' HAS ONLY ONE DATA POINT...TIME='
        *,F14.3,'*******')
         GO TO 91
   60    WRITE(6,61)
   61    FORMAT(' ERROR-OBSERVATIONS NOT IN ASCENDING ORDER')
   99    STOP
   24    S=0.
         B=0.
         PN=2*N+2
         YP(I)=JC/PN
         SUMXQ=X(I)*X(I)
         SUMXY=X(I)*YP(I)
         SUMX=X(I)
         SUMY=YP(I)
    4    I=I+1
         IF(I.GT.1000)STOP
         YT=Y
         GO TO 94
   27    AREA=AREA+(X(I)-X(I-1))*YT
         YP(I)=(JC+KC-1)/PN
         ERRP=ERR
         SP=S
         BP=B
    7    SUMXQ=SUMXQ+X(I)*X(I)
         SUMXY=SUMXY +X(I)*YP(I)
         SUMX=SUMX+X(I)
         SUMY=SUMY+YP(I)
         S=(I*SUMXY-SUMX*SUMY)/(I*SUMXQ-SUMX*SUMX)
         B=(SUMY-S*SUMX)/I
         ERR=0
         DO 5 J=1,I
         PERR=ABS((YP(J)-S*X(J)-B)/SQRT(1+XMAX*XMAX*S*S))
         IF(PERR-EPS) 5,6,6
    5    ERR=PERR+ERR
         GO TO (11,4),ILAG
    6    TERR=TERR+ERRP
```

```
          SS(MC)=SP
          BB(MC)=BP
          X(1)=X(I-1)
          YP(1)=YP(I-1)
          X(2)=X(I)
          YP(2)=YP(I)
          SUMXQ=X(1)*X(1)
          SUMY=YP(1)
          SUMXY=X(1)*YP(1)
          SUMX=X(1)
          I=2
          MC=MC+1
          GO TO 7
 11       TERR=TERR+ERR
          SS(MC)=S
          BS(MC)=B
          CMEAN=XMAX-AREA
C COMPUTING THE LOWER AND UPPER INTERSECTION
          NOP=MC+1
          IF(MC.EQ.1) GO TO 29
          DO 21 I=2,MC
          IM1=I-1
          XINCPT(I)=(BB(I)-BB(IM1))/(SS(IM1)-SS(I))
 21       YINCPT(I)=SS(IM1)*XINCPT(I)+BB(IM1)
 29       YINCPT(1)=0.
          XINCPT(1)=-BB(1)/SS(1)
          IF(XINCPT(1) .GE. 0.0) GO TO 12
          XINCPT(1)=0.
          YINCPT(1)=BB(1)
 12       XINCPT(NOP)=(1.0-BS(MC))/SS(MC)
          YINCPT(NOP)=1.
          WRITE(6,50)
 50        FORMAT(////' POINTS ON ESTIMATED CUMULATIVE PROBABILITY CURVE'
         * //'   POINT   COORDINATES            SLOPE              INTERCEPT')
           DO 52 I=1,MC
           WRITE(6,51) I,YINCPT(I),XINCPT(I)
 51        FORMAT(I5,'     (',F6.4,',',F5.2,')')
 52        WRITE(6,53) SS(I),BB(I)
 53        FORMAT(25X,E12.6,2X,E12.6)
           WRITE(6,51) NOP,YINCPT(NOP),XINCPT(NOP)
C         CHECK - SEQUENCE OF X AND Y POINTS
          NN=-1
 80       NN=NN+1
          M=NOP-1
          DO 84 I=1,M
          IF(XINCPT(I)-XINCPT(I+1))81,83,83
 81       IF(YINCPT(I)-YINCPT(I+1))84,83,83
 83       IF(I.LT.M) GO TO 82
          IF(YINCPT(I+1).EQ.1.0) GO TO 87
          NOP=NOP-1
          GO TO 80
 82       IF(YINCPT(I).GT.YINCPT(I+2).OR.XINCPT(I).GT.XINCPT(I+2)) GO TO 87
          K=I+1
          GO TO 89
 87       K=I
 89       NOP=NOP-1
          DO 88 J=K,NOP
          XINCPT(J)=XINCPT(J+1)
 88       YINCPT(J)=YINCPT(J+1)
          GO TO 80
 84       CONTINUE
C         CALCULATE MEAN OF FUNCTION
          RMEAN=0.
          DO 23 I=2,NOP
 23       RMEAN=0.5*(XINCPT(I)-XINCPT(I-1))*(YINCPT(I)+YINCPT(I-1))
         *+RMEAN
          RMEAN=XINCPT(NOP)-RMEAN
          AERR=TERR/(MC+KL)
          WRITE(6,65) AERR
 65       FORMAT('0THE AVERAGE ERROR WAS ',F7.5)
           IF(NN.NE.0) WRITE(6,63) NN
 63        FORMAT('0WARNING : BECAUSE OF SEQUENCE ERRORS',I3,
         * ' POINT(S) WERE DELETED')
           WRITE(6,64) CMEAN,RMEAN
 64        FORMAT(///' THE MEAN OF THE SAMPLE OBSERVATIONS WAS      ',F9.3,
         *            //' THE EXPECTED VALUE OF THE GPSS FUNCTION IS ',F9.3,
         *            //' BELOW IS THE LISTING OF THE FUNCTION'///)
 74       IF(NOP .GE. 100) GO TO 17
```

## COMPUTER PROGRAM LISTING (CONTINUED)

```
          IF(NOP .GE. 10) GO TO 18
          GO TO 19
17        WRITE(7,14) NAME,NOP,RMEAN
          WRITE(6,14) NAME,NOP,RMEAN
          GO TO 13
18        WRITE(7,15) NAME,NOP,RMEAN
          WRITE(6,15) NAME,NOP,RMEAN
          GO TO 13
19        WRITE(7,16) NAME,NOP,RMEAN
          WRITE (6,16) NAME,NOP,RMEAN
14        FORMAT(1X,A5,T8,'FUNCTION',T19,'RN1,C',I3,T36,'* MEAN= ',F9.3)
15        FORMAT(1X,A5,T8,'FUNCTION',T19,'RN1,C',I2,T36,'* MEAN= ',F9.3)
16        FORMAT(1X,A5,T8,'FUNCTION',T19,'RN1,C',I1,T36,'* MEAN= ',F9.3)
13        WRITE(7,199) (YINCPT(I),XINCPT(I),I=1,NOP)
          WRITE(6,199) (YINCPT(I),XINCPT(I),I=1,NOP)
199       FORMAT(6(F6.4,F6.2))
299       FORMAT('0',6(F6.4,F6.2))
          GO TO 28
          END
```

## EXAMPLE OF COMPUTER PROGRAM PRINTOUT

```
FUNCTION NAME ------------------ TEST
NUMBER OF SAMPLE OBSERVATIONS --       10
MAXIMUM OBSERVED VALUE ---------  85.00
EPSILON ----------------------- 0.0150
FORMAT OF OBSERVATIONS --------- (F3.0)


POINTS ON ESTIMATED CUMULATIVE PROBABILITY CURVE
 POINT  COORDINATES          SLOPE        INTERCEPT
   1   (0.0870,   0.0 )
                         0.605164E-02  0.870091E-01
   2   (0.2734,  30.79)
                         0.361896E-01  -.841040E 00
   3   (0.6343,  40.77)
                         0.622545E-02  0.380524E 00
   4   (1.0000,  99.51)
THE AVERAGE ERROR WAS 0.00192

THE MEAN OF THE SAMPLE OBSERVATIONS WAS        40.600
THE EXPECTED VALUE OF THE GPSS FUNCTION IS     41.432
BELOW IS THE LISTING OF THE FUNCTION

TEST   FUNCTION    RN1,C4           * MEAN=    41.432
.0870   0.0 0.2734 30.790.6343 40.771.0000 99.51
```