

and the second second

HILGARDIA

JOURNAL OF AGRICULTURAL SCIENCE PUBLISHED BY HE CALIFORNIA AGRICULTURAL EXPERIMENT STATION

# Computer Generation of Points on a Plane

Volume 43, Number 5 · June, 1975

# Treatment of Boundary Line Overlap in a Forest-Sampling Simulator

Lee C. Wensel

USA ISSN 0073-2230

UNIVERSITY OF CALIFORNIA DIVISION OF AGRICULTURAL SCIENCES



#### COMPUTER GENERATION OF POINTS ON A PLANE

A statistical model for the generation of random, contagious, and uniform spatial patterns is developed. Points are located on the plane one at a time with each point modifying the probability matrix for the next point. For random patterns no change is made in the probabilities. For contagious patterns, location of a point increases the probability of locating another point near it. However, for uniform patterns the probability of locating another point near previously established points is reduced.

#### TREATMENT OF BOUNDARY LINE OVERLAP IN A FOREST-SAMPLING SIMULATOR

A procedure is given for treating boundary line overlap in computer simulated sampling. This procedure, referred to as algorithm EDGE, insures that each point in the rectangular population has the same probability of being included in the sample, thereby eliminating possible edge-effect bias. The effectiveness of EDGE in producing a more realistic variance/plot size relationship is demonstrated by comparing the variance functions with uncorrected samples and samples corrected using a previously reported weighting scheme.

#### THE AUTHOR:

Lee C. Wensel is Associate Professor of Forestry, Department of Forestry and Conservation, College of Natural Resources, University of California, Berkeley, California 94720.

# Treatment of Boundary Line Overlap in a Forest-Sampling Simulator

SIMULATION STUDIES USING either real populations that have been mapped or artificially-generated populations promise to be highly useful in studying the characteristics of biological communities. However, in dealing with these representations of real populations one must be careful in treating population elements located near the edge of the population. This is particularly important in computer-stored populations because they are usually quite small. This paper proposes a procedure for projecting opposite sides of a population onto one another, thus making the computer-stored population effectively without boundaries. That is, population elements along opposite boundaries are treated as if they are physically adjacent to each other. An application of this concept in generating spatial patterns is made in subroutine LOCATE (see Appendix, starting page 147).

#### COMPUTER SAMPLING OF MAPPED POPULATIONS

In studying the efficiency of various forest-sampling designs it is common practice to map forest stands and store the Cartesian coordinates of each tree. together with the characteristics of that tree (diameter, height, volume, etc.), in a computer for study. This data set is then "processed" by a series of computer operations to simulate operations that would take place if the stand were sampled physically in the field. Using this technique, the efficiency of many different sampling designs may be tested easily once the initial mapping has been done. (O'Regan and Palley, 1965; Kulow, 1966; Wensel and John, 1969.)

Whether sampling in the field or with a computerized simulator, there exists the problem of getting a representative sample of trees near the forest's edge. If the probability of selecting a tree near the edge is different from the probability of selecting a similar tree in the interior, and if that probability difference is not considered in the estimate, the resulting estimator is biased. This is referred to as "slopover" bias (Grosenbaugh, 1958).

Let us consider fixed-radius plot sampling with a plot size of a acres. Plot size a is referred to as the *nominal* plot area, while the actual plot area may be somewhat less than a because of the boundary overlap. In figure 1, a tree is counted if the randomly located sample point falls within the plot area shaded for each tree. For tree 1, actual and nominal plot areas are equal; for tree 2, the actual plot area is less than the nominal because of boundary overlap. Unbiased estimates of the total number of trees, total basal area, etc., will result if the actual probabilities of selecting the individual trees are used in developing the estimators.

Let us also consider the following esti-

<sup>&</sup>lt;sup>1</sup> Submitted for publication October 26, 1973.



Fig. 1. Tree-plot boundary overlap.

mator  $\hat{N}$  of N, the total number of trees in a forest of size A:

$$\hat{N} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k_i} \sum_{j=1}^{k_j}$$

where n is the number of randomly located sample points,  $k_i$  is the number of trees counted at point i, and  $P_j$  is the probability of selecting tree j from a randomly located plot center. Note that

$$P_j = \frac{a_j}{A}$$

where  $a_j$  is the plot area containing all sample points that would cause tree jto be selected. If actual probabilities or plot areas are used,  $\hat{N}$  is an unbiased estimator of N. Wensel and John (1969) give the necessary equations to compute the actual plot area using distances measured to the nearest boundaries. This approach, while time-consuming in the field and in the computer, gives exact results. However, for extremely small

To compute the distance  $r_i$  that a point *i* with coordinates  $(x_i^i, y_i^i)$  is from a plot center at point  $(x_o, y_o)$  we proceed as follows:

Fig. 2. Projection of overlap area to opposite boundaries.

populations where nominal and actual probabilities of selection can be quite different, the variance function (variance as a function of plot size) of the mapped stand may be somewhat different from the actual population being investigated.

In computer-stored populations it is possible to project opposite sides of the population onto one another so that, for example, the plot area that overlaps on the south boundary is transferred to the north boundary. Tree 2 in figure 2 would be counted if the randomly located sample points fall anywhere in the shaded areas. Thus the actual and nominal plot areas are always the same and p, the probability of selecting a given tree, is a where a is the nominal plot

area. The estimator  $\hat{N}$  of N then becomes

$$\hat{N} = \left(\frac{\mathbf{l}}{n}\right) \left(\frac{A}{a}\right) \sum_{i=1}^{n} k_{i}.$$

#### ALGORITHM EDGE

$$x_{i} = \min [|x_{i}^{2} - x_{o}|, x_{\max} - |x_{i}^{2} - x_{o}|]$$
  

$$y_{i} = \min [|y_{i}^{2} - y_{o}|, y_{\max} - |y_{i}^{2} - y_{o}|]$$

HILGARDIA · Vol. 43, No. 5 · June, 1975

$$r_i = \sqrt{x_i^2 + y_i^2}$$

where  $x_{\max}$  and  $y_{\max}$  represent the maximum coordinates in the x and y directions, respectively. The individual *i* would be included in the sample only if

 $r_i$  is smaller than the plot radius R. A brief FORTRAN function to determine which individuals should be included in the plot when using this algorithm is given below.

	FUNCTION EDGE (R, X, Y, XO, YO, XMAX, YMAX)
С	
С	EDGE = 0. IF POINT IS OUTSIDE RADIUS R
С	<b>EDGE = 1. IF POINT IS INSIDE RADIUS R</b>
С	WHERE
С	(X,Y) = COORDINATES OF POINT I
С	(XO,YO) = COORDINATES OF PLOT CENTER
С	$\mathbf{R} = \mathbf{PLOT} \mathbf{RADIUS}$
С	POPULATION DIMENSIONS ARE XMAX AND YMAX
C.	
	$\mathbf{XX} = \mathbf{ABS}(\mathbf{X} - \mathbf{XO})$
	XX = MIN1(XX, (XMAX - XX))
	$\mathbf{Y}\mathbf{Y} = \mathbf{A}\mathbf{B}\mathbf{S}(\mathbf{Y} - \mathbf{Y}\mathbf{O})$
	YY = MIN1(YY, (YMAX - YY))
	EDGE = 0.
	$IF(XX^{**2} + YY^{**2} . LE. R^{**2}) EDGE = 1.$
	RETURN
	END

It is here assumed that any plot radius R is less than one-half the smallest dimension of the area being sampled. Because the plot area must be small relative to the area of the population being studied for the sampling problem to make sense at all, this simplifying assumption is not restricting.

## EFFECT OF ALGORITHM EDGE ON SIMULATED SAMPLING

In computer-simulated forest sampling, one starts with a matrix of tree coordinates and tree characteristics and selects random sample points within the boundaries of the forest. Wensel and John (1969) corrected for the fact that some randomly selected plots overlap the forest boundaries by computing the actual plot area to be used in the estimators (algorithm wgr).

Using data from the Cutfoot Experimental Forest (Wensel and John, 1969), figure 3 shows the variance functions for no-overlap correction, WGT correction discussed by Wensel and John (1969), and for the EDGE correction discussed above. These data resulted from the selection of 200 randomly located points within the bounds of the computer-represented Cutfoot forest and then constructing each of the three estimates at each point.

The uncorrected estimates resulted in a much higher variance than did corrected estimates (for all except the smallest plot sizes). The WGT-corrected estimates are plotted on nominal plot sizes and not on actual plot sizes—plotting on the actual plot size would move the WGT-corrected line to the left and make it even closer to the EDGE-corrected line shown in figure 3. In terms



Fig. 3. Logarithm of variance of numbers of trees per acre on the logarithm of plot size.

HILGARDIA · Vol. 43, No. 5 · June, 1975

of statistical efficiency, the WGT and EDGE corrections are comparable.

In computer-simulated sampling studies, EDGE correction permits each tree within the population to be treated the same regardless of its proximity to the boundary.

### LITERATURE CITED

BEERS, T. W.

1964. The direct correction for boundary-line slopover in horizontal point sampling. Purdue Ag. Expt. Sta. Research Progress Report 224.

GROSENBAUGH, L. R.

1958. Point sampling and line sampling: probability theory, geometric implications, synthesis. So. For. Expt. Sta. Occ. Paper 160.

KULOW, D. L.

1966. Comparison of Forest Sampling Designs. Journal of Forestry 64(7):469-74.

O'REGAN, W. G. and M. N. PALLEY

1965. A Computer Technique for the Study of Forest Sampling Methods. For. Sci. 11(1):99-114.

WENSEL, LEE C.

1975. Computer Generation of Points on a Plane. Hilgardia 43(5):131-42.

WENSEL, L. C. and H. H. JOHN

1969. A Statistical Procedure for Combining Different Types of Sampling. For. Sci. 15(3): 307-17.

### APPENDIX

#### Computer Program<sup>1</sup>

$\mathbf{Routine}$	Function
PROGRAM STAND	Main control of program
SUBROUTINE MODIFYP	Modify probability matrix after each individual is located.
SUBROUTINE LOCATE	Locate the coordinates corresponding to the ran- dom number drawn.
FUNCTION F	Regular probability modification function
FUNCTION G	Contagious probability modification function
SUBROUTINE PLOT	Plot the pattern generated on the line printer
FUNCTION RANDOM	Generate uniform pseudo-random numbers
FUNCTION LSHIFT	Packs characters into computer words for MAP

<sup>1</sup> SUBROUTINE TSORTE, which sorts the indices of coordinates points modified by MODIFYP, is not listed. The user is advised to use whatever sort routine is available on his local computer.

```
148
                                     Wensel: Treatment of Boundary Line Overlap
               STAND(INPUT, OUTPUT, TAPE6=OUTPUT, PUNCH)
      PROGRAM
С
С
С
C
      COMMON /DATA/ N+NX+NY+NXY+W+A
      COMMON /ID/ ID
      COMMON /SCALER/ X+ Y
      DIMENSION P(6401) + KOURD(200)
      DIMENSION JP(4+80)
      EQUIVALENCE (P+JP)
      INTEGER RR(200)
      DIMENSION AMEAN(10) . AVAR(10) . RADIUS(10) . PLOTS(10) . TREE(10) .
                 XX(10),YY(10),NAMFILE(8),
                                                   C(2+200)+BOUND(2+2)
     1
      COMMON /BLOCK/BOUND.UNIT.SEED.N1.N2.N4.12.KP.NAMFILE
      COMMON/MOD/IX.IY.DMAX.SUMP1.SUMP2.IK.DNN
      COMMON /GRID/ GRID
      COMMON /PAR/BA+ BB+ BK+ B2+ X0+ X1+ HM+ R
      COMMON /IP/ IP(25)
С
      DOUBLE PROB. SUMPROB
      READ 916.NAMFILE
      READ 917.M.N1.N2.N4.I2.KP.UNIT
      READ 918.NOTE. (RADIUS(I).I=1.M)
C*****MODIFICATION FOR MULTIPLE SEEDS
      READ 905. N. NX. NY. NXY. W. A
      READ 910. NITER. SEEDI
      IF (NOTE.EQ.3HSIZ) GO TO 15
      DO 13 I=1.M
   13 PLOTS(I)=(3,1415926536*RADIUS(I)**2)/UNIT
      GO TO 18
   15 DO 17 I=1.M
      PLOTS(I)=RADIUS(I)
   17 RADIUS(J)=SORT(UNIT*PLOTS(I)/3.1415926536)
   18 CONTINUE
      BOUND(1.1)=0.
      BOUND (2.1)=0.
      BOUND(1+2)=NX
      BOUND (2+2)=NY
      DO 90 ITER=1+NITER
      PRINT 919. NAMFILE
      READ 906. ID. HM. XO. XI. BA. BB. BK. B2
      SFED = SEED1
      N2 = 1
      N = (NXY/W**2) * (4./3.14156)
      PRINT 50
      PRINT 902. N. NX. NY. NXY. W. A. ID
      PUNCH 912. N.NX.NY.ID.W.A.HM.XO.X1.BA.BB.BK.B2
      PUNCH 1912
      IF (ID)6.7.8
    6 PRINT 1902+ X0+ HM+ BA+ BB+ BK
      PRINT 907
      PRINT 908
      GO TO 9
    7 PRINT 913
      GO TO 9
```

```
HILGARDIA • Vol. 43, No. 5 • June, 1975
                                                                 149
     8 PRINT 2902. X0. X1. HM. BA. BB. BK. B2
       PRINT 914
       PRINT 915
С
     GENERATE INITIAL PROBABILITIES
     9 PROB = 1.0/NXY
       NXY1=NXY+1
       SUMPROB = -PROB
       DO 10 I = 1. NXY1
       SUMPROB = SUMPROB + PROB
    10 P(I) = SUMPROB
       DNN = W
       GRID=SORT(A/NXY)
C
      DMAX IS THE MAXIMUM DISTANCE FROM TREE AT WHICH ANY POINT WILL
С
              BE MODIFIED.
С
      DMAX = DNN+X1
      R=DMAX/GRID
                    +0.5
      PRINT 904, GRID. DNN. DMAX
    CHECK ROUND OFF ERROR
С
С
    I-TH TREE TO BE LOCATED HAS DIAMETER DIAM(I)
С
      ISTART = 0
      PRINT 910.ISTART.SEED
      PRINT 800
      DO 30 1=1.N
      II = I
    SELECT RANDOM NUMBER
С
      RAND=RANDOM(SEED)
    LOCATE INDEX K SUCH THAT P(K=1) .LT. RAND .LE. P(K)
С
С
      FATAL = 0.
      K = RAND*NXY
      CALL LOCATE (RAND. K. P. NXY1)
      K=K=1
      IX = MOD(K = 1 + NX) + 1
      IY = K / NX + 1
      C(1,I)=IX
      C(5,1)=IA
       KOORD(II) = IX*8**5 + IY
   25 PRINT 901.1, RAND, K. RAND. IX. IY
       IF (TD.NE.0)
      ICALL MODIFY P (ID. K. P. ISTART, FATAL, NXY1.
                                                           NK)
       IF ( FATAL.GT.0. ) GO TO 90
    30 CONTINUE
       PUNCH 916. NAMFILE
       PUNCH 909. (KOORD(I). I=1.N)
       CALL PLOT (KOORD.N .8.80.77777000008.777778.1.8.JP.15.0.NAMFILE)
       CALL SPATIEL (AMEAN, AVAR, RR, C, RADIUS, PLOTS, TREE, XX, YY, MON)
    90 CONTINUE
       PRINT 903. FATAL
       STOP
    50 FORMAT (1H + //// 15X+ *DATA* //+ 10X+ *SYMBOL*+ 3X+ *VALUE*+ 7X+
      1 *DEFINITION*/+ 10X, *----++ 3X, *----++ 7X, *----+/)
   800 FORMAT (1H1, // 7X, *I*, 9X, *R*, 9X, *K*, 8X, *P*,28X, *SUMP1*,
      1 5X. *SUMP2* /)
   901 FORMAT ( 18, F12.5, 18, 10X, 020, 2110)
```

Wensel: Treatment of Boundary Line Overlap

902 FORMAT (12X, \*N\*+ 2X+ I8+ 6X+ \*NUMBER OF INDIVIDUALS\* //+ 11X. \*NX\*.2X. IS. 6X. \*NUMBER OF POINTS ON GRID IN X DIRFC 1 STION\* //.11x.\*\* Y\*, 2X, 18, 6X.\*NUMBER OF POINTS ON GRID IN Y DIREC 3TION\* //.JOX.\*NXY\*.2X .I8.6X.\*TOTAL NUMBER OF POINTS ON GRID\*//. 4 12X. +W+, 2X, F8.2.6X .+AREA / (ND. OF TREES) 5/. 12X. \*4\*. F10.0. 6X. \*AREA COVERED BY POPULATION\* 6 // 11X. \*ID\*. 110. 6X. \*INDEX OF CONTAGION\*. //) 1902 FORMAT ( 11X+\*X0\*+F10.2+ 7 6X . \* )\* • /11X\*HM\*•F10\_2•6X•\*)\* · /12X+\*A\*. 8F10.5.6X.\*) PARAMETERS\*. /12X\*8\*.F10.5.6X.\*)  $Y = F(A \cdot B \cdot K \cdot X) *$ 9 /12X+\*K\*+F10.5+6X\*)\* //) 2902 FORMAT (11X, 2HX0+F11+2, 6X+ 1H) / 11X+ 2HX1+ F11+2+ 6X+ 1H) / 1 11X, 2HHM, F11.2, 6X, 1H) / 12X, 1HA, F11.5, 6X, 32H) G(X)=A+R+ FOR X .LE. 1.0/ 12X.1HB.F11.5. 6X. 1H) / 11X. 2HA2. F11.5. 2X G(X)=A2+H2\*X FOR 1.0 .LT. X .LE. X1 / 11X.2HB2.F11.5. 3 6X.40H) 4 6X. 1H) ) 903 FORMAT (+1FATAL ERROR NUMBER \*+F3.0+\* CALL EXIT\*) 904 FORMAT (F22.4+ E27.4+ E13.4 ) 905 FORMAT (4(4x+16), 2(4x+F6.0)) 906 FORMAT (4X+ 16+ 7(2X+F8.0)) 907 FORMAT (201 +50H HM T ..... REGULAR 1 /20X. 50H 1 5/20X. 50H ī STAND 3 / 20%. 50H I 4 /20X. 50H T 5 /20X, 50H I 6 /20X. 50H 1 6 /20X. 50H1.0 T \*\*\*\*\*\*\* 50H 7 /20X+ T 8 /20X. 50H T ١ 908 FORMAT(20X+50H T 1 /20X. 50H T 5 /SOX\* 50H 1 . 3 /20X+ 50H T 50H0.0 I\* 4 /20X+ 5 /15X.45HREL. 0.0 X 0 2-x0 1.0 6 /15X,45HGFID DNN DMAX ) 909 FORMAT(8010) 910 FORMAT (13+2X+020) 912 FORMAT ( 415, F5,0, F7.0, 3F5,2, 4F8,4 ) 1912 FORMAT (78H N NX NY ID W ARFA HM X O X 1 ۸ K/A2 1 В 82) 913 FORMAT (20X, 50H HM J RANDOM STAND 50H I 1 /20X. 5 150X. 3 /20X. 50H T 4 /20X. 50H T . 5 /20X. 50H T 50H I 6 /20X. 50H0.0 I\*-7 /20X, 8 /15X.45HREL. 0.0 1.0 9 /15X.45HGRID DNN ) 914 FORMAT(20X+50H HM I CONTAGIOUS STAND 1 /20X+ 50H 1 \* 5 /20X+ 50H I 3 /20X. 50H I

150

4 /20X.	50H	I *				
5 /20X.	50H	1	*			
6 /20X.	50H	I	*			
7 /20X.	50H1.0	I	*		*	
8 /20×,	50H	I	• *		* .	
9 /20X.	50H	I	•	* *	•	)
915 FORMAT(2	0X+50H	Ţ	•	*	•	
1 /20X.	50H	1	•	•	•	
5 /20X+	50H	I	•	•	•	
3 /20X.	50H	I	•	•	•	
4 /20X+	50H0.0	J *	*			
5 /15X+45	HREL. 0	• 0	<b>X</b> 0	1.0	× 1	
6 /15X+45	HGRID			DNN	DMAX	)
916 FORMAT (	8A10)					
917 FORMAT (	4(15+5X)+A	5+5X+15+F1	0.2)			
918 FORMAT (	5X+A5+10F5	• 0 )				
919 FORMAT (	1H1,8A10)					
END						

```
152
                                      Wensel: Treatment of Boundary Line Overlap
    SUBROUTINE MODIFY P ( ID+KXY+P+ ISTART+ FATAL+ NXY1+
                                                                NK)
 EXISTING PROBABILITIES ARE MODIFIED BASED UPON
    THE OUTCOME OF THE PREVIOUS TREE LOCATION.
  THE MAGNITUDE AND SIGN OF ID ARE USED TO SPECIFY THE KIND
    OF MODIFICATION TO HE MADE. ID= -3+-2+-1+0+1+2+3
         ID NEGATIVE = REGULAR DISTRIBUTIONS
         ID ZERO
                     = RANDOM DISTRIBUTION
         ID POSITIVE = CONTAGIOUS DISTRIBUTIONS
 ISTART = 0 TO INITIALIZE CONSTANTS
   DIMENSION P ( NXY1), INDX (850),0(850)
   COMMON /DATA/ N.NX.NY.NNN.W.A
   COMMON/MOD/IX+IY+DMAX+SUMP1+SUMP2+IK+DNN
   COMMON /GRID/ GRID
   COMMON /IP/ IP(25)
   COMMON /PAR/BA+ BP+ BK+ B2+ X0+ X1+ HM+ R
   SUMP1 = 0.
   SUMP2 = 0.
   NXY=NXY1=1
   IF (ISTART .NE. 0 ) GO TO 25
 I. MODEL
   REGULAR DISTRIBUTION
   (A) EQUATION Y=K+X+(B=EXP(-A+X))
   (B) FOR 0.LE.X.LE.X1.0.Y=F(X)=K*X*(B=EXP(-A*X))
   (C) FOR 1.0 .LT.X.LT. (2=X0) + Y= F(2=X)
           WHERE X_2 = (X_0 - (X - X_0)) / (1 - X_0)
   (D) FOR 1.LT.X.LF. (2-X0). Y=F1(-X3)
           WHERE X3 = (X0 - ((2+X) - X0) / (1+X0))
    CONTAGIOUS DISTRIBUTION
    (A) EQUATION X=DISTANCE FROM TREE
       (FOR X.EQ.0) Y=0.
                          Y=A+BX
       (FOR 0.LT.X.LE.1)
       (FOR 1.LT.X.LT.X1) Y=A2+B2*X
II. VARIABLES, SCALF
    DNN=DISTANCE TO NEAREST NEIGHBOR ON HEXAGONAL SPACING.
           DNN=W/2.*SQRT(5.) WHERE W=SQRT(A/N).
        DNM IS THE SCALING PARAMETER ON THE X AXIS FOR THE MODELS
            IN SECTION I.
      AWAREA COVERED BY POPULATION
      NENUMBER OF INDIVIDUALS IN POPULATION
   GRID=PHYSICAL DISTANCE BETWEEN POINTS ON GRID (SQRT(A/NXY))
    NXYSTOTAL NUMBER OF POINTS ON GRIDENX*NY
   PONNEPROBABILITY PEAK AT DNN. SCALING PARAMETER ON THE Y AXIS FOR
```

000000000

C C

С С С С С С č Ç C С С С С С С С С С С С С С C С С С С С С С C С С С

C

```
153
HILGARDIA • Vol. 43, No. 5 • June, 1975
               THE MODELS IN I, EXPRESSED INITIALLY AS PERCENT OF YO THEN
C
               TRANSFORMED TO PROBABILITY BY PDNN=PDNN/100.*YO
С
C
С
      SUMP1 = 0.
      SUMP2 = 0.
   25 \text{ ISTART} = 1
С
      DEFINE COORDINATE LIMITS OF POINTS TO BE MODIFIED.
С
С
      IX = MOD(KXY=1+NX) + 1
      IY=(KXY-1)/NX+1
      KK = 0
      J=0
       I = INT(R)
     1 IF(SORT(FLOAT(I*I+J*J)) .LE. R) GO TO 2
      MM = IX + NX - J + 1
      MN = IX + NX + J = 1
      NM=IY+NY=I
      NN=IY+NY+I
      I=I=1
      GO TO 3
     2 MM=IX+NX=I
       MN=IX+NX+I
       NM=IY+NY+J
       NN=IY+NY+J
       J=J+1
     3 DO 4 K=MM+MN
       LI=MOD(K-1+NX)+1
       LJ=MOD(NM=1,NY)+1
       II = K - IX - NX
       IJ = NM - IY - NY
       IF (ID.LT.0) GO TO 6
     5 FUNCT=G(II+IJ+BA+BB+BK+B2+X0+X1+DNN+GRID)
       GO TO 7
     6 FUNCT=F(II+IJ, BA, BB, BK, X0, DNN+ GRID )
     7 KIJ = (LJ=1) \neq NX + LI
С
C
       ALLOW FOR ZERO ELEMENT IN ARRAY
C
       PP = P(KTJ+1) = P(KTJ)
       SUMP1=SUMP1+PP
       PP = PP* FUNCT
       KK = KK + 1
       Q(KK)=PP
       SUMP2=SUMP2+PP
       INDX(KK) = KIJ
       IF (NN .EQ.NM) GO TO 4
       LJ=MOD(NN=1+NY)+1
       KIJ = (LJ-1)*NX+LI
C
       ALLOW FOR ZERO ELEMENT IN ARRAY
C
C
       PP = P(KIJ+1) = P(KIJ)
       SUMP1=SUMP1+PP
       PP = PP* FUNCT
```

```
KK = KK + 1
      Q(KK)=PP
      SUMP2=SUMP2+PP
      INDX(KK) = KIJ
    4 CONTINUE
      IF(I .GE. J) GO TO 1
С
    RESCALE P FOR POINTS REFERENCED IN MOD
С
      PRINT 95. SUMP1. SUMP2
С
      DD 30 I=1+KK
   30 Q(I) = Q(I) + SUMP1 / SUMP2
   90 FORMAT (60X, 2F10.6)
   91 FORMAT(2H (+ 14+ 1H++ 14+ 1H)+ 7H KIJ =+ 14+ 5H D =+ F7.2)
      D0 220 1=1+KK
  220 INDX(I)=INDX(I)*4096+I
      CALL SORTR(INDX,-KK,1)
      MIN=INDX(1)/4096
      MAX=INDX(KK)/4096
      P_{1=P(MIN)}
      J=1
      KIJ=MIN
      DO 240 TEMIN.MAX
      P2=P(I+1)
      IF (I.ER.KIJ)G0 TO 230
      P(I+1)=P(I)+P(I+1)=P1
      GO TO 240
  230 IJ=INDX(J)=4096*KIJ
      P(I+1) = P(I) + Q(IJ)
      J=J+1
      KIJ=INDX(J)/4096
  240 P1=P2
      GO TO 9999
 9999 RETURN
      END
```

```
SUBROUTINE LOCATE (RAND+K+P+NXY)
                                                           VERSION 6/15/70
   LOCATE INDEX K SUCH THAT P(K=1) .LT. RAND .LE.P(K)
   DIMENSION P(NXY)
   IF (K.LT.1) K=1
IF (K.GT.NXY) K=NXY
   INC = 8
20 IF (JNC.LT.1) INC=1
    TEST FOR K=0
   IF (K.GT.INC) GO TO 30
   K=INC
   IF (P(INC).GF.RAND) GU TO 60
   K=2+INC
30 IF (RAND.LF.P(K)) GO 10 50
IF (K+INC .GT. NXY) GO TO 60
40 K=K+INC
   IF (RAND.LE.P(K)) GO TO 60
IF (K+INC.LT.NXY) GO TO 40
   GO TO 60
50 IF (RAND.GT.P(K=INC)) GO TO 60
   K=K=INC
   IF (K-INC.GT.0) GO TO 50
60 IF (INC.FO.1) GO TO 70
   INC=INC/2
   GO TO 20
70 CONTINUE
   RETURN
   END
```

```
FUNCTION G (I.J.A.B.A2.B2.X0.X1.DNN.GRID)
   CONTAGIOUS DISTRIBUTION
   MODEL .... G = A + B*X
  CALCULATE PROBABILITY MODIFICATION
        X = DISTANCE FROM ORIGIN
        X0 = DISTANCE WHERE RELATIVE PROBABILITY INCREASE CROSSES 1.0
  G = 1.
  X=FLOAT(]) *GRID/DNN
   Y=FLOAT(J) +GRID/DNN
   X = SQRT(X + X + Y + Y)
   IF (X.GT.X1) GU TO 40
IF (X .GT. 1.) GO TO 10
5 G = A + B*X
  IF (X .EQ. 0.) G = 0.
40 RETURN
10 G=A2+82*X
   G0 T0 5
   END
```

С

C

С

С

C C

С

C C

C C

с с

С

```
FUNCTION F (I.J.A.R.K.X0.DNN.GRID)
    DIMENSION C(3)
    REAL K
     REGULAR DISTRIBUTION
     MODEL_{\bullet\bullet\bullet\bullet}F(A \bullet B \bullet K \bullet X) = K * X * * P * (1, 0 - E X P (+ A * X))
    CALCULATE PROBABILITY MODIFICATION
           X = DISTANCE FROM ORIGIN
           X0 = DISTANCE WHERE RELATIVE PROBABILITY INCREASE CROSSES
           FROM .LT. 1 TO .GT. 1
YO = MAXIMUM PROHABILITY INCREASE (PEAK OF CURVE).
    F = 1.
X=FLOAT(I) *GRID/DNN
    Y=FLOAT(J) *GRID/DNN
    X = SQRT(X + X + Y + Y)
 IF (X .GT. 2.-X0) GO TO 40
IF (X .GT. 1.) GO TO 10
5 F = K*X**B*(1.0-EXP(-A*X))
40 RETURN
10 X = 2, - X
GO TO 5
    END
```

```
158
                                       Wensel: Treatment of Boundary Line Overlap
  SUBROUTINE PLOT (IZ,NPTS,MAXABS,MAXORD,ABSMASK,ORDMASK,ISCALE,
 1LINE.
              TP+IXPACK+ IYPACK+BLURP)
  SUBROUTINE PLOTS POINTS ON PRINTER WITHMAXABS AND MAXORD BEING THE
  MAXIMUM NUMBER OF GRID POINTS IN THE HORIZONTAL (X) AND VERTICAL (Y)
  DIRECTIONS.
  NPTS
             NUMBER OF POINTS TO RE PLOTTED
             MASK TO BE APPLIED TO Z TO OBTAIN X
  ABSMASK
  ORDMASK
             MASK TO BE APPLIED TO Z TO OBTAIN Y
  SCALE
             NUMBER OF UNITS BEYWEEN GRID POINTS ON PLOT
  LINE
             MAXIMUM NUMBER OF POINTS TO BE PLOTTED PER LINE DIVIDED BY 10
  BLURP VECTORUE LENGTH 10 TO BE PRINTED (10A10) TO IDENTIFY PLOT
  DIMENSION BLURP(8)
  DIMENSION IZ(NPTS) . IP(MAXAHS, MAXORD)
  K=MAXABS*MAXORD
  DO 5 I=1.K
5 IP(I)=0
  K = 458
  DO 10 1=1.NPTS
  IX=(IZ(I).AND.ABSMASK)
  IY=(IZ(I) .AND. ORDMASK)
  IX = LSHIFT(IX+IXPACK) / ISCALE + 1
  IY = LSHIFT(IY+IYPACK) / ISCALE + 1
  PRINT 91. IX. IY
   IF (IX.GT.MAXABS*10) IX=MAXABS*10
   IF (IY.GT.MAXORD) IY=MAXORD
  PACK 10 CHARACTERS PER WORD
   IWORD=(IX=1)/10+1
   ICHAR=IX=(IWORD=1)*10
10 IP(IWORD+IY) = IP(IWORD+IY) +OR+ USHIFT (K+ +6*(128+ICHAR))
  PRINT POINTS
  **** PAGE 1****
  KNDELINE
  PRINT 90.BLURP
  PRINT 91+(I+I=1+KNU)
  PRINT 92
  PRINT 93
   DO 15 J=1.MAXORD
   JJ=MAXORD=J+1
   L10=578
   L9=
          JJ/100
          (JJ+L9*100)/10
   L8=
   L7=338+(JJ=L9*100=L8*10)
   L8 = L8 + 33B
   L9 = L9 + 33B
   IF (L8 .EQ. 33B) L8 = 55B
   IF (L9 EQ_{\bullet} 338) L9 = 558
          (L7+64*(L8+64*(L9+64*L10)))
   L =
   L = LSHIFT (L. = 36)
15 PRINT 94, JJ, (IP(I, JJ), I=1, KND), L
   PRINT93
```

> C C

> с

00000

C

```
HILGARDIA • Vol. 43, No. 5 • June, 1975
   PRINT 92
   PRINT 91.(I.I=1.KND)
   PRINT 90
   RETURN
90 FORMAT (1H1/1X+10A10)
91 FORMAT (5X+12110)
92 FORMAT (5X+8(9X+1H0))
93 FORMAT (4HZ + 82(1H+))
94 FORMAT (1HZ+13+1H++12A10)
   END
  FUNCTION RANDOM(R)
5 R=MOD(IFIX(R*2147483647.0)*455470314,2147483647)/2147483647.
  RANDOMER
  RETURN
  END
           IDENT LSHIFT
             ENTRY LSHIFT
 FUNCTION LSHIFT(L,N)
SHIFT
         BSS
                   1
         SA1
                   81
                                         L INTO X1
         SA2
                   82
                                         N INTO X2
 POSITIVE SHIFT IS HIGHT END-OFF
 NEGATIVE SHIFT IS LEFT CIRCULAR
         SB2
                   X 2
                                         N IN B2
         AX6
                   82+×1
                                         L SHIFTED N#3 BITS INTO X6
         JP
                   LSHIFT
          END
```

To simplify information, trade names of products have been used. No endorsement of named products is intended, nor is criticism implied of similar products which are not mentioned.

31m-6,'75 (S5300L) VL

141

The journal HILGARDIA is published at irregular intervals, in volumes of about 650 to 700 pages. The number of issues per volume varies.

Single copies of any issue may be obtained free, as long as the supply lasts; please request by volume and issue number from:

> PUBLICATIONS UNIVERSITY OF CALIFORNIA DIVISION OF AGRICULTURAL SCIENCES

> > 1422 South 10th Street Richmond, California 94804

