Computers & Geosciences, Vol. 4, pp. 131–141
© Pergamon Press Ltd., 1978. Printed in Great Britain

0098-3004/78/0601-0131/$02.00/0

THE MONTE-CARLO GENERATION OF RANDOM POLYGONS

IAN K. CRAIN
Computer Systems, Planning and Evaluation, Health Protection Branch, Department National Health and Welfare
Ottawa, Ontario, Canada

(*Received* 3 *June* 1977)

Abstract—Random surface-covering aggregates of polygons are of interest as a stochastic model for several geological applications. The approach and algorithms for the generation of random polygons are presented along with a FORTRAN program designed to generate the random Voronoi polygons. To demonstrate the use of the program, the results of the generation of 57,000 Voronoi are tabulated in histogram form, for use in hypothesis testing. The computer program can be adapted easily to other random polygon generations.

Key Words: Algorithm, FORTRAN, Subroutine, Geometry, Simulation, Statistics, General, Geography, Geomorphology.

INTRODUCTION

The statistics of randomly generated surface-covering polygons are of interest in several geological applications. For example, Beard (1959) and Smalley (1966) studied them in relation to columnar jointing, Lachenbruch (1962) examined polygonal structures due to permafrost, Crain (1976) discussed random polygons in the context of geotectonics, and Gray and others (1976) produced results in a more general geological context.

Applications of statistics based on random polygons abound in other sciences as well, particularly geography (Boots, 1973; Boots, 1976), biology (Hamilton, 1971) and metallurgy (Smith, 1951), to cite a few examples.

Of particular interest are the so-called Voronoi polygons, which represent the extension of the Poisson process into two dimensions and thus form a model for various natural processes, such as, columnar jointing, mudcracking, crystallization, etc. These polygons also are referred to as Theissen polygons, Dirichlet polygons, or cellular networks (see, for example, Boots, 1973). This polygonal tessellation can best be thought of by visualizing the uniform growth about uniformly Poisson-random points in the plane. These pseudocrystals grow until mutual contacts (which will be straight lines) prevent further growth. The result is a covering of the plane with a quasi-hexagonal appearance. Figure 1 shows one result of such a process. Each polygon has the property that every point within it is closer to that polygon's nucleus than to any other polygon's nucleus. Thus the sides of the polygons are the right bisectors of lines joining the nuclei.

This paper describes the algorithms necessary for the generation and mensuration of random polygons, with particular reference to the Voronoi polygons, although the techniques and related computer programs could be adapted easily to other particular polygonal tessellations (see, for example, Note 1 on the program listing). The statistical results of the generation of a large number of independent random Voronoi polygons are presented. These results have potential value for hypothesis testing in various applications.

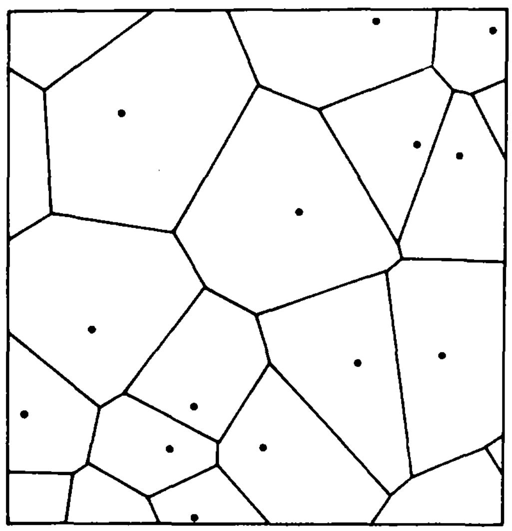

Figure 1. Random Voronoi polygons.

The computer programs were developed initially to count and measure the polygons due to random lines in the plane (Crain and Miles, 1976), the results of which were applied in a geotectonic study (Crain, 1972b). The programs were modified later to generate the Voronoi polygons and preliminary results of the generation of 11,000 polygons were published some years ago (Crain, 1972a). In this paper the results of the generation of 46,000 new polygons have been added, greatly increasing the accuracy of any hypothesis testing based on the resulting histograms.

THEORETICAL ASPECTS

The variables of natural interest in describing a polygonal tessellation are the number of sides, the length of the sides, the perimeter and the area, to be denoted N, L, S, and A, respectively. The length, perimeter, and area distributions are related to the intensity of the Poisson process, ρ, that is, the number of points per unit area used to generate the tessellation. Because the process is ergodic, isotropic, uniform, etc. the distribution of N is independent of ρ.

Theoretical results for the parameters for the Voronoi polygons are relatively scarce. Miles (1970) gives the following:

$$E(N) = 6, E(S) = 4/\rho^{1/2}, E(A) = 1/\rho, E(A^2) = 1.28/\rho^2.$$

Clearly $E(L) = E(S)/6 = 2/3\rho^{1/2}$.

(The expression $E(X)$ is used to denote the expected value (mean) of the variable X).

Notationally it is convenient to drop the ρ from the expressions, so in this paper the following replacements are made:

$$l = 3\rho^{1/2}L/2$$
$$s = \rho^{1/2}S/4$$
$$a = \rho A$$

Thus $E(l) = E(s) = E(a) = 1$.

The individual probabilities of obtaining polygons of 3, 4, 5,... sides will be denoted $p_3, p_4, p_5, \ldots$, where, of course

$$\sum_{i=3}^{\infty} p_i = 1.$$

Theoretical results for the p_i are not known.

ALGORITHM

Concentually the algorithm for the generation and counting of the polygons proceeds in four main phases: generation of the polygons, location of the first vertex, outlining the polygon and, mensuration of the polygon.

Phase I: polygon generation

For the Voronoi polygons this consists of, first, generating a relatively large number of Poisson random points in a finite square, and, second, calculating all the *possible* sides of the Voronoi polygon surrounding the most central point in the region. These consist of the right bisectors of the lines joining the central point to each of the other points. Only the most central polygon is studied, to avoid the inherent "edge-effects" of planar sampling (Miles, 1974).

Associated equations:

The origin is shifted to the most central point and the equation of the right bisector of the line joining the origin to the point (x_i, y_i) can be calculated as:

$$u = -x_i/y_i$$
$$v = (y_i - ux_i)/2 \qquad (1)$$

where the desired equation is $y = ux + v$.

Phase II: location of first vertex

The strategy is to locate the nearest vertex on the nearest line (where "nearest" refers to the distance from the central point). This nearest line is located by a simple scan of all possible sides (lines) as calculated in Phase I. All intersections of this line with the other lines then are calculated and the nearest one of these points is identified. This point is surely one of the vertices of the desired polygon and the two lines which intersect at this point are two sides of the polygon.

Associated equations:

Given two lines defined by linear equations

$$y = u_1x + v_1$$
$$y = u_2x + v_2$$

the point of intersection (x_i, y_i) is given by

$$x_i = (v_1 - v_2)/(u_2 - u_1)$$
$$y_i = (u_2v_1 - u_1v_2)/(u_2 - u_1). \qquad (2)$$

Phase III: outlining the polygon

Having located one vertex of the polygon (and thus two of the bounding lines as well) the procedure is to determine in which direction to proceed along one of these lines in order that the polygon will be traced out in a specified sense (in this situation clockwise, that is, when "walking" around the boundary of the polygon the interior of the polygon will be to the right of the "walker".)

Once the initial direction is established, the nearest intersecting line in that direction is located, and this sequence continued until the polygon is closed. This procedure is elaborated in some detail by Crain and Miles (1976), so that reference should be made to that publication for the subleties involved.

Phase IV: mensuration of the polygon

In this final phase, the number of sides, side lengths, perimeter, and area are calculated. The results of these calculations are accumulated in histograms. The algorithm then reverts to Phase I and the complete cycle is repeated to generate as many independent polygons as desired.

Associated equations:

The length of a side joining the points (x_1, y_1) and (x_2, y_2) can be calculated trivially as

$$L = \sqrt{((x_1 - x_2)^2 + (y_1 - y_2)^2)}. \qquad (3)$$

The area is calculated by subdividing the polygon into triangles and using the equation

$$A = \sqrt{(w(w - r_1)(w - r_2)(w - r_3))} \qquad (4)$$

where r_1, r_2, r_3 are the length of the sides of the triangle and $w = (r_1 + r_2 + r_3)/2$.

The total area of the polygon will be the sum of $N - 2$ of these triangles.

THE COMPUTER PROGRAM

General

The program listed here (Appendix) is designed to run on a CDC CYBER 74 computer. The only sections of the program which are particular to that series of machines are the random number generation functions RANSET, RANF, and RANGET. For these equivalent routines exist on most installations.

The trivial main calling program uses an 'O' (octal) format to read the starting random number. This will require modification to suit the random number generator employed. The I/O conventions are: unit 1 = card reader, unit 3 = printer, and unit 29 = tape drive. Historically, the program was written for an IBM 360 machine, on which the original work was done (Crain, 1972a) and was converted to the CDC machine with no difficulty some years later.

Memory requirements are small (15 K words) and depend mainly on the choice of 35 as the maximum number of points to be generated in one replication.

Input

Three input parameters are required. These are, in order, a starting random number, or "seed", F, the number of random points to be generated in each replication, NPX (35 is recommended), and the number of replications, NTIMES (no limit except for your computer budget). The starting random number is read in by the calling program as a 20-digit octal number, in columns 1 to 20, the number of random points as an integer constant ending in column 25 and the number of replications as an integer constant ending in column 45. (FORMAT (020, 15, 110)).

Output

The primary subroutine VOR2 has minimal outputs. A line containing the starting random number and number of points in the generation region is printed initially. A print line is generated after the completion of each 100 polygons. This is an aid in debugging or in setting time limits for future runs. This is followed by an error indicator which should be zero for a valid run.

The information on the resulting polygons is printed in histogram form by subroutine HIST (See sample output). The count in each window interval is indicated at the far right. The left-hand numbers indicate the lower limit of the interval.

Individual polygon detailed information (vertices) are written on peripheral unit 29 for archival retention and possible later statistical analysis.

Error messages

In theory this program needs no error messages as it cannot fail, *provided* the generation of the random polygon is correct, that is there exists a closed convex polygon surrounding, but not touching, the central point. Error exits are as follows:

Subroutine VOR. A count is taken of all polygons having in excess of 20 sides. For Voronoi polygons these are extremely unlikely, and normally would represent an error condition. This count is printed out as noted previously. A nonzero value would indicate a difficulty in the polygon generation, such as an incorrect random number generator, or memory over-writing etc.

Subroutine NEXTPT. "NO POINT FOUND BY NEXTPT". This indicates the *no* points of intersection exist on this line in the search direction being attempted. This would indicate normally that the polygon is not closed and could be due to incorrect polygon generation, or too few generated points in the region.

Subroutine HIST. An initialization call is necessary to this routine in order to zero out arrays and initialize counters. Failure to do this results in a warning message on attempting to add a value to the histogram.

VOR2—description

This primary subroutine serves to generate the random polygons and do some of the work of delineating the polygon. The rather messy procedure of determining the first vertex and the direction of search for the next vertex is performed here, as in the test for closure of the polygon. Subroutines are called to determine the next vertex (after the first), to calculate points of intersection, perimeters, and areas, and to form histograms.

Subroutines—description

Subroutine INTER calculates the point of intersection of two lines, PERIM and AREA, the perimeter. and area of a polygon respectively. HIST accumulates and prints histograms.

Subroutine NEXTPT performs the function of locating the nearest vertex in a given direction on a given line. This is done by examining all points of intersection in a given direction (previously calculated by INTER and stored in arrays XI and YI) to locate the nearest.

Symbolic names

The arrays XI and YI store the mutual points of intersection of lines. A and B contains the slope and intercepts respectively of the bounding lines (the u_i and v_i of equation 1), XP and YP contain the coordinates of the generating Poisson random points, NPTS the number of points in the region and NTIMES the number of repetitions to be performed. Other symbolic names are used in obvious ways for intermediate storage, and so on.

RESULTS

Table 1 shows the distribution of sides of 57,000 independent random Voronoi polygons. These include the 11,000 previously published (Crain, 1972a, 1972b). Esti-

Table 1. Sample frequencies of sides of random Voronoi polygons

N	Frequency	P_N
3	628	0.0110
4	6145	0.1078
5	14783	0.2594
6	16825	0.2952
7	11306	0.1984
8	5105	0.0896
9	1686	0.0296
10	428	0.00751
11	81	0.00142
12	10	0.000175
13	3	0.000053
Total sample	57,000	

mates of the p_i also are given based on these results. The most likely shape is the hexagon having a probability of occurrence of approximately 0.295. Estimates of the probabilities of the many-sided polygons of course are less accurate. No polygon of 14 sides was detected. The observed mean number of sides of the 57,000 polygons was 5.996, in close agreement with the theoretical value.

Figure 2 shows the histogram of the distribution of the normalized perimeter, s, and Figure 3 the normalized side length l. The additional vertical axes give an approximation of the probability density functions $f(s)$ and $f(l)$. The mode of the s density is near 0.75 as it is for l. The value of $f(l)$ at $l = 0$ is approximately 0.46 and $f(l)$ for small l would seem to have a small positive slope.

Figure 4 shows the histogram of the area of the polygons. The mode is approximately at 1.0, where $f(a) = 0.89$. In applying these histograms in hypothesis testing, the "a" distribution is one of the most useful, because in a naturally occurring polygon, measuring the number of sides and perimeter may be more uncertain than estimating the area, for example, from a vertical photograph. (In Figures 2, 3, 4, only the 46,000 new polygons of this study are included).

Table 2 gives estimates of the moments of N, s, l, and a. The area measurements were used to estimate ρ, so that the first moment of a is automatically exactly 1.0. The table includes only the recent work as higher order moments were not calculated in the preliminary study.

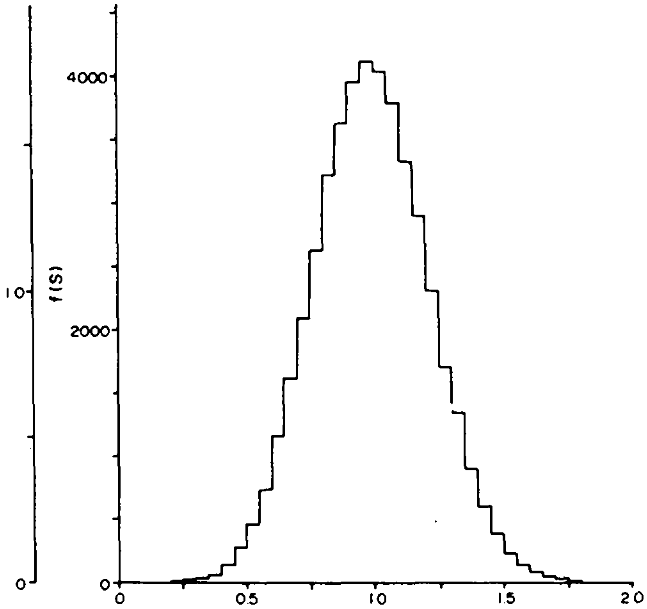

Figure 2. Histogram of frequency of perimeters, s, of random Voronoi polygons. Sample size 46,000. Additional vertical scale shows estimate of probability-density function.

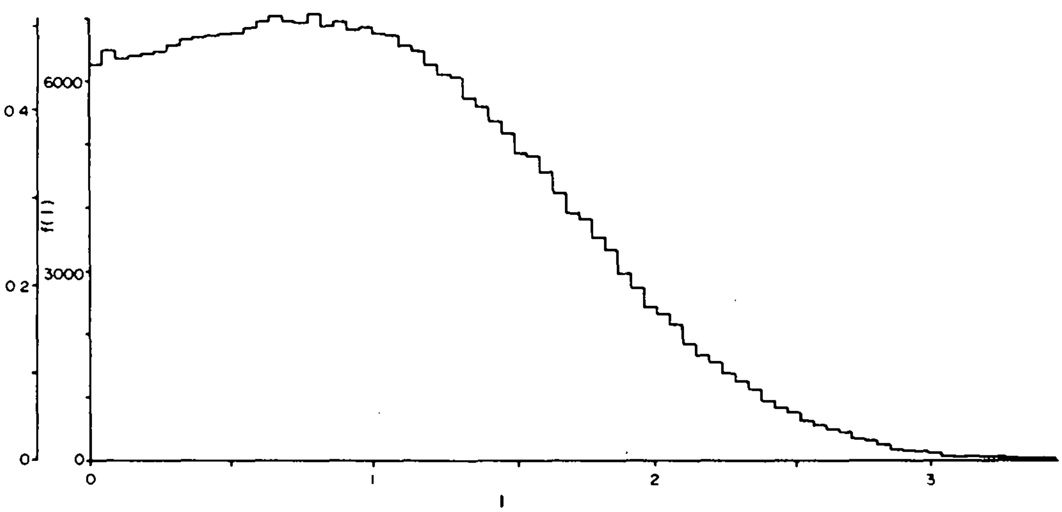

Figure 3. Histogram of frequency of side length, l, of random Voronoi polygons. Sample size 46,000. Additional vertical scale shows estimate of probability-density function.

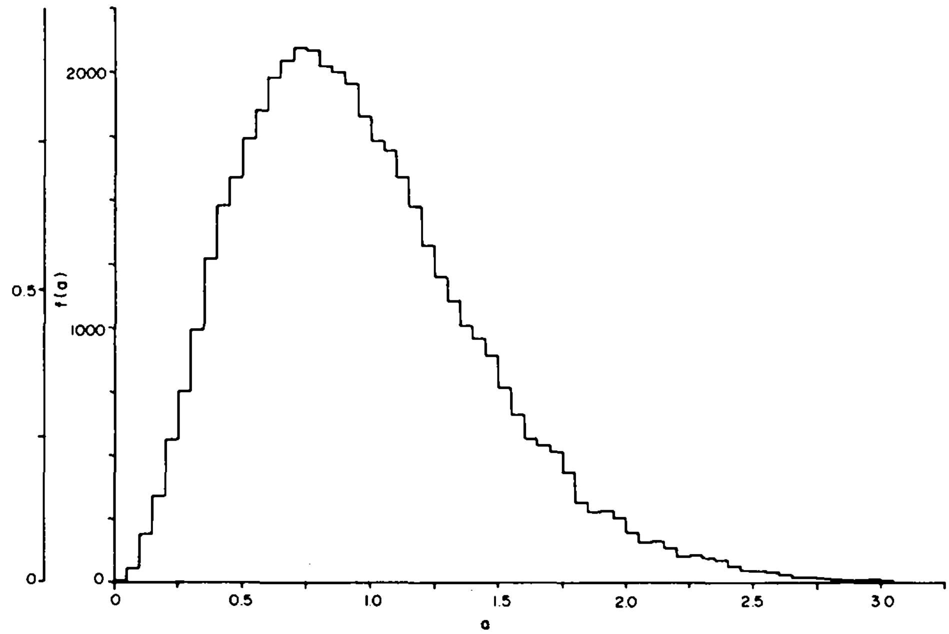

Figure 4. Histogram of frequency of area, a, of random Voronoi polygons. Sample size 46,000. Additional vertical scale shows estimate of probability-density function.

Table 2. Sample estimates of moments of N, s, l, a

Moment	N	s	l	a
1	6.007	1.004	0.988	1.000
2	37.87	1.060	1.525	1.240
3	249.9	1.166	2.800	1.817
4	1722	1.335	5.777	3.026
5	12360	1.581	13.03	5.832
6	92260	1.936	31.60	12.45
Sample size 46,000				

Correlation coefficients are as follows:

$$(N, s) = 0.5341, (N, a) = 0.6051, (s, a) = 0.9501.$$

Note how highly correlated are the perimeter and area, showing that the polygon shapes tend to be equant or quasicircular rather than elongated.

Of some interest is the manner in which the polygons change "shape" as the number of sides varies. Table 3 gives conditional first and second sample moments of s and a, given N and respective correlation coefficients.

Table 3. Conditional moments and correlation coefficients

N	$\bar{s}$	$\overline{s^2}$	$\bar{a}$	$\overline{a^2}$	$r(s, a\|N)$
3	0.707	0.360	0.538	0.169	0.9549
4	0.815	0.567	0.702	0.401	0.9369
5	0.915	0.773	0.873	0.713	0.9422
6	1.010	0.994	1.056	1.138	0.9476
7	1.094	1.214	1.235	1.667	0.9539
8	1.182	1.447	1.431	2.321	0.9502
9	1.267	1.710	1.641	3.184	0.9578
10	1.310	1.850	1.744	3.639	0.9589
Sample size 25,000					

This table is based on a sample of 25,000 polygons which the author was able to archive on tape for further study. Because of the conditions under which some of the computer time was available (see acknowledgments) it was possible to keep stored permanently output on only about half of the runs.

As with the Poisson polygons (Crain and Miles, 1976) the correlation coefficients increase with increasing N, indicating an increase in "circularity" of the polygons. The high value of $r(a, s)|N = 3$ may not be significant as the numbers of triangles in this sample are rather low. Further results would be needed to clarify the significance of this value.

Acknowledgments—The original seeds of this project go back a number of years, and much thanks is due to Dr. Roger Miles for interesting me in this field of study, and for helpful advice. I was particularly fortunate to have been able to perform a number of computer runs free of charge as a test program while the Energy, Mines and Resources Computer Centre was debugging a new operating system. For this I am grateful to Mr. Robert Hardin. Several scientific users of the same computer centre have contributed computer time as well, in end-of-fiscal year generosity, which assisted in the production of the results shown here. Particular thanks go to Ms. G. M. Martin for assistance with the algorithm, program conversion, and encouragement to write up this work finally in publishable form.

REFERENCES

Beard, C. N. 1959, Quantitative study of columnar jointing: Geol. Soc. America Bull. v. 70, no. 3, p. 379–382.

Boots, B. N., 1973, Some models of the random subdivision of space: Geografiska Annaler, Ser. B., v. 55, no. 1, p. 34–48.

Boots, B. N., 1976, A bibliography of theoretical and empirical studies of cellular networks: Council of Planning Libraries, Exchange Bibliographies, no. 970, 14 p.

Crain, I. K., 1972a, Monte-Carlo simulation of the random Voronoi polygons—preliminary results: Search, v. 3, no. 5, p. 220.

Crain, I. K., 1972b, Statistical approach to the analysis of geotectonic elements: unpubl. doctoral dissertation, Australian National University, 130 p.

Crain, I. K., 1976, Statistical analysis of geotectonics, *in* Random processes in geology; Springer-Verlag, New York, p. 3–15.

Crain, I. K., and Miles, R. E., 1976, Monte-Carlo estimates of the distributions of the random polygons determined by random lines in a plane: Jour. Stat. Computer Simulation, v. 4, no. 2, p. 293–325.

Gray, N. H., Anderson, A. B., Devine, J. C., and Kwasnik, J. M., 1976, Topological properties of random crack networks: Jour. Math. Geology, v. 8, no. 6, p. 617–626.

Hamilton, W. D., 1971, Geometry for the selfish herd: Jour. Theoretical Biol., v. 31, no. 2, p. 295–311.

Lachenbruch, A. H., 1962, Mechanics of thermal contraction cracks and ice-wedge polygons in permafrost: Geol. Soc. America Sp. Paper 70, 69 p.

Miles, R. E., 1970, On the homogeneous planar Poisson point process: Math. Biosciences, v. 6, no. 1, p. 85–127.

Miles, R. E., 1974, On the elimination of edge-effects in planar sampling, *in* Stochastic geometry: John Wiley & Sons, London, p. 228–247.

Smalley, I. J., 1966, Contraction crack networks in basalt flows: Geol. Mag., v. 103, no. 2, p. 110–114.

Smith, C. S., 1951, Grain shapes and other metallurgical applications of topology: Trans. Am. Soc. Metals, Preprint no. 37, 42 p.

APPENDIX

Program Listing

```
      PROGRAM VORM(INPUT,OUTPUT,TAPE1=INPUT,TAPE3=OUTPUT,TAPE29)
C *****
C *****  CALLING PROGRAM FOR VOR2
C *****
      READ(1,1)F,NPX,NTIMES
    1 FORMAT(O20,I5,I10)
      CALL VOR2(F,NPX,NTIMES)
      CALL EXIT
      END
```

```
C ***** SUBROUTINE TO GENERATE RANDOM VORONOI POLYGONS
C *****
C ***** IAN K. CRAIN
C ***** PLANNING AND EVALUATION DIRECTORATE
C ***** HEALTH PROTECTION BRANCH
C ***** DEPARTMENT OF NATIONAL HEALTH AND WELFARE
C ***** TUNNEY'S PASTURE, OTTAWA CANADA
C *****
C ***** PROGRAM REVISED SEPT, 1977.
C *****
      SUBROUTINE VOR2(F,NPX,NTIMES)
      COMMON /STOR/ XI(35,35),YI(35,35),A(35),B(35),XG(35),YG(35),
     1 NPTS
      DIMENSION XP(35),YP(35),SIDES(35)
C *****
C ***** INITIALIZATION
C *****
      NPTS=NPX
      CALL RANSET(F)
      RHO=NPTS/4.
      SQR=SQRT(RHO)
      W=0.
      WX=20.
      WW=9.9
      WWW=0.1
      CALL HIST(1,W,WX,1.,W,1,1)
      CALL HIST(2,W,4.9,0.05,W,1,1)
      CALL HIST(3,W,4.9,0.05,W,1,1)
      CALL HIST(4,W,4.9,0.05,W,1,1)
      WRITE (3,100) F,NPTS
  100 FORMAT(1H ,O20,I5)
      NIT=0
      NX=0
C *****
C ***** REPEAT LOOP FOR NUMBER OF REPLICATIONS DESIRED
C *****
      DO 7000 IT=1,NTIMES
      NIT=NIT+1
      IF(NIT-100)2356,2357,2357
 2357 WRITE(3,2358)IT
      NIT=0
 2356 CONTINUE
 2358 FORMAT(1H ,I6,20H POLYGONS GENERATED )
C *****
C ***** GENERATE LINES WHICH BOUND RANDOM POLYGONS
C *****
```

```
C      ***** THIS SECTION CAN BE MODIFIED TO GENERATE
C      ***** OTHER THAN VORONOI POLYGONS
          DO 50 I=1,NPTS
          F=RANF(DUM)
          XP(I)=2.*(F-.5)
          F=RANF(DUM)
       50 YP(I)=2.*(F-.5)
          DO 51 I=1,NPTS
          A(I)=(XP(I)/YP(I))*(-1.)
       51 B(I)=YP(I)/2.-A(I)*XP(I)/2.
C     *****
C     ***** END OF POLYGON GENERATION
C     *****
C
C     *****
C     ***** FIND SIDES AND VERTICES OF POLYGON
C     *****
C     ********** CALCULATE INTERSECTION POINTS OF ALL LINES
          DO 52 I=1,NPTS
          L=I
          DO 53 J=L,NPTS
          IF(I-J)55,53,55
       55 CALL INTER(I,J,XXX,YYY)
          XI(I,J)=XXX
          YI(I,J)=YYY
       53 CONTINUE
       52 CONTINUE
          DO 62 J=1,NPTS
          L=J
          DO 63 I=L,NPTS
          IF(I-J)65,63,65
       65 XI(I,J)=XI(J,I)
          YI(I,J)=YI(J,I)
       63 CONTINUE
          XI(J,J)=100.
       62 YI(J,J)=100.
C     ********** FIND MOST CENTRAL GENERATING POINT
          DMIN=100.
          DO 71 I=1,NPTS
          DIST=XP(I)**2+YP(I)**2
          IF(DIST-DMIN)73,71,71
       73 DMIN=DIST
          IMIN=I
       71 CONTINUE
C     ********** FIND NEAREST POINT ON NEAREST LINE
C     ********** (THIS WILL BE THE FIRST VERTEX)
          DMIN=100.
          DO 72 I=1,NPTS
          IF(I-IMIN)170,72,170
      170 DIST=XI(IMIN,I)**2+YI(IMIN,I)**2
          IF(DIST-DMIN)171,72,72
      171 DMIN=DIST
          JMIN=I
       72 CONTINUE
C     ********** FIND DIRECTION OF SEARCH TO LOCATE SECOND VERTEX
          XG(1)=XI(IMIN,JMIN)
          YG(1)=YI(IMIN,JMIN)
      103 FORMAT(1H ,2F10.8,2I4)
          I1=IMIN
          I2=JMIN
          V1=XP(I1)
          V2=YP(I1)
          W1=XP(I2)
          W2=YP(I2)
          CROS=V1*W2-V2*W1
          IF(CROS)80,80,81
       80 IL=I2
          GO TO 82
       81 IL=I1
       82 IF(B(IL))91,91,90
       91 IF(A(IL))92,92,93
       92 IUP=1
          GO TO 96
       93 IUP=2
          GO TO 96
       90 IF(A(IL))94,94,95
       94 IUP=2
          GO TO 96
       95 IUP=1
       96 CONTINUE
```

```
      AA=XG(1)
      BB=YG(1)
C **********  FIND SECOND VERTEX
      CALL NEXTPT(IL,IUP,AA,BB,AAA,BBB,ILX)
C **********  FIND DIRECTION OF SEARCH TO LOCATE NEXT VERTEX
      XG(2)=AAA
      YG(2)=BBB
      N=2
  200 IL=ILX
      N=N+1
      IF(N-20)1220,1220,1221
 1221 NX=NX+1
      GO TO 7000
 1220 IF(B(IL))191,191,190
  191 IF(A(IL))192,192,193
  192 IUP=1
      GO TO 196
  193 IUP=2
      GO TO 196
  190 IF(A(IL))194,194,195
  194 IUP=2
      GO TO 196
  195 IUP=1
  196 CONTINUE
      NM=N-1
      AA=XG(NM)
      BB=YG(NM)
C **********  FIND NEXT VERTEX
      CALL NEXTPT(IL,IUP,AA,BB,AAA,BBB,ILX)
      XG(N)=AAA
      YG(N)=BBB
C **********  IS POLYGON CLOSED?
      IF(XG(N)-XG(1))200,201,200
  201 IF(YG(N)-YG(1))200,202,200
  202 CONTINUE
C **********POLY IS CLOSED.. ADD TO HISTOGRAM, STORE VERTICES ON TAPE,
C **********  RETURN TO REPEAT FOR NEXT POLYGON
      N=N-1
      CALL PERIM(N,S,SIDES)
      CALL AREA(N,AB)
C **********  FACTORS RHO AND SQR NORMALIZE AREA AND PERIM TO MEAN 1.
C **********  IF OTHER THAN VORONOI POLYS ARE BEING GENERATED, THEN
C **********  THESE FACTORS WILL HAVE TO BE CHANGED ACCORDINGLY.
      AA=AB*RHO
      SS=S*0.25*SQR
      FN=N
      CALL HIST(1,GW,GW,GW,FN,1,2)
       CALL HIST(2,GW,GW,GW,SS,1,2)
      CALL HIST(3,GW,GW,GW,AA,1,2)
      DO 772 KW=1,N
      AL=SIDES(KW)*SQR*1.5
      CALL HIST(4,GW,GW,GW,AL,1,2)
  772 CONTINUE
      WRITE(29) N,SS,AA,(XG(I),YG(I),I=1,N)
C *****
C *****  END OF POLYGON DELINATION LOOP
C *****
 7000 CONTINUE
      WRITE(3,2019)NX
C *****
C *****  PRINT OUT HISTOGRAMS OF RESULTS
C *****
      DO 774 KW=1,4
      WRITE(3,775)
  775 FORMAT(1H1)
      CALL HIST(KW,W,W,W,W,1,3)
  774 CONTINUE
 2019 FORMAT(1H ,I6)
      CALL RANGET(GG)
      WRITE(3,99) GG
   99 FORMAT(19H LAST RANDOM NUMBER,O20)
      RETURN
      END

      SUBROUTINE INTER(L,M,X,Y)
C *****
C *****  SUBROUTINE TO CALCULATE POINT OF INTERSECTION X,Y OF
C *****   LINES L,M
C *****
```

```
      COMMON /STOR/ XI(35,35),YI(35,35),A(35),B(35),XG(35),YG(35),
     1 NPTS
      IF(L-M) 371,371,372
  372 I=M
      J=L
      GO TO 373
  371 I=L
      J=M
  373 CONTINUE
      A1=A(I)
      A2=A(J)
      B1=B(I)
      B2=B(J)
      X =(B1-B2)/(A2-A1)
      Y =(A2*B1-A1*B2)/(A2-A1)
      RETURN
      END
```

```
      SUBROUTINE HIST(L,VLOW,VHIGH,VINT,V,IW,ISW)
C *****
C ***** SUBROUTINE TO FORM AND PRINT HISTOGRAMS
C *****
      DIMENSION IHIST(6,100),IROW(100),VL(6),VH(5),VI(6),NV(6),INIT(6)
      DATA INIT/0,0,0,0,0,0/
      DATA IAST,IBLK/1H*,1H /
      GO TO (1,2,3),ISW
C ********** INITIALIZATION
C ********** L      NO. OF HISTOGRAM TO BE INITIALIZED (MAX 6)
    1 DO 30 I=1,100
C ********** VHIGH UPPER LIMIT FOR HISTOGRAM
C ********** VINT  HISTOGRAM WINDOW INTERVAL
      IROW(I)=IBLK
   30 IHIST(L,I)=0
      VL(L)=VLOW
      VH(L)=VHIGH
      VI(L)=VINT
      NV(L)=((VHIGH-VLOW)/VINT)+2.0001
      IF(NV(L)-100)60,60,61
   61 NV(L)=100
   60 INIT(L)=999
      RETURN
C **********     .. ADDS A VALUE TO A HISTOGRAM
C ********** L     NO. OF HIST TO WHICH VALUE IS TO BE ADDED
C ********** IW    NUMBER OF COUNTS OF THIS VALUE
    2 CONTINUE
      IF(INIT(L)-999)63,62,63
   63 WRITE(3,400)L
  400 FORMAT(1H ,9HHISTOGRAM,I4,24HHAS NOT BEEN INITIALIZED)
      RETURN
   62 NVV=NV(L)
      VLW=VL(L)
      VHI=VH(L)
      VIN=VI(L)
      IF(V-VHI)40,40,41
   41 IHIST(L,NVV)=IHIST(L,NVV)+IW
      RETURN
   40 IF(V-VLW)43,43,42
   43 IHIST(L,1)=IHIST(L,1)+IW
      RETURN
   42 VV=(V-VLW)/VIN
      IV=VV+2.
      IHIST(L,IV)=IHIST(L,IV)+IW
      RETURN
C **********     .. PRINTS A HISTOGRAM
C ********** L  NO. OF HISTOGRAM TO BE PRINTED
    3 CONTINUE
      MAX=0
      ITOT=0
      NVV=NV(L)
      DO 44 I=1,NVV
      IF(IHIST(L,I)-MAX)44,44,100
  100 MAX=IHIST(L,I)
   44 CONTINUE
      IF(MAX-100)33,33,32
   32 FMAX=MAX
      SCALE=FMAX/100.
      GO TO 34
   33 SCALE=1.
   34 VLIM=0.
```

```
      VIN=VI(L)
      VLM=VL(L)
      ILEN=IHIST(L,1)/SCALE
      IF(ILEN)79,79,78
   78 DO 51 J=1,ILEN
   51 IROW(J)=IAST
   79 WRITE(3,101)VLIM,IROW,IHIST(L,1)
  101 FORMAT(1H ,F9.4,3X,100A1,1X,I4)
      ITOT=ITOT+IHIST(L,1)
      VLIM= VLM-VIN
      DO 70 I=2,NVV
      VLIM=VLIM+VIN
      DO 36 K=1,100
   36 IROW(K)=IBLK
      ILEN=IHIST(L,I)/SCALE
      ITOT=ITOT+IHIST(L,I)
      IF(ILEN)70,70,77
   77 DO 47 J=1,ILEN
   47 IROW(J)=IAST
   70 WRITE(3,101) VLIM,IROW,IHIST(L,I)
      WRITE(3,106) SCALE,ITOT
  106 FORMAT(1H0,12HSCALE FACTOR,F10.5/1H0,12HTOTAL SAMPLE,I6)
      DO 48 I=1,100
   48 IROW(I)=IBLK
      RETURN
      END

      SUBROUTINE PERIM(N,S,SID)
C     *****
C     ***** FINDS PERIMETER OF POLYGON
C     *****
      COMMON /STOR/ XI(35,35),YI(35,35),A(35),B(35),XG(35),YG(35),
     1 NPTS
      DIMENSION SID(21)
      S=0.
      NM=N-1
      DO 66 I=1,NM
      SID(I)=SQRT((XG(I)-XG(I+1))**2+(YG(I)-YG(I+1))**2)
   66 S=S+SID(I)
      SID(N)=SQRT((XG(1)-XG(N))**2+(YG(1)-YG(N))**2)
      S=S+SID(N)
      RETURN
      END

      SUBROUTINE NEXTPT(IL,IUP,X,Y,XX,YY,ILX)
C     *****
C     ***** FINDS NEXT VERTEX OF POLYGON  GIVEN:
C     *****    IL   CURRENT LINE TO BE SEARCHED
C     *****    IUP  SENSE OF SEARCH
C     *****    X,Y  COORDINATES OF LAST VERTEX
C     *****
      COMMON /STOR/ XI(35,35),YI(35,35),A(35),B(35),XG(35),YG(35),
     1 NPTS
      GO TO (1,2),IUP
C     ********** SEARCH UP LINE TO FIND NEAREST INTERSECTING LINE
    1 YMIN=1000.
      DO 60 I=1,NPTS
      IF(I-IL)61,60,61
   61 IF(YI(I,IL)-Y)60,60,62
   62 D=YI(I,IL)-Y
      IF(YMIN-D)60,60,65
   65 YMIN=D
      ILX=I
      IK=1
   60 CONTINUE
      IF(IK-1)101,200,101
  200 XX=XI(ILX,IL)
      YY=YI(ILX,IL)
      RETURN
C     ********** SEARCH DOWN LINE TO FIND NEAREST INTERSECTING LINE
    2 YMIN=1000.
      DO 70 I=1,NPTS
      IF(I-IL)71,70,71
   71 IF(YI(I,IL)-Y)72,70,70
   72 D=Y-YI(I,IL)
      IF(YMIN-D)70,70,75
   75 YMIN=D
      ILX=I
```

```
      IK=1
   70 CONTINUE
      XX=XI(ILX,IL)
      YY=YI(ILX,IL)
  700 IF(IK-1)101,100,101
  101 WRITE(3,202)
  202 FORMAT(25H NO POINT FOUND BY NEXTPT)
  100 RETURN
      END

      SUBROUTINE AREA(N,AA)
C *****
C ***** SUBROUTINE TO CALCULATE AREA OF A CONVEX POLYGON
C *****
      COMMON /STOR/ XI(35,35),YI(35,35),A(35),B(35),XG(35),YG(35),
     1 NPTS
      AA=0.
      NT=N-2
      X1=XG(1)
      Y1=YG(1)
      DO 66 I=1,NT
      X2=XG(I+1)
      Y2=YG(I+1)
      X3=XG(I+2)
      Y3=YG(I+2)
      E=SQRT((X2-X3)**2+(Y2-Y3)**2)
      D=SQRT((X1-X3)**2+(Y1-Y3)**2)
      C=SQRT((X1-X2)**2+(Y1-Y2)**2)
      S=0.5*(C+D+E)
      AS=S*(S-C)*(S-D)*(S-E)
      IF(AS)790,790,791
  790 AS=0.
      GO TO 66
  791 AS=SQRT(AS)
   66 AA=AA+AS
      RETURN
      END
```