United States
Department of
Agriculture

Forest Service

**Pacific Northwest
Forest and Range
Experiment Station**

**Administrative
Report
March 1982**

Computer Programs for Simulating the Line Intersect Process for Residue Inventory

S.G. Pickford, J.W. Hazard, and Jill Hoopes

Abstract

This paper describes the concepts and operations of two programs, SLASH, which simulates forest-residue populations, and INTRSCT, which performs line intersect residue inventories on these populations. Program SLASH creates residue pieces on a 5.07-acre square area to specified orientation and spatial distributions. The user can specify constant geometric piece shapes or create populations with length/diameter distributions based on actual residue inventories. Program INTRSCT samples this population, using a user-determined number and configuration of sample legs per transect and transects per experiment. The results of these simulations may be used to plan residue inventories and perform technique studies to determine optimum sample designs. Edge effects, boundary problems, and program calibration are discussed.

Keywords: Residue surveys, sampling design, population sampling, computer programs/programing.

Introduction

We recently reported the results of a study that examined the statistical properties of the line intersect method of forest-residue inventory (Pickford and Hazard 1978). The study used computer simulation rather than field trials because the cost of measuring actual residue populations is prohibitively large. This paper presents and describes the computer routines developed for the study; it provides a reference for the above-mentioned research and for applications by others to expand the research in this field.

The concept of the simulation process is straightforward. A simulated population of residue elements of known characteristics is created by specifying the midpoint coordinates, length, diameter, and orientation of each element. Then, the sampling process is simulated by defining the two ends of a randomly oriented line segment, called a leg, searching for population elements that intersect that leg, and accumulating the information of interest concerning the intersection (e.g., diameter, squared diameter, end diameters, and element count). Figure 1 presents the flow diagrams of the logic for the two programs, SLASH and INTRSCT, that accomplish these tasks. A FORTRAN listing of each program is given in the Appendix.

S.G. Pickford is associate professor, College of Forest Resources, University of Washington, Seattle, Washington 98195; J.W. Hazard is Station statistician, and Jill Hoopes was programer/analyst, Pacific Northwest Forest and Range Experiment Station, 809 N.E. Sixth Avenue, Portland, Oregon 97232.

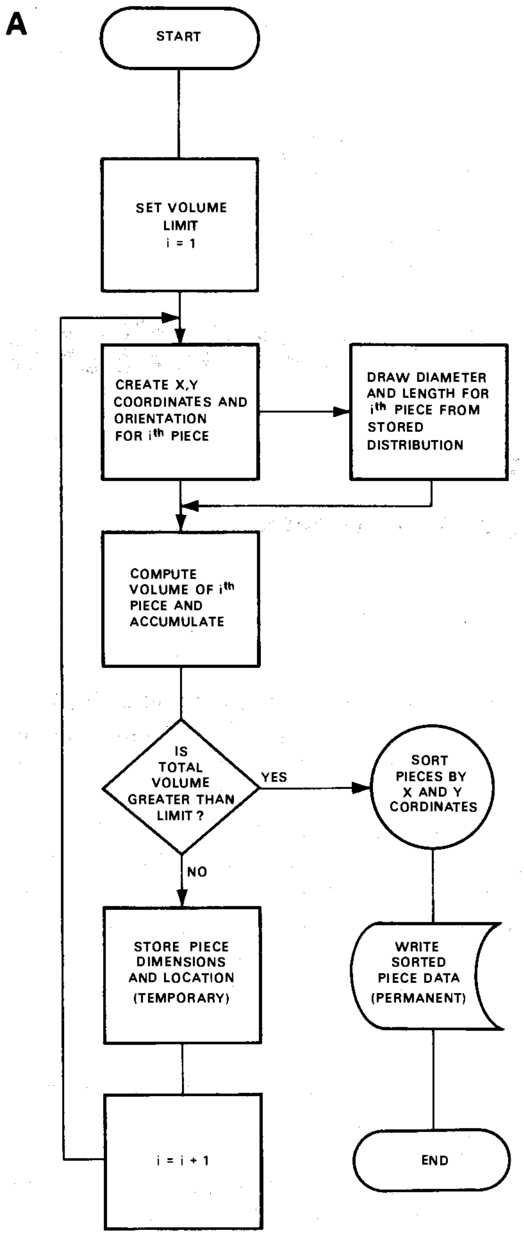


Figure 1a.—The logic of program SLASH.

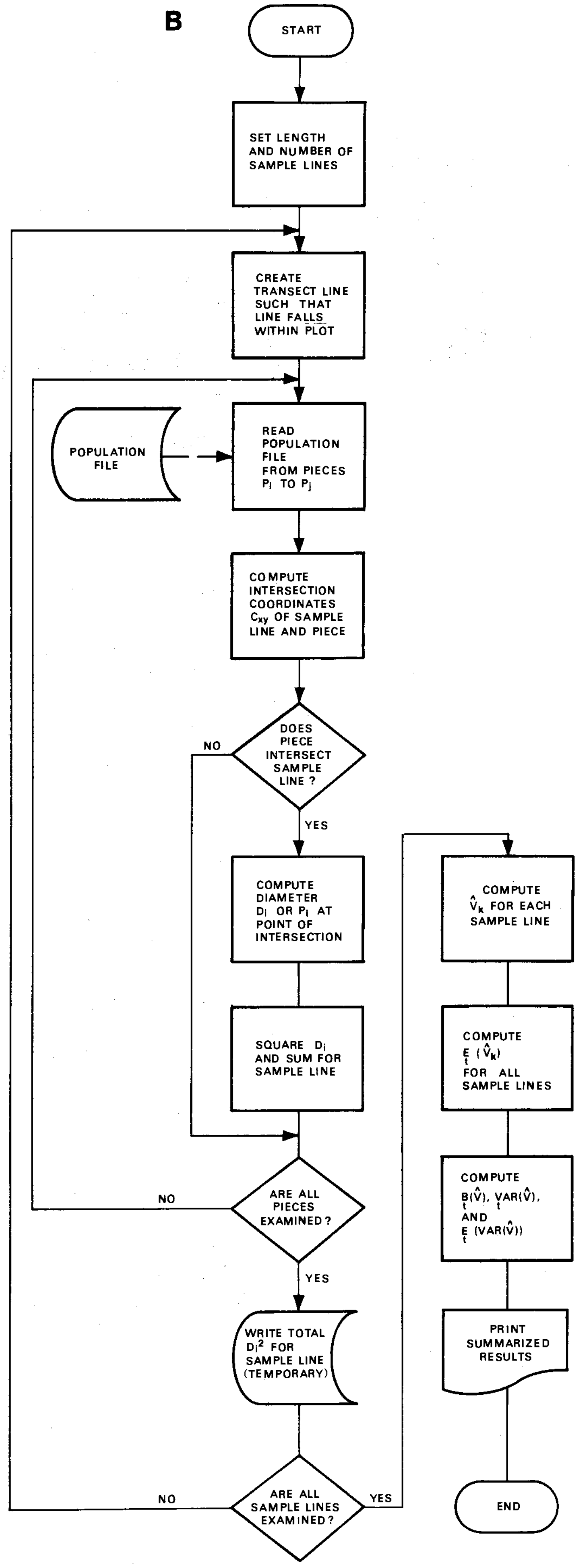


Figure 1b.—The logic of program INTRSCT.

Description of the Program

Program SLASH

The program that creates the population of residue elements for sampling is called SLASH. SLASH creates elements on an arbitrary plot so that they can be readily sampled by simulating the line intersect inventory process. As presently configured, the sample plot is a 470 feet square (i.e., 5.07 acres). Although the size could be changed, it should be done with care. The plot dimensions are entered in the program as numeric constants, and certain adjustments of piece orientation are made near plot boundaries. These constraints (10.0, 460.0, 470.0) are encountered frequently in conditional statements in the main program and several subroutines.

The process of creating a population for sampling could be done on an area of irregular or nonrectangular shape, but the changes must be made carefully because the logic in the current version of SLASH presupposes a square plot.

Location and orientation of pieces. — Midpoint coordinates and axial orientation of individual pieces are computed in subroutine COORD, using a random number generator to create three stochastic variables, t_1, t_2, and t_3. Coordinates of piece midpoints are them computed by

$$x_i = 470\ t_1 \qquad (1)$$

$$y_i = 470\ t_2 \qquad (2)$$

Azimuth direction (θ) for each piece axis is generated by using t_3 by

$$\theta = 2\pi t_3 \qquad (3)$$

Equation (3) produces an axial piece orientation such that $0 < \theta \leq 2\pi$. If the pieces are cylinders, it is only necessary that $0 < \theta \leq \pi$. If tapered pieces are used, however, then random orientation requires that $0 < \theta \leq 2\pi$, and the two ends of each piece must remain distinct and identifiable. We adopted the convention that the large end of a piece lies in the θ direction from point ($x_i y_i$).

Program SLASH can currently generate only uniform random distributions of piece midpoints over the plot, and a uniform random distribution of piece axial orientation. Real-world residue populations often appear strongly oriented with respect to piece axes (e.g., cable yarding systems on steep ground, directional felling), or show positional bias in one or both horizontal directions (e.g., cable roads, skid trails), or show both biases simultaneously. Although the program can not create populations with these types of bias, it would only require generating a random stochastic variable $\underline{t}'$ with an appropriate distribution for use in subrouting COORD (refer to any standard text on computer simulation).

Any time that a population of linear elements is created inside a fixed boundary, the creation process must be adjusted in the vicinity of that boundary. If not given proper attention, these adjustments may cause unexpected anomalies in the results. We adjusted piece distribution and orientation so that the results approximate the real-world edge effects on logging units. We discuss the actual consequences of these adjustments in the following section on performance of the line intersect method. The adjustments to piece orientation and distribution occur in subroutines LID, SIDE, and CORNER.

If a piece midpoint is generated at a distance less than a half piece length from the boundary, random orientation of the piece axis (θ) might result in part of the piece length lying outside the sample plot. Because of the way in which we controlled total piece volume on the plot, each piece created had to lie entirely within the plot. To ensure that a piece close to the boundary would indeed lie wholly within the plot, the range of values of θ over which axial orientation of that piece was permitted to vary was restricted to values that excluded intersection with the boundary. The result of this particular logic is to create a band near the plot boundaries where pieces are more oriented in the direction parallel to the boundary. This occurs in real populations where a tractor-built fireline around the unit pushes pieces back into the unit and when snags are felled back into the unit.

Dimensions of pieces. — SLASH randomly draws two numbers representing length and diameter for each piece created. These numbers are drawn from a cumulative distribution of piece lengths and diameters that operates as a look-up table. The distribution is supplied by the user and is read in as cards from subroutine DISTRB (array variables LEN, DIA). The user must supply limits for each length and diameter class (array variable LIM). The combination of subroutines PIECE, DISTRB, LIMIT, and SIZER allows considerable flexibility in creating populations based on residue inventory data. We derived our size distributions from unpublished residue inventory data collected for other research purposes. Similar data published by Howard (1973), however, are adequate for use in this program. For example, to convert Howard's table 2, the volumes in each cell must be divided by the total volume. This gives the cumulative frequency distribution of pieces by length and diameter size classes.

PIECE, DISTRB, LIMIT, and SIZER subroutines can be removed if a simple population of uniform cylinders or other shape is to be created. In this event, statement 20 in the main program should be replaced with appropriate coding to generate the desired piece length and diameter. Alternatively, these can be assumed at the outset, and the variable VOL can be incremented by a constant each time through the loop, or dispensed with altogether and the volume controlled by the total number of pieces generated.

Density of pieces. — One major advantage of simulating residue populations and inventories is that the true population volume on the sample area is much easier to determine than in studies using real residues on actual logging units. When empirically derived distributions of piece length and diameter such as described above are used, the total volume on the sample plot is controlled by limiting the number of pieces (to 1150, in the version of SLASH discussed here). The plot volume can be estimated by computing the average piece dimensions and then computing the expected total volume. Exact volume on the plot can be obtained by computing volumes of individual pieces from piece length and diameter, and summing for all pieces on the plot. This results in a known density of pieces but a total volume that varies somewhat about the expected value.

Variation in total plot volume about a desired limit can be reduced (but not eliminated) by setting a volume limit and testing it against the variable SVOL, which accumulates total volume on the plot as each piece is created (main program). The fact that the population volume deviates slightly from an expected value does not create a problem because the simulation process is estimating the realized population values.

Program INTRSCT

Once SLASH has created a file containing the location, length, diameter, and orientation of each piece in the population, the file is used as input for INTRSCT, the line intersect inventory simulator.

INTRSCT provides the logic for drawing random samples of (x,y) point pairs within a two-dimensional grid indexing the plot. The points locate a randomly oriented line on the population. We call this line a sample "leg." The length and number of such legs that make up one complete sample are fixed by the user. A simulation experiment then consists of a fixed number of repeated samples (transects). The current configuration of INTRSCT generates only randomly located sample legs. We are currently modifying INTRSCT to sample residue populations systematically as well.

Generation of sample legs. — A sample leg is generated by choosing the (x,y) coordinates of one end of the leg. A random number generator is used in the same fashion as in program SLASH; i.e., the starting points of sample legs are randomly located within the sample area.

Orientation of the sample leg is also randomly chosen, but the leg length is set by the user as an input variable (TLEG). The endpoint coordinates of the sample leg are computed from the starting coordinates, leg length, and leg orientation.

If an endpoint of a sample leg falls outside the sample area, the portion of the leg that extends outside the leg is reflected back from the boundary at 180 degrees minus the incident angle.

This preserves the randomness of the distribution of the sample leg because the distribution of transect starting points is not affected; the orientation of the reflected length is random because it is determined by the randomly chosen orientation of the incident length; and the expected total length of line in the vicinity of the boundary using reflection is the same as if the boundary did not exist. Sample legs generated beyond the boundary were permitted to enter the sample area from the outside. Thus, little or no "edge effect" is created by reflection.

The piece-sampling process. — As each leg is created, it is searched along its length for intersections with pieces in the sample population. The search is conducted according to a screening procedure that progressively reduces the number of pieces to be examined. The population of pieces created by SLASH was previously ordered on the (x,y) midpoint coordinates. Then the search area is established by a binary search routine (subroutine XRANGE), which identifies that portion of the sample population where intersections of pieces with the transect leg are possible. This area is a rectangle whose diagonal is the line segment formed by adding half the length of the longest piece in the population to the (x,y) coordinates of the endpoints of the transect leg. Only those pieces whose midpoint (x,y) coordinates fall within this rectangle are examined further.

Within the search area, the (x,y) coordinates of the intersection of a piece with the transect leg are determined by solving the two simultaneous linear equations in two unknowns that define, respectively, the sample leg and the piece axis. If a unique solution exists and if the intersection lies in the search area, then the final step is to determine whether the intersection falls within the length of the piece. An intersection is valid if the distance from the piece midpoint to the intersection does not exceed half the length of the piece. When the leg intersects the piece axis at its endpoint, only alternate intersections are accepted.

Once a valid intersection occurs, the piece diameter at the point of intersection is determined (subroutine SAMPL). If the pieces are cylinders, the program uses the input piece diameter. If pieces are tapered, the intersected diameter is computed by correcting the closer end diameter for piece taper (inches per foot) times the distance from the close end to the intersection. The diameter at the intersection is squared and accumulated for each transect leg. When the last piece in the search area has been tested for a valid intersection, the sum of the squared diameters is written out on a disk file for later use in computing estimated volume and summary statistics.

Statistics and estimates. — After all transect legs have been created and searched, the disk file is used by subroutine SUMMARY to compute estimated volume per acre and associated statistics. Because the number of legs per transect (LSZ) is constant throughout each experiment, the LSZ data points on the disk file constitute the data for each transect.

The summary statistics defined by Pickford and Hazard (1979) are symbolized in INTRSCT as follows:

$$\mathrm{YJK} = \frac{373.1977\ \mathrm{DSUM}}{\mathrm{TLEG}}$$

where DSUM is the accumulated, squared, intersected piece diameter in square inches, and TLEG is the transect leg length in feet.

YJK is the estimated volume per acre for a single leg of one particular transect, and the estimated volume for the j^{th} transect is

$$\mathrm{YK}_j = \frac{\sum_{i=1}^{\mathrm{LSZ}} \mathrm{YJK}_i}{\mathrm{LSZ}}$$

where LSZ is the number of legs per transect. Then, for all transects in the experiment (ISMPL), the expected value of the estimated volume per acre is

$$\mathrm{EYK} = \frac{\sum_{j=1}^{\mathrm{ISMPL}} \mathrm{YK}_j}{\mathrm{ISMPL}}$$

The bias in the estimated volume per acre (BYK) is

BYK = EYK - ACRVOL

where ACRVOL is the true volume per acre in the population. The Monte Carlo variance—i.e., the variance of the ISMPL estimates of the volume per acre—is symbolized by VYK, and the Monte Carlo estimate of the expected value of the sample-based estimate of variance is symbolized by ESMVYK. The bias in the variance estimate, BSMVYK, equals ESMVYK-VYK. The estimated volume, true variance, and sample-based estimate of variance can be printed out, along with bias in volume and variance estimates for increments of every 100 transects. In this way, the point in the simulation process where the estimates for each experiment become stable is apparent.

Uses of SLASH and INTRSCT

The outputs of the sampling subroutine of INTRSCT generate two types of information to be used as background information for examining residue-survey alternatives. These are (1) estimates of statistical properties of populations that possess certain distinct characteristics found commonly in nature, and (2) estimates of expected values of various statistical properties of populations under different population characteristics, sampling rules, or sampling-unit designs.

Estimates of statistical properties are useful for planning residue inventories. An estimate of the population variance for a chosen line length and arrangement are used in the computation of the number of such lines or line clusters required to meet an expected precision. Pickford and Hazard (1978) provide insight into this problem.

The second type of information will have direct applicability in sampling nonrandom population conditions. It will provide guidelines about when sampling certain populations with the line intersect method is efficient or not, and what kind of sampling units to use. As mentioned previously, this subject is currently under study by the authors.

For example, the characteristics produced as output of INTRSCT are: bias in estimated average residue volume per unit area; variance of the estimated residue volume per unit area; and bias in the sample-based estimated variance of the average volume per unit area.

These three statistical properties of a particular sampling design will be known only if the true average residue volume per unit area and the population variance are known. Both of these parameters generally require that the population be enumerable in terms of the aggregate of all the sampling units contained in it. This is not possible with the line intersect sampling unit because potential locations for lines are infinite. The total or mean residue per unit area is determined, as mentioned earlier, by ignoring the line intersect sampling unit and simply accumulating the volume of all the pieces of residue in the population created by SLASH. The true variance in mean volume per acre is unknown, and thus must be estimated by the Monte Carlo variance.

The other factor that must be introduced to complete the decision problem is cost. If a cost function relates the cost of sampling to different populations, different sampling rules, and different sampling-unit designs, then the optimum choice of sampling designs and sampling-unit designs can be made for specific population characteristics. Further discussion of this topic is beyond the scope of this paper.

Thus, SLASH and INTRSCT are programs that can be used for both planning residue inventories and performing technique studies to determine the optimum designs and circumstances under which to sample with the line intersect sampling method.

Program Applications

To illustrate the uses of SLASH and INTRSCT, we will describe several of our simulation experiments. The first experiments of prime importance are the calibration runs for testing the entire system.

Calibration

We calibrated the simulation system by creating 20,000 standard normal deviates (i.e., 20,000 random numbers, normally distributed with mean (μ) equal to 0 and variances (σ^2) equal to 1). The deviates were partitioned into 2,000 samples of size 10 ($n = 10$). Each deviate was assumed to be an accumulated sum of squared piece diameters for an individual line and was run through the summary routine to get estimates of the Monte Carlo statistics. After the 2,000 repeated trials, BYK was 0.31 and BSMVYK was 0.0041. Thus, we concluded that the simulator was operating within practical limits.

We repeatedly sampled a population of uniform cylinders created by SLASH and observed the performance of the Monte Carlo statistics. This step established the minimum amount of variability that might be incurred in a population of uniform pieces and provided insight into the number of repeated samples required for uniform populations to get the Monte Carlo estimates to converge to the expected values of the population characteristics.

Cylinders in the population were 12 inches in diameter and 20 feet long. We called this our "matchstick" population. The pieces were created with random orientation and location over the area. The true volume created was 3,562.14 cubic feet per acre (249.25 m^3/ha). This is a realistic volume that might be encountered in a Douglas-fir clearcutting.

Figures 2 and 3 show results from two experiments on the population of cylinders. Samples consisting of ten 75-foot (22.86-m) lines were randomly located over the area. Two thousand such samples were taken. In summary, note that:

- The Monte Carlo estimates of volume per acre (fig. 2) are within about 1 percent of the theoretical volume of 3,562.14 cubic feet per acre (249.25 m^3/ha).

- The variance (fig. 3) stabilizes rather quickly (i.e., at about 600-800 repeated samples).

- The bias in the variance of the estimated total volume does not approach 0 until at least 1,600 repeated samples are taken (fig. 3). The difference at 2,000 repeated samples is less than 2 percent.

Figure 2.—Monte Carlo estimates of cubic feet of residue per acre compared with the true volume per acre for increasing numbers of transects; the population is a random population of cylinders (3.562.1 cubic feet per acre).

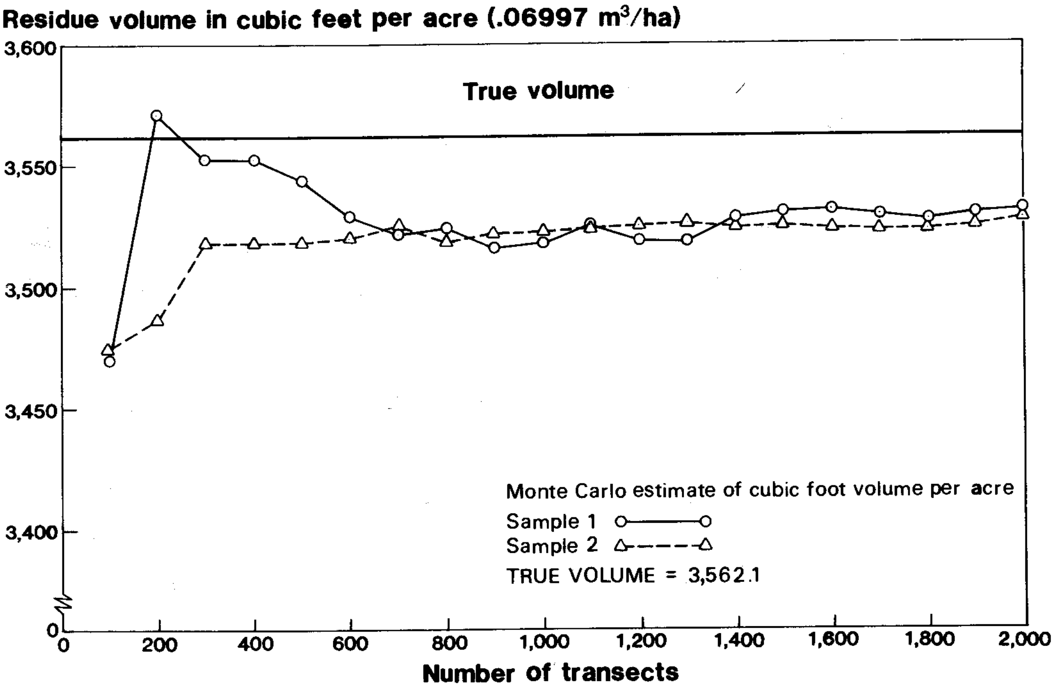
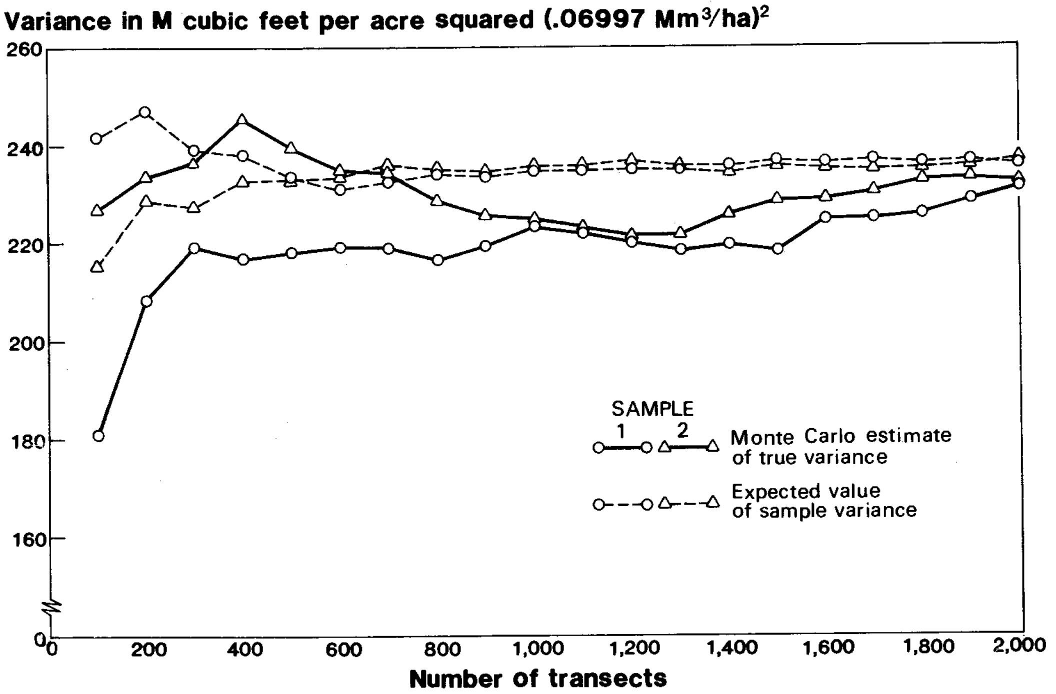


Figure 3.—Monte Carlo estimates of the variance and the expected value of the sample-based estimate of variance for two experiments over increasing numbers of transects; the population is a random population of cylinders (3,562.1 cubic feet per acre).



Performance of the Line Intersect Method

The theory for estimation with the line intersect sampling method for random populations tells us that the estimates of mean residue volume and its estimated variance are unbiased, barring inaccuracies in our simulation procedure. An apparent bias exists in our simulation procedure as an underestimate of residue volume per acre for the population that had pieces reoriented in the vicinity of the boundary. A bias in the variance estimate also exists and usually amounts to an overestimate of about 2 percent. For our purposes, we accept these biases as being within a practical limit.

In the calibration section, we mentioned that legs 75 feet in length produced a bias of approximately a negative 1 percent. Actually, we ran legs of 10, 25, 50, 75, 125, 150, and 250 feet; the bias was plotted over leg length for the matchstick and for a frustum population (table 1, fig. 4). Note that the bias decreased from about -2 percent for legs of 10 feet to a small positive bias for legs of 250 feet. The negative bias approaches or slightly exceeds zero bias for legs $\geq$ 250 feet in the matchstick population. In the frustum population—with taper (i.e., pieces 3 in x 8 in x 20 ft), the bias also decreased with increasing leg length. One possible explanation is that short legs have a larger probability per unit length of line of intersecting the boundary area, where nonrandom orientation occurs for the fixed-dimensioned populations, than do longer legs. In this instance, the density of pieces in the corners of the plot is less than the remainder of the plot, and the distribution of piece volume along the straight boundary will tend to be oriented along the boundary. With shorter sample legs, samples are more likely to be taken entirely within boundary or corner regions of nonuniform piece density or piece volume. Longer legs will tend to smooth out these local population variations.

Table 1 — Bias in volume-per-acre estimates as influenced by sample leg length for populations that required modifying the orientation of pieces intersecting the boundary

	Bias* in frustum population (loading = 799.83 ft^3/ac)			Bias* in matchstick population (loading = 3,562.14 ft^3/ac)		
Leg length	Legs/ transect	Ft^3/ac	Per- cent	Legs/ transect	Ft^3/ac	Per- cent
10	75	-15.06	-1.88	75	-59.29	-1.66
25	30	-14.11	-1.76	30	-47.73	-1.34
50	15	-9.21	-1.15	30	-30.80	-0.86
50				15	-17.35	-0.49
75	10	-10.35	-1.29	10	-37.46	-1.05
75				10	-31.68	-0.89
125	6	-4.08	-0.51	12	-7.63	-0.21
125				6	-13.82	-0.39
150	5	-0.51	-0.01	5	-26.28	-0.74
250	3	+5.39	+0.68	3	-4.84	-0.14
250				3	+7.92	+0.22

*Bias is defined as the difference between the actual and estimated volume per acre. Residue pieces are randomly distributed on a 5.07-acre (470 x 470 feet square) area. Estimates are the average of 1,500 repeated trials.

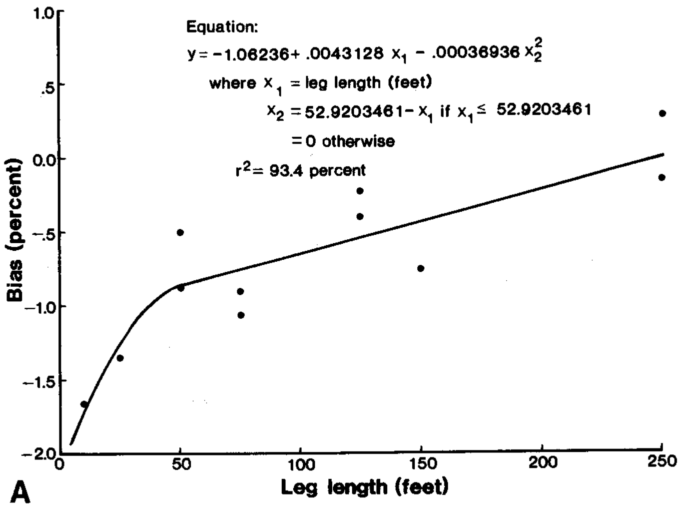


Figure 4a.—Bias in the estimate of residue volume per acre for a frustum population.

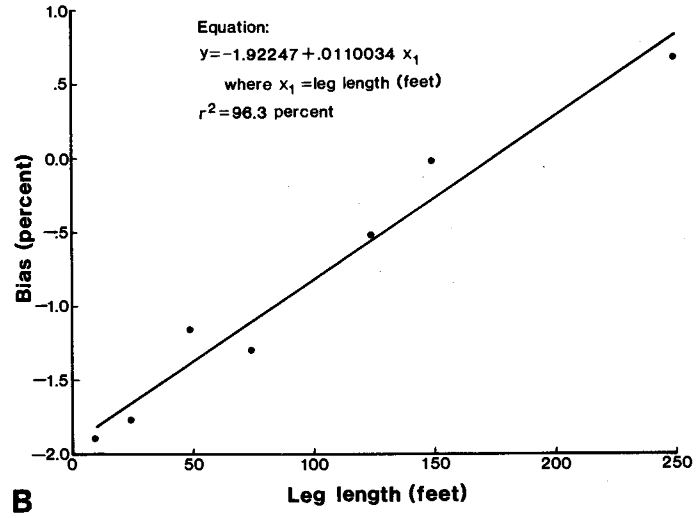


Figure 4b.—Bias in the estimate of residue volume per acre for a matchstick population.

After our initial investigations of the population of cylinders, we introduced constant piece-taper, change in geometric configuration, variation in population density, and length and diameter distributions into the populations. Such experiments should attempt to produce populations with characteristics as near as possible to actual residue populations.

Selecting orientation and spatial-distribution parameters of simulated populations can be arbitrary for the purposes of studying their effect on estimates. Both Warren and Olson (1964) and Bailey (1968) suggest that the orientations of cable-yarded residues can be described by a triangular frequency distribution. Warren and Olson further suggest that, for their method at least, the differences in results between triangular and random orientation are small and can be ignored. Van Wagner (1968) suggests, however, that strong orientation of elements in a population can lead to biased estimates. De Vries (1972) agreed and verified Van Wagner's estimates of possible bias.

Very little information exists on the orientational distribution of residue elements. One example of empirical spatial distributions is our use of low-level aerial photographs of clearcut residues of sufficient scale and resolution to permit measurement. Each photo contains a tenth-acre plot marked on the ground; thus, we can measure not only orientation, but length and spatial distribution as well, for individual elements.

The photographed plots contain strongly oriented logging residue that appear typical of cable-yarding logging (fig. 5), as well as plots where residue appeared nearly random in both orientation and spatial distribution (fig. 6).

Figure 5.—Strongly oriented logging residue, typical of cable-yarded logging.

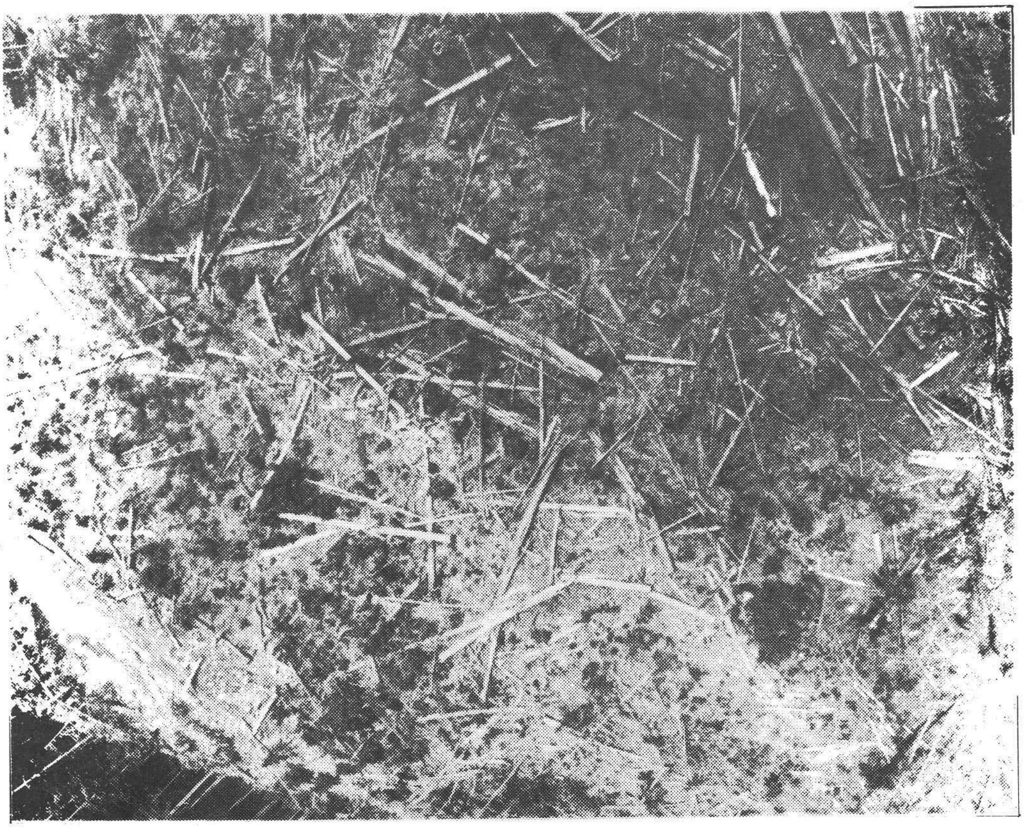


Figure 6.—Randomly oriented and distributed logging residue.

We analyzed the photos by digitizing both end points of each residue element in each plot; correcting for distortion in the photo; and plotting histograms of orientation, length, and distribution of midpoints in the x- and y-coordinate directions (figs. 7-14). Although these are only two case-histories, we consider them typical of patterned and random distributions.

The residue in figure 6 appears nearly random in orientation and distribution; that in figure 5 is strongly oriented, with its axial orientation apparently triangularly distributed. The spatial distribution, although visually nonrandom, is not a simple function of either x- or y-coordinate location. Axial orientation could influence inventory results; the spatial distribution might not, if transects are randomly located. These observations would not be important except for the desirability of systematic location of sample legs in actual inventories. When applied to populations, such as those represented by figure 6, the performance of systematically located, line intersect samples is unknown and unpredictable. Residue populations resembling figure 5 are common, yet are not the most extreme directional orientation that can be encountered. If cableways are parallel to each other and at regular intervals, a strongly nonrandom orientation is imposed on the resulting residues. We need to know how systematic adaptations of the line intersect method will perform in such instances. To test these populations, we are currently developing populations with certain arbitrary distributions. We are looking at random clumps to simulate tractor logging, row-converging to simulate cable logging, and row-parallel to simulate a skyline logging operation between parallel roads.

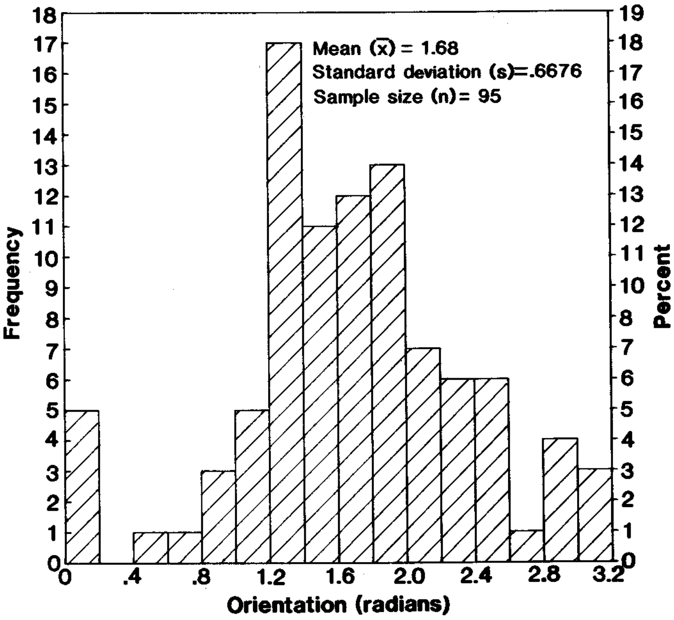


Figure 7.—Frequency distribution of piece orientation for the residue appearing in figure 5. Orientation is expressed in radians (180° = 3.1416 radians).

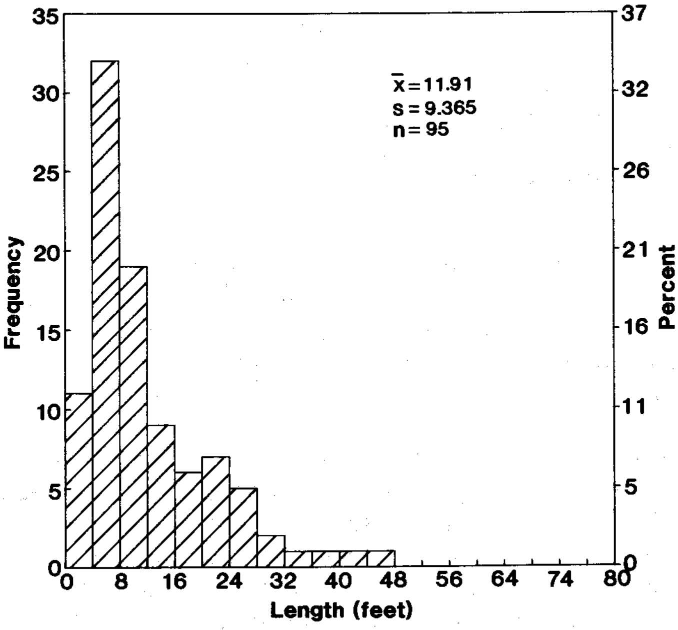


Figure 8.—Frequency distribution of piece length in feet, for the population in figure 5.

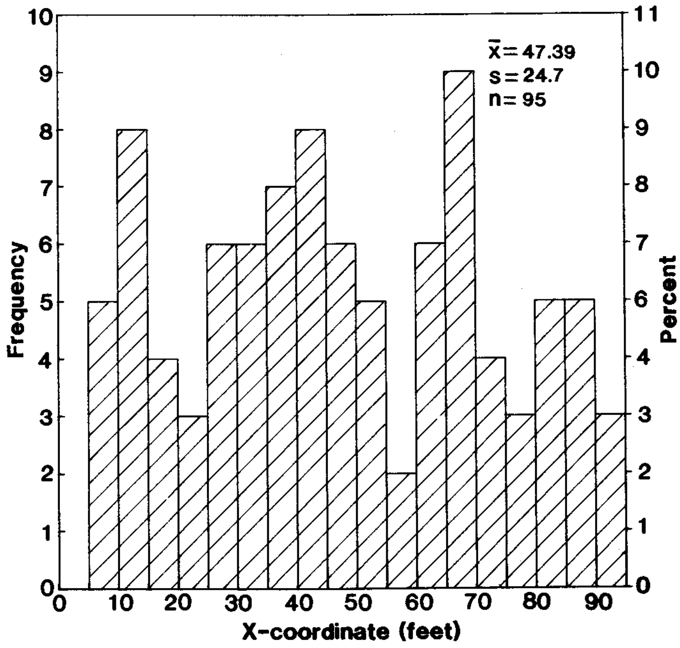


Figure 9.—Frequency distribution of spatial distribution in the x-coordinate in feet, for the population in figure 5.

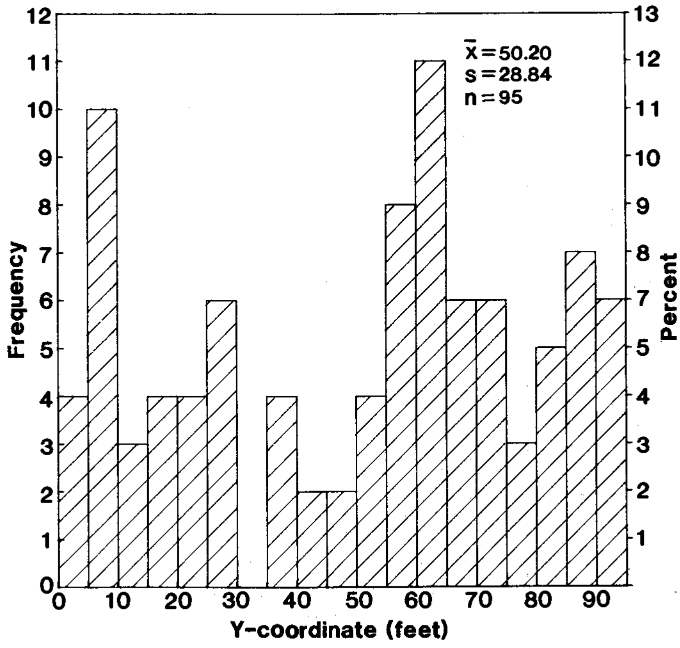


Figure 10.—Frequency distribution of spatial distribution in the y-coordinate in feet, for the population in figure 5.

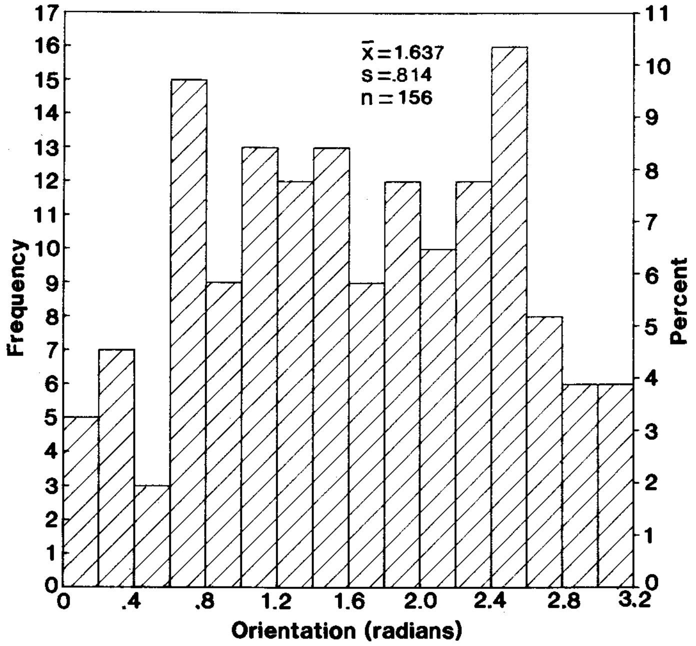


Figure 11.—Frequency distribution of piece orientation for the population in figure 6. Orientation is expressed in radians (180° = 3.146 radians).

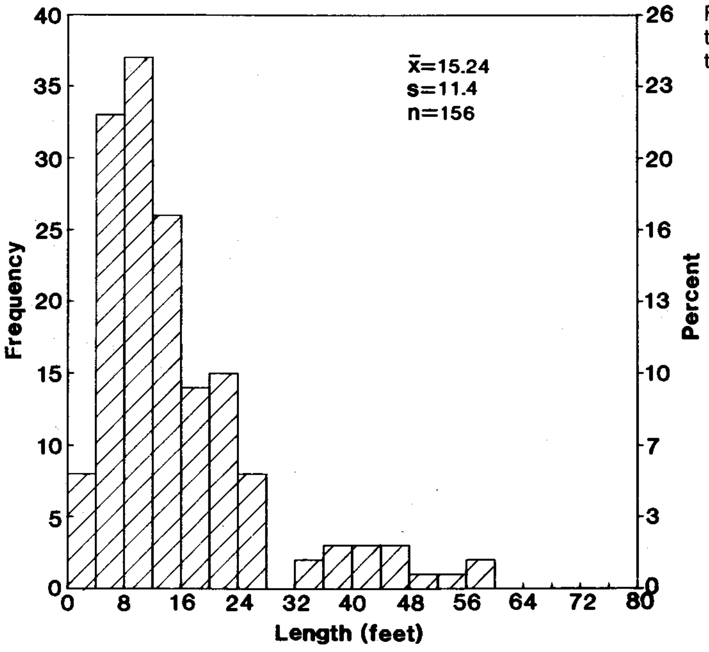


Figure 12.—Frequency distribution of piece length in feet, for the population in figure 6.

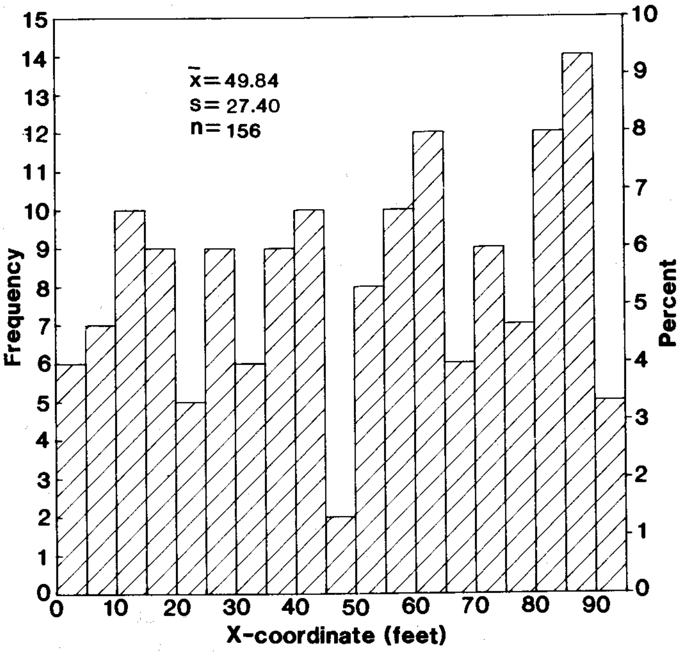


Figure 13.—Frequency distribution of spatial distribution in the x-coordinate in feet, for the population in figure 6.

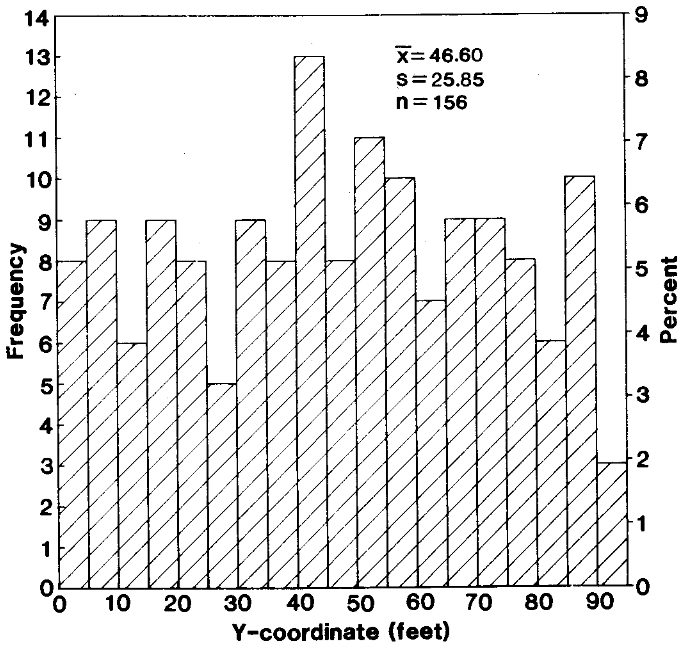


Figure 14.—Frequency distribution of spatial distribution in the y-coordinate in feet, for the population in figure 6.

Significance of These Simulators

Although the concept of simulating line intersect inventories seems simple enough, development of a working simulator exposed numerous subtleties in computer logic, the modeling of geometric populations, and statistical problems of Monte Carlo sampling with replacement. The obvious applicability of this simulator, and the need to answer certain questions about the properties of line intersect sampling made us wonder why such a simulator had not already been developed. The problems we encountered in developing SLASH and INTRSCT seem to us to be reason enough why this is the first, if not the only, such simulator described in the literature. We hope that these programs have addressed these problems in a fashion sufficiently general to permit other users of the line intersect technique to explore its properties and to improve and expand its usefulness.

Acknowledgment

We thank Franklin Ward, Pacific Northwest Forest and Range Experiment Station, Portland, for the aerial photographs of clearcut residues.

Literature Cited

Bailey, G.F. Evaluation of the line-intersect method of logging residue. Report VP-X-23. Victoria, B.C.: Canadian Department of Fisheries and Forestry, Forest Products Laboratory; 1969. 41 p.

DeVries, P.G. A general theory on line intersect sampling with application to logging residue inventory. Report 73-11. Wageningen, Netherlands: Madelingen Landbouwhogeschool; 1973. 23 p.

Howard, J.O. Logging residue, volume, and characteristics. Resour. Bull. PNW-44. Portland, OR: U.S. Department of Agriculture, Forest Service, Pacific Northwest Forest and Range Experiment Station; 1971. 26 p.

Pickford, S.G.; Hazard, J.W. Simulation studies on line intersect sampling of forest residue. For. Sci. 24(4):469-483.

Van Wagner, C.E. The line intersect method in forest fuel sampling. For. Sci. 14(1): 20-26.

Warren, W.G.; Olsen, P.E. A line intersect technique for assessing logging waste. For. Sci. 10(3):267-276.

Appendix

FORTRAN Listings of Programs

```
      PROGRAM SLASH (INPUT,OUTPUT,TAPE3)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C            PURPOSE. GENERATE COLLAWASH POPULATION
C                LOADING =NO. PIECES/32.55.  1000 PIECES APPROX =
C                 30 TONS/ACRE OR 8751 CU FT.
C
C            INPUT.
C                     CARD-FREQ. DIST. FOR GENERATING FUELBED
C                      DATA READ BY SUBROUTINE *DISTRIB*
C                       CARD 1     FMT(8A10)  IDENTIFIER
C                       CARD 2-9   FMT(4F10.0) CUM. FREQ. OF PIECES
C                                     IN SIZE CLASS (I,J)
C                       CARD 10-11 FMT(4(F2.0,3X)/9(F2.0,3X))
C                                  4 LENGTH AND 9 DIAMETER CLASSES
C            OUTPUT.
C                    TAPE3 - MODE IS BLOCKED BINARY
C
C            SUBROUTINES.
C              *COORD*    GENERATES RANDOM X,Y MIDPOINT COORDINATES OF
C                         SLASH PIECES, AND RANDOM ORIENTATION OF AXIS
C              *CORNER*   ADJUSTS PIECE ORIENTATION NEAR PLOT CORNERS
C                         SO PIECE LIES WHOLLY IN PLOT, OR REJECTS
C              *DISTRIB*  READS CUM. FREQ. DISTRIB. OF PIECE DIMENSIONS
C                         FROM INPUT
C              *LID*      ADJUSTS PIECE ORIENTATION NEAR PLOT BOUNDARY
C                         (UPPER) SO PIECE LIES WHOLLY IN PLOT
C              *LIMIT*    RANDOMLY DRAWS PIECE LENGTH AND DIAM. FROM
C                         CUM. FREQ. DIST. -- CALLED BY *PIECE*
C              *PIECE*    GENERATES PIECE DIMENSIONS AND SUMMARIZES
C                         PIECE POPULATION
C              *SIDE*     ADJUSTS PIECE ORIENTATION NEAR PLOT SIDE
C                         BONDARIES SO PIECE LIES WHOLLY IN PLOT
C              *SIZER*    USED TO ASSIGN LENGTH TO PIECE--CALLED BY
C                         *LIMIT*
C              *FTNBIN*   CREATES BLOCKED BINARY OUTPUT DISK FILE
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      COMMON/C/DUMMY(53)
C
C                        BLK/C/ CONTAINS LENGTHS, SMALL-END DIAM.
C                         AND CUM. FREQ. DIST. OF PIECE LEN/DIAM.
C                         USED TO GENERATE THE FUELBED AND ARE PASSED
C                         TO *DISTRIB*,*LIMIT*,*PIECE*, AND *SIZER*
C
      DIMENSION DATA(6,1000)
C
      DATA LOGUN/3/
      DATA VOL/0.0/
      DATA SVOL,N,HML/0.,0,0./
C
C                         INITIALIZES SUMMING VARIABLES "VOL", "SVOL",
C                         COUNTERS "N,","HML".  SETS LOGICAL OUTPUT
C                          UNIT TO 3 AND OVERIDES SYSTEM DEFAULT TO
C                          CREATE BLOCKED BINARY DISK FILE.  "HML"
C                          IS VARIABLE USED TO STORE 1/2 MAX. PIECE
C                          LENGTH WHICH IS USED BY FOLLOWING ROUTINE
C                          *INTRSCT*
C
C
      CALL FTNBIN(1,1,LOGUN)
      REWIND 3
      CALL DISTRIB
C
C                SEED FOR RANDOM NUMBER GENERATOR SET BY
C                USER TO INSURE DIFFERENCES BETWEEN RUNS
C                AND TO ALLOW REPEATABILITY FOR SUCCESSIVE
C                RUNS WHEN DESIRED.
C
      R = RANF(164249.)
C
C                GENERATE FUELPIECE DATA
C
  100 CALL COORD(X,Y,T)
C
```

```
C                DETERMINE LOCATION OF PIECE MIDPOINT WITH RESPECT
C                TO PLOT BOUNDARY AND PIECE LENGTH
C
      IF(X.LT.10.) 1,7
    1 XP=X
      IF(Y.LT.10.) 5,2
    2 IF(Y.GT.460.) 4,3
C
C                MIDPOINT ALONG LEFT HAND SIDE
C
    3 CALL SIDE(XP,T)
      GO TO 20
C
C                MIDPOINT IN UPPER LEFT CORNER
C
    4 YP=470.-Y
      ISIGN=-1
      GO TO 6
C
C                MIDPOINT IN LOWER LEFT CORNER
C
    5 ISIGN=1
      YP=Y
    6 CALL CORNER(XP,YP,T,ISIGN,IFLAG)
      IF(IFLAG.EQ.1) 100,20
    7 IF(X.GT.460.) 8,15
    8 XP=470.-X
      IF(Y.LT.10.) 13,10
   10 IF(Y.GT.460.) 12,11
C
C                MIDPOINT ALONG RIGHT HAND SIDE
C
   11 CALL SIDE(XP,T)
      GO TO 20
C
C                MIDPOINT IN UPPER RIGHT CORNER
C
   12 ISIGN=1
      YP=470.-Y
      GO TO 14
C
C                MIDPOINT IN LOWER RIGHT CORNER
C
   13 ISIGN=-1
      YP=Y
   14 CALL CORNER(XP,YP,T,ISIGN,IFLAG)
      IF(IFLAG.EQ.1) 100,20
   15 IF(Y.LT.10.) 16,17
C
C                MIDPOINT  ALONG BOTTOM EDGE
C
   16 YP=Y
      GO TO 19
   17 IF(Y.LT.460.) GO TO 20
C
C                MIDPOINT ALONG TOP EDGE
C
      YP=470.-Y
   19 CALL LID(YP,T)
   20 CALL PIECE(P,D)
C
C                  COMPUTE END DIAMETERS OF PIECE USING TAPER OF
C                  1 INCH IN 4 FEET WHERE "P" IS PIECE LENGTH,
C                  "D" IS SMALL END DIAMTER, "D2" IS LARGE END
C                  DIAMETER.
C
      D2 = (.25 * P) + D
      VOL=VOL + (3.1416*P*(D**2 + D*D2 + D2**2) / 1728.)
      SVOL = VOL
   21 CONTINUE
      N=N+1
      IF(P.GT.HML) HML = P
      DATA(1,N) = X
      DATA(2,N) = Y
      DATA(3,N) = T
      DATA(4,N) = P
      DATA(5,N) = D
      DATA(6,N) = D2
```

```
C
C                 TEST IF END OF JOB
C
C                 "N" IS NO. OF PIECES GENERATED.  THIS PROGRAM
C                  GENERATES 1000 PIECES ON A 470 X 470 FT.
C                  AREA.
C
      IF(N .EQ. 1000) 200,100
C
C                 END OF JOB
C
  200 CONTINUE
      ZERO=0.
C
C                    PRINT OUT SUMMARY TABLE OF ACTUAL AND DESIRED
C                    PIECE DIMENSION DISTRIBUTIONS.  ENTRY POINT
C                    *LOOKSEE* IN SUBROUTINE *DISTRIB*
C
      CALL LOOKSEE(ZERO,ZERO)
      PRINT 301,VOL
  301 FORMAT(///1H0,*1000 PIECES WITH TOTAL VOLUME OF *,F10.2)
      HML = AINT(HML/2. + 0.5)
      WRITE(3) SVOL,N,HML
      END FILE 3
      DO 310 J=1,1000
  310 WRITE(3) (DATA(I,J),I=1,6)
      REWIND 3
      STOP
      END

      SUBROUTINE COORD(X,Y,T)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C                    PURPOSE.  GENERATE FUEL PIECE X,Y COORDINATES
C                    OF PIECE MIDPOINT, AND RANDOM PIECE AXIAL
C                    ORIENTATION.  THE FACTOR "470" IS THE PLOT
C                    SIZE AND THE FACTOR "6.28319" IS 2 PI RADIANS
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      X=RANF(0.)*470.
      Y=RANF(0.)*470.
      T=RANF(0.)*6.28319
      RETURN
      END

      SUBROUTINE CORNER (X,Y,T,ISIGN,IFLAG)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C            PURPOSE. IF PIECE MIDPT. LIES IN PLOT CORNER, AND IF
C            PIECE CAN BE ROTATED WHOLLY INTO PLOT, DO SO, ELSE
C            SET "IFLAG"=1 WHICH CAUSES MAIN PROGRAM TO REJECT PIECE.
C            IF PIECE MIDPT. NOT IN CORNER, "IFLAG"=0 AND PIECE
C            ACCEPTED.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      IFLAG=0
      Z=SQRT(X**2+Y**2)
      IF(Z.LT.14.) 2,1
C
C              "Z" = DISTANCE FROM CORNER TO PIECE MIDPT.
C              IF Z> HALF THE LENGTH OF LONGEST PIECE, IT
C              MAY FALL PARTLY OUTSIDE AREA BOUNDARY, AND
C              PIECE IS REJECTED.  IF Z<= HALF LENGTH OF
C              LONGEST PIECE, THE PIECE ORIENTATION IS
C              RANDOMLY ASSIGNED AMONG THE POSSIBLE
C              ORIENTATIONS WHICH WILL KEEP ENTIRE PIECE
C              WITHIN THE AREA.  IN THIS PROGRAM, THE
C              MAX. PIECE LENGTH WAS 28 FT, SO THE TEST
C              CONSTANT WAS 14 (Z.LT.14.).
C
```

```
    1 A=ASIN(X/14. )
      B=ACOS(Y/14. )
      C=A-B
      T=1. 5708-ISIGN*(RANF(0. )*C+B)
      RETURN
    2 IFLAG=1
      RETURN
      END

       SUBROUTINE DISTRIB
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C                 PURPOSE.  READ IN THE CUM. FREQ. DISTRIBUTION
C                           USED TO GENERATE FUELBED PIECE
C                           DIMENSIONS.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                         GENERATE THE FUELBED
      DIMENSION TITLE(24)
      REAL LIM, LEN, L
      COMMON/C/LEN(4), DIA(9), LIM(9, 4), L, D, M, K
      READ 15, TITLE
   15 FORMAT(8A10)
      PRINT 16, TITLE
   16 FORMAT(1H1, 3(13X, 8A10, /)///)
      READ 12, ((LIM(I, J), J=1, 4), I=1, 9)
   12 FORMAT (4F10. 0)
      READ 10, LEN, DIA
   10 FORMAT (4(F2. 0, 3X)/9(F2. 0, 3X))
      RETURN
      END

      SUBROUTINE LID(Y, T)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C              PURPOSE. ROTATE PIECE LYING NEAR PLOT BOUNDARY
C                       SO THAT IT LIES WHOLLY WITHIN PLOT.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      T=ASIN(Y/14. )*(RANF(0. )-0. 5)*2.
C              THE CONSTANT "14. " IS HALF THE MAXIMUM PIECE
C              LENGTH IN THE POPULATION FOR THIS VERSION OF
C              THE PROGRAM.  SEE NOTE IN SUBROUTINE *CORNER*
C
      IF(T. LT. 0. ) T=T+3. 141593
      RETURN
      END

      SUBROUTINE LIMIT
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C             PURPOSE.  COMPUTE "K" WHICH LOCATES THE DIAMETER
C                       ASSIGNED TO A FUEL PIECE BY LOCATING
C                       THE POSITION OF A RANDOM NUMBER (URN)
C                       WITHIN THE CUM. FREQ. DIST. LIST.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
%
      REAL LIM, LEN, L
      COMMON/C/LEN(4), DIA(9), LIM(9, 4), L, D, M, K
      URN=RANF(0. )
      DO 2 K=1, 9
      IF (LIM(K, 4). LT. URN) 2, 1
    1 CALL SIZER(URN)
      RETURN
    2 CONTINUE
      END

      SUBROUTINE PIECE (A, B)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C             PURPOSE.  1. GENERATE THE LENGTH AND DIAMETER OF
C                          EACH FUEL PIECE USING THE CUM. FREQ.
C                          DIST.
C                       2. USING ENTRY POINT *LOOKSEE*, LIST THE
C                          ACTUAL FREQ. DIST. CREATED.
C
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      DIMENSION ARRAY(9,4),DIF(9,4)
      REAL LEN,LIM,L
      COMMON/C/LEN(4),DIA(9),LIM(9,4),L,D,M,K
      DATA ARRAY/36*0./
      CALL LIMIT
      A=L
      B=D
    1 ARRAY(K,M)=ARRAY(K,M)+1.
      RETURN
      ENTRY LOOKSEE
      TMP=0.
      DO 30 I=1,9
      DO 30 J=1,4
      SAVE = ARRAY(I,J)/1000.
      ARRAY(I,J) = SAVE + TMP
      DIF(I,J) = LIM(I,J) - ARRAY(I,J)
   30 TMP = ARRAY(I,J)
      PRINT 23
   23 FORMAT(*0*,15X,*ACTUAL DISTRIBUTION*,15X,*CALCULATED DISTRIBUTION
     1 (1000 VALUES)*, 5X,*DIFFERENCE BETWEEN ACTUAL AND CALCULATED*/)
      DO 24 K=1,9
   24 PRINT 11, (LIM(K,M),M=1,4),(ARRAY(K,M),M=1,4),(DIF(K,M),M=1,4)
   11 FORMAT (*0*,4(4X,F6.4), 5X,4(4X,F6.4), 5X,4(4X,F6.4))
      RETURN
      END

      SUBROUTINE SIDE(X,T)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C                 PURPOSE.  ROTATE A PIECE WHICH LIES NEAR
C                           SIDE BOUNDARY OF THE PLOT SO
C                           THAT IT LIES WHOLLY WITHIN PLOT
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      T=1.5708+ASIN(X/14.)*(RANF(0.)-0.5)*2.
C
C              THE CONSTANT "14." IS HALF THE MAXIMUM PIECE
C              LENGTH IN THE POPULATION FOR THIS VERSION OF
C              THE PROGRAM.  SEE NOTE IN SUBROUTINE *CORNER*
C
      RETURN
      END

      SUBROUTINE SIZER(RAN)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C               PURPOSE.  DETERMINE LENGTH AND DIAMETER OF
C                         PIECE USING "K" FROM *LIMIT* AND
C                         "M" DETERMINED IN THIS ROUTINE,
C                         BY TESTING AGAINST RANDOM NUMBER
C                         "RAN" PASSED AS "URN" FROM
C                         *LIMIT*.  LENGTH AND DIAMETER
C                         FOUND IN LOOKUP TABLE READ IN FROM
C                         *DISTRIB*
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      REAL LIM,LEN,L
      COMMON/C/LEN(4),DIA(9),LIM(9,4),L,D,M,K
      DO 2 M=1,4
      IF(LIM(K,M).LT.RAN)2,1
    1 L=LEN(M)
      D=DIA(K)
      RETURN
    2 CONTINUE
      END
```

```
      PROGRAM INTRSCT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,SLASH,TAPE2=
     1SLASH,DATA,TAPE1=DATA)
C
C     PURPOSE. LINE INTERSECT SAMPLING
C
C     I/P.
C                 INPUT - CARD READER
C                 TAPE2 - (SLASH) DATA           (TAPE,BLK BIN)
C                 TAPE5 - (INPUT) RUN HEADER
C     I/O.
C                 TAPE1 - (DATA)  SUMMED DIAM**2  (DISK,BLK BIN)
C     O/P.
C                 OUTPUT- PRINTER
C                 TAPE6 - (OUTPUT)
C     FILE DESCRIPTION.
C **         TAPE5 (INPUT) - RUN HEADER DATA
C
C             CARD COL  FMT  VAR      DESC
C
C               1  1-5  F5.0 TLEG     TRANSECT LEG LENGTH
C               2  1-3  I3    LSZ     NMBR LEGS/TRANSECT
C               3  1-4  I4    ISMPL   NMBR OF TRANSECTS
C                  5-8  I4    INCR    SUMMARY PRINT INCREMENT
C               4  1-6  I6    IX      ODD NUMBER
C
C **         TAPE2 (SLASH) / PLOT DATA
C                 FILE 1 - PLOT VOLUME, PIECE COUNT
C                 FILE 2 - FUEL PIECE DATA
C     SUBROUTINES.
C                 RSEED  - SET SEED FOR RANDOM NMBR GENERATOR
C                 SAMPL  - CALC,SUM SQ OF PIECE INTRSCTION DIAM BY TR LEG
C                 SUMMARY- SUMMARIZE DATA, PRINT REPORT
C                 TCORD  - TRANSECT COORDINATES
C                 XRANGE - FIND PIECE TABLE X COORD RANGE
      DIMENSION LOGUN(2)
C
      COMMON ARRAY(6,1500),NUMPC,XLL,XUL,LL,LU
      COMMON/CROSS/YI
      COMMON/DATAIN/ISMPL,ACRVOL,TLEG,ALINE,LSZ,NSEED,INCR
      COMMON/PIECE/X,Y,T,P,D1,D2
C
      DATA LOGUN/1,2/
C
C          INPUT FORMATS
C
    1 FORMAT(F5.0/I3/2I4/I6)
C
C          PRINT FORMATS
C
   10 FORMAT(1H1,A10,5X,*S.PICKFORD*,5X,*LINE INTERSECT SAMPLING - *,
     A  *MATCHSTICK FUELBED (TAPE 2641,MATCHSTICK 3/75)*//)
   55 FORMAT(1X,*NO EOF ENCOUNTERED ON FILE -SLASH-*)
C
C---       INITIALIZE
C
      CALL DATES(DAT)
      CALL FTNBIN(1,2,LOGUN)
      NSW1 = 0
C
C          SET SEED FOR RANDOM NMBR GENERATOR
C
      R = RANF(2.)
      PRINT 10, DAT
C
C          READ HEADER INFO
C
      READ 1, TLEG,LSZ,ISMPL,INCR,IX
      NSEED = IX
   20 ALINE=TLEG*LSZ
      NN=ISMPL*LSZ
C
C---       READ PLOT HEADER DATA
C
   50 READ (2) TRUVOL,NUMPC,HML
      READ (2)
      IF (EOF,2) 56,54
```

```
   54 PRINT 55
      GO TO 300
   56 DO 57 MINPT=1,NUMPC
C
C---       READ FUEL PIECE DATA
C
   57 READ (2) (ARRAY(I,MINPT),I=1,6)
      ACRVOL=TRUVOL/5.07117
C
C---       SEARCH ROUTINE
C             (A) CREATE TRANSECT
      DO 100 ILEG=1,NN
      CALL RSEED (IX)
C
C          1ST ENDPOINT OF LEG
C
      CALL TCORD(TX1,TY1)
C
C          2ND ENDPOINT OF LEG
C
  110 ORIENT=RANF(0.)*3.14159*2.
      TXDELT=TLEG*COS(ORIENT)
      TYDELT=TLEG*SIN(ORIENT)
      XNEW=TXDELT+TX1
      YNEW=TYDELT+TY1
C
C          TEST IF TRANSECT LEG FALLS WITHIN PLOT
C               PLOT SIZE IS 5 ACRES (470 X 470 FT)
C
      IF(XNEW.LT.0..OR.XNEW.GT.470..OR.YNEW.LT.0..OR.YNEW.GT.470.)111,
     1 112
  111 GO TO 110
  112 TX2=XNEW
  120 TY2=YNEW
      XMAX=AMAX1(TX1,TX2)
      XMIN=AMIN1(TX1,TX2)
      YMAX=AMAX1(TY1,TY2)
      YMIN=AMIN1(TY1,TY2)
      DELX=TX1-TX2
      DELY=TY1-TY2
      A=DELY/DELX
      E=A*TX1-TY1
      IFLAG=0
      XLL = XMIN - HML
      XUL = XMAX + HML
      YLL = YMIN - HML
      YUL = YMAX + HML
C
C            (B) SEARCH FOR PIECE INTERSECTIONS
C
      CALL XRANGE
      DO 98 IPC=LL,LU
      X=ARRAY(1,IPC)
      Y=ARRAY(2,IPC)
C
C          TEST IF PIECE MIDPOINT COORDINATES WITHIN RECTANGLE
C          FORMED BY LEG + HALF THE LENGTH OF THE LONGEST PIECE
C
      IF(X.LT.XLL .OR. X.GT.XUL) 98,70
   70 IF(Y.LT.YLL .OR. Y.GT.YUL) 98,71
   71 CONTINUE
      T=ARRAY(3,IPC)
      C = -TAN(T)
      F=C*X+Y
C
C          G = A*D - B*C = A + C   WHERE D=1, B=-1
C
      G = A + C
      IF (ABS(G) .GT. 0.) 72, 98
C
C          XI = (D*E - B*F)/(A*D - B*C) = E+F/A+C WHERE D=1, B=-1
C
   72 XI = (E + F)/G
      IF (XI .LT. XMIN .OR. XI .GT. XMAX)  98,73
   73 YI = (A*F - C*E)/G
      IF (YI .LT. YMIN .OR. YI .GT. YMAX)  98,74
```

```
   74 ALN = SQRT((X-XI)**2 + (Y-YI)**2)*2.
      P=ARRAY(4,IPC)
C
C         TEST IF INTERSECTION IS VALID
C
      IF(ALN - P) 85,80,98
C
C         IF PIECE INTERSECTS LEG ENDPOINT, ACCEPT ALTERNATE INTRSCTIONS
C
   80 IF(NSW1) 82,81,82
   81 NSW1 = 1
      GO TO 85
   82 NSW1 = 0
      GO TO 98
   85 CONTINUE
      IFLAG = 1
      D1=ARRAY(5,IPC)
      D2=ARRAY(6,IPC)
      CALL SAMPL(ILEG,ALN,IFLAG)
   98 CONTINUE
      IF(IFLAG.EQ.0) 99,100
   99 CALL SAMPL(ILEG,ALN,IFLAG)
  100 CONTINUE
      IF(IFLAG .EQ. 1) CALL SAMPL(-9,0,1)
      ENDFILE 1
      CALL SUMMARY
  300 STOP
      END


      SUBROUTINE RSEED (IX)
C
C     PURPOSE. SET SEED FOR RANDOM NUMBER GENERATOR
C              SERIES BASED ON A PRIME NUMBER
C
C              REF--IBM/360 ≠RANDU≠ ROUTINE
C                     IBM SCIENTIFIC SUBROUTINE PACKAGE, H20-0205-0
C     NOTE.    1.≠IX≠ IS ANY ODD INTEGER NUMBER WITH 9 OR FEWER DIGITS
C                ≠IY≠ IS INTEGER BETWEEN 0 AND 2**31
C                ≠YFL≠ IS RANDOM NUMBER IN THE RANGE 0 TO 1.0
C              2. ROUTINE WILL PRODUCE 2**29 TERMS WITHOUT REPEATING
C
C              ARGUMENT FOR FUNCTION ≠RANF≠
C                X .GT. 0 - NEW SERIES STARTED BASED ON LRGST PRIME IN X
C                X .EQ. 0 - RANDOM NMBR GIVEN FROM AN ESTABLISHED SERIES
C                X .LT. 0 - LAST RANDOM NMBR GIVEN
C              RESULT IS REAL NMBR ≠R≠, WHERE 0 .LT. R .LT. 1
C
      DATA IX/135731/
C
      IY = IX * 65539
      IF(IY) 5,6,6
    5 IY = IY + 2147483647 + 1
    6 YFL= IY
      YFL= YFL*.4656613E-9
      IX = IY
      R = RANF(YFL)
      RETURN
      END


      SUBROUTINE SAMPL(N,ALN,IFLAG)
C
C     PURPOSE.
C              CALC PIECE DIAMETER AT POINT OF INTERSECTION
C              SUM SQUARES OF DIAMETERS BY TRANSECT LEG
C     PARAMETERS.
C         N      - (I/P) CURRENT TRANSECT LEG
C         ALN    - (I/P) 2XDIST FROM PIECE MIDPOINT TO PT OF INTERSECTION
C         IFLAG - (I/P) 0-INVALID INTERSECTION, 1- VALID INTERSECTION
C     O/P.
C              TAPE1
      COMMON/CROSS/YI
      COMMON/PIECE/X,Y,T,P,D1,D2
C
      LOGICAL L1,L2
      DATA DSUM/0./
      DATA I/1/
```

```
C
      IF(IFLAG.EQ.1) 100,200
  100 IF(N.EQ.I) 1,2
    1 DM = (D2+D1)/2.
      TAPR = (D2-D1)/P
      L1 = (T-3.14159) .LT. 0.
      L2 = (Y .LT. YI)
      IF((L1.AND.L2) .OR. .NOT. (L1.OR.L2))10,11
   10 DI=DM+ALN*TAPR/2.
      GO TO 12
   11 DI=DM-ALN*TAPR/2.
   12 DSQ=DI**2
      DSUM=DSUM+DSQ
      RETURN
    2 WRITE (1) DSUM
C
C          TEST IF END OF JOB
C
      IF(N .LT. 0) RETURN
      DSUM=0.
      I=I+1
      GO TO 1
  200 WRITE (1) DSUM
      I = I + 1
C
C          TEST IF PREV LEG HAD NO INTERSECTIONS
C
      IF(DSUM .EQ. 0.) RETURN
      DSUM=0.
      WRITE (1) DSUM
      I = I + 1
      RETURN
      END


      SUBROUTINE SUMMARY
C
C     I/P.
C               TAPE1
      COMMON /DATAIN/ISMPL,ACRVOL,TLEG,ALINE,LSZ,NSEED,INCR
C
C          PRINT FORMATS
C
    9 FORMAT(1H ,4X,F10.2* CU. FT./ACRE, TRUE VOLUME*/
     1  11X,I4* TRANSECTS*/12X,I3* LEGS/TRANSECT*/10X,F5.0* FT/LEG*/
     2  10X,F5.0* FT/TRANSECT*/ 9X,I6,* 1ST SEED*//)
   11 FORMAT(1H0,10X,I4,* TRANSECTS*/
     8        25X,F10.2* = M.C. ESTIMATE OF VOLUME/ACRE*/
     1  20X,F15.2* = M.C. ESTIMATE OF TRUE VARIANCE OF YK*/
     2 20X,F15.2* = M.C. ESTIMATE OF E.V. OF SAMPLE BASED EST. OF VAR(YK
     3)*/
     425X,F10.2* = BIAS OF YK*/
     5   20X,F15.2* = BIAS OF LITTLE V(YK)*)
C
C
C
      CVOL(X,Y) = 373.1944*(X/Y)
      REWIND 1
      TYJK=TYJK2=TYK=TYK2=SMVYK=0.
      PRINT 9,ACRVOL,ISMPL,LSZ,TLEG,ALINE,NSEED
      N = 0
      IF(INCR .LT. 1) INCR = ISMPL
      DO 50 K=1,ISMPL,INCR
      LIM = INCR
      IF(INCR .EQ. ISMPL) GO TO 25
      NWRK = K + INCR - 1
      IF(NWRK .GT. ISMPL) LIM = ISMPL - K - 1
   25 CONTINUE
      DO 40 L=1,LIM
      N = N + 1
      DO 30 J=1,LSZ
      READ (1)  DSUM
      YJK=CVOL(DSUM,TLEG)
      TYJK=TYJK+YJK
      TYJK2=TYJK2+YJK**2
   30 CONTINUE
      YK=TYJK/LSZ
      TYK=TYK+YK
      TYK2=TYK2+YK**2
      SMVYK=SMVYK+(TYJK2-TYJK**2/LSZ)/((LSZ-1)*LSZ)
      TYJK=TYJK2=0.
```

```
   40 CONTINUE
      EYK = TYK /N
C
C         (JWH 4/21/76) DENOMINATOR CHANGED FROM (N-1) TO N BECAUSE
C                       VYK ESTIMATES (SIGMA**2/J),  NOT (S**2/J)
C
      VYK =(TYK2 - TYK**2 / N) / N
      ESMVYK = SMVYK / N
      BYK=EYK-ACRVOL
      BSMVYK=ESMVYK-VYK
      PRINT 11, N,EYK,VYK,ESMVYK,BYK,BSMVYK
   50 CONTINUE
      RETURN
      END


      SUBROUTINE TCORD(X,Y)
      R=RANF(0.)
      R1=AINT(R*1000.)/1000.
      R2=AINT((R-R1)*1000000.)/1000.
      X=470. * R1
      Y = 470. * R2
      RETURN
      END


      SUBROUTINE XRANGE
C
C     PURPOSE. FIND PIECE TABLE SUBSCRIPT RANGE FOR X COORD OF MIDPOINT
C              SUCH THAT RANGE INCLUDES LEG + HALF THE LENGTH OF THE
C              LONGEST PIECE
C     PARAMETERS.
C              ARRAY - (I/P) FUEL PIECES
C              NUMPC - (I/P) NMBR PIECES IN ≠ARRAY≠
C              XLL   - (I/P) LOWER LIMIT LEG X COORD
C              XUL   - (I/P) UPPER LIMIT LEG X COORD
C              LL    - (O/P) LOWER LIMIT PIECE X COORD SUBSCRIPT
C              LU    - (O/P) UPPER LIMIT PIECE X COORD SUBSCRIPT
C
      COMMON ARRAY(6,1500),NUMPC,XLL,XUL,LL,LU
C
      INDEX = NUMPC/2
      IS = INDEX
C
C---      LOWER X SEARCH LIMIT
C         TEST IF LEG X COORD LOWER LIMIT IN LOWER HALF PIECE TABLE
C
      IF(XLL .LT. ARRAY(1,IS)) GO TO 50
C
C *       LEG IN UPPER HALF
C
   25 INDEX = INDEX/2
      LI = IS
      IS = IS + INDEX
      IF(IS .GE. NUMPC) GO TO 35
      IF(ARRAY(1,IS) .LE. XLL) GO TO 25
   35 LL = LI
      GO TO 75
C
C *       LEG IN LOWER HALF
C
   50 CONTINUE
      IS = IS/2
      IF(IS .GT. 1) GO TO 55
      IS = 1
      GO TO 60
   55 IF(XLL .LE. ARRAY(1,IS)) GO TO 50
   60 LL = IS
   75 CONTINUE
      DO 80 IS=LL,NUMPC
      IF(ARRAY(1,IS) .GE. XLL) GO TO 85
   80 CONTINUE
      IS = NUMPC + 1
   85 LL = IS - 1
C
C---      UPPER X SEARCH LIMIT
C
  100 CONTINUE
      DO 120 JS=LL,NUMPC
      IF(ARRAY(1,JS) .GT. XUL) GO TO 130
  120 CONTINUE
      JS = NUMPC
  130 LU = JS
      RETURN
      END
```