A DISCRETE BIO SIMULATION - THE POPULATION REGULATION OF TURTLES

Keith P. Tognetti
Australian National University
Canberra, Australia

Summary

The nesting behaviour of sea turtles has been simulated using SIMSCRIPT 1.5. It is known that these nesting turtles may destroy the eggs in other nests in attempting to dig their nests. This simulation has been used to determine a quantitative relationship between the fraction of nests destroyed by the self induced mortality due to digging and the level of the turtle population.

This paper emphasises the programming and simulation aspects of the study.

Introduction

This simulation is based on the biological work carried out by H.R. Bustard of the Australian National University on a population of green sea turtles (Chelonia mydas).

The model is briefly as follows: A turtle attempts to dig a pit in its nesting beach. If the pit does not collapse it lays its eggs and covers the nest with sand. It may make up to four abortive attempts during the one night before it is successful. It then returns to the nesting beach for up to four more lays at intervals of about a fortnight.

While digging its pit the turtle may destroy other nests. The purpose of this simulation is to provide a discrete model of the digging and laying behaviour of a realistic population of turtles and obtain a quantitative relationship between the fraction of nests destroyed and the population level. This relationship is to be used as a basis for determining whether the self-destruction of nests by the turtles is a limiting factor in the regulation of their numbers.

The paper details the simulation and programming techniques in an attempt to demonstrate that the modelling of ecological systems is a task that could be readily undertaken by a biologist with the help of a language such as SIMSCRIPT.

Simplified Model and its Extension to the Simulation Model

An oversimplified form for the model would be as follows: A wave of turtles arrives at a beach and lays at random on the beach. A second wave of turtles of the same size arrives and also lays at random. In the process some of the previous nests are destroyed. Thus formulated the problem is equivalent to a sampling process without replacement, i.e. the distribution of nests overlaid for each wave of turtles is according to a hypergeometric distribution. Even this simplification makes the computation of the distribution function for the number of nests destroyed after a given number of nights rather difficult. However a certain structure can be obtained by dealing with expectations (refer Appendix 2).

This structure is lost as soon as the model is extended. Firstly the number of turtles arriving per night for their first beaching is highly variable. Secondly, the number of aborted digs (each of which can result in nest destruction) is a random variable. Thirdly, each turtle has to be scheduled to return to the beach a random number of times at intervals such that the turtles arriving on any given night are a mixture of first, second, third and fourth layers. Additionally there are minor extensions to account for the arrival distribution around high tide and the difference in nest occupancy times between layers and non layers.

General Organisation of the Simulation

The program is best (and fully explained) by referring to the flow chart (figs 1-8), decision table (fig 9) and the SIMSCRIPT listing (fig 10). Each vertical column in the decision table represents a logical path through the procedure scheduled by the event AR. The entries in the table above the double horizontal line simply represent the decisions to be tested, whereas the entries below indicate the action to be taken if these decisions are satisfied. The numbers on the flow chart are related to the statement numbers on the SIMSCRIPT listing. To approximate the field data, digging sites are selected from a distribution which is uniform parallel to the beach and triangular inland.

The digging activity for a turtle for any one night is according to the branching process detailed in Appendix 1.

The simulation was programmed in SIMSCRIPT 1.5 for the Control Data 3600 at the Commonwealth Scientific and Industrial Research Organisation installation in Canberra, Australia.

An event notice (4 words of storage) is allocated for each active turtle in the system. A temporary event (4 words) is allocated for each nesting site on the beach in which at some stage eggs have been laid.

Turtle Storage. Apart from FIR which determines the run characteristics and is entered once only for each run, there is only one other endogenous event AR. AR decides whether the turtle is starting or finishing digging (NTYP). It schedules the number of times it will return to the beach (LAYT) and calls a subroutine (SCRT) to decide how many aborted attempts (ABORT) will be made before laying. AR could have been broken up into many events, one for each begin dig and one for each finish dig. This would have simplified decision logic somewhat, however the associated increase in the work required to dynamically transfer attributes from one notice to the next far outweighs this. Consequently the simulation is organised with essentially one event notice for each turtle and this remains in the same location in memory for its digging life. This notice is of course rescheduled each time a new start or finish dig is required.

As a result, the organisation of the simulation is similar to G.P.S.S. which has only the one transaction for each entity. In fact, if one wished to go to the trouble AR could have been set out in a block layout equivalent to G.P.S.S.

Laying Site Storage. Sites that have nests are stored in a list ranked on the coordinates. Each site has attributes for coordinates, list processing (successor predecessor), laying status (occupied, eggs, no eggs) and number of sets destroyed.

Whenever a turtle has to select a spot for laying (subroutine SPOT) firstly a site is selected from the bivariate distribution for the laying pattern and then the site list is scanned to determine whether a new addition to the list is required. If the turtle selects a new site but does not lay, this site is removed from the list when it has finished digging.

Burst of Turtles. The program has been written so as to allow burst of turtles (either normal first arrivals or diggers only) to be scheduled on a particular night. Such bursts may be superimposed on a normal run to assess the destructive worth of a turtle with time. Some care has been taken to ensure that the original sequence of pseudorandom numbers in the run is not affected by the burst.

Discrete Simulation Languages

There are three major reasons for using a special simulation language for discrete modelling based on the following requirements.

The first is the requirement for a list processor so as to store information dynamically and with more flexibility than provided by that of subscripted arrays. The second is to ease the housekeeping associated with scheduling and keeping track of the various events. The third is the requirement to be able to easily extend or modify the program.

The need for list processing arises from the uncertainty in the length of queues, sets and other lists that are part of the model. In this simulation, for example, neither the number of turtles active in the system at any time nor the number of laying sites is known. Consequently an overstatement in the storage allocated to either of these reduces the storage available to the other. This difficulty is compounded when the turtles are broken up into lists of begin laying and finish laying types for the first, second, third, etc. time.

An efficient timing routine based on list processing operation on a calendar set with most of the housekeeping kept out of sight of the programmer is a necessity for the efficient modelling of a large stochastic system to ensure that the programmer will not be distracted from his proper business of structuring the system.

Apart from these major reasons for using a discrete simulation language, there are many minor advantages available, such as: automatic provision of routines for sampling from tabular data representing probability distribution functions; routines for evaluating mean, variance, etc. of lists; accumulation of time weighted quantities and so on. All of these minor advantages could be provided in the form of subroutine extensions to a FORTRAN program, however none of the major characteristics could be easily provided in this manner.

An attempt was made to translate this SIMSCRIPT program into the exclusively IBM language G.P.S.S./360. It was difficult to make a comparison between these languages in terms of compilation times and memory because apart from the different machines the IBM system does not appear to readily provide these details, but it is known that in such a comparison G.P.S.S. II compared very badly with SIMSCRIPT I requiring twice the storage and being seven times slower. However on the basis of structure the G.P.S.S. language appears to be deficient for this simulation. Firstly, it is not a general purpose language but rather a special purpose language. It is believed that its special purpose nature makes it reasonable for modelling traffic systems and job shop scheduling. However it does expect lists to be associated with calendar sets and because of this it sets up the structure of members of this list (transactions) so that they may be used in the timing sets (both future and current). The result is that there is no provision in G.P.S.S. for the list processing of the temporary entities of SIMSCRIPT without paying a severe penalty (9 full words per entity, compared with less than one word in SIMSCRIPT). To represent the site storage in the list processing available in G.P.S.S. required the splitting of transactions and the placing of these transactions on a user chain, rather a messy business. Additionally G.P.S.S. suffers because its block structure is supposed to be adequate for its specific purpose, and it has only a vestigial language which must return everything in fixed point form and requires tedious scaling. The fact that FORTRAN subroutines are not available to G.P.S.S. gives it very little appeal to a programmer who wishes to generate his random samples using analytic inversion. Additionally because of its interpretive nature it would appear that G.P.S.S. is quite unsuitable for interactive on line simulations as it has to be recompiled for each run.

Recently IBM have released their language SEAL which is simply an extension of SIMSCRIPT. Unbelieveably this package is claimed to have a memory requirement of 215K whereas the memory available to a 256K 360/50 installation with monitor is 214K! Consequently we are not as yet in a position to use the compiler on the Australian National University installation. However it is certainly structurally very attractive. Apart from free form initialisation and definition the input output format has been extended to character manipulation and logical variables. An extremely attractive feature is the relationship of entities to events.

To overcome the difficulty described earlier of being required to supply a unique event notice entity for each event, SEAL allows the use of compound structure for the event name.

Thus we could have

CREATE TURT

CAUSE ARRIV OF TURT AFTER DEL

Then later after the event ARRIV OF TURT has occurred we could schedule

CAUSE DEP OF TURT AT TIME + DEL

Both events ARRIV and DEP OF TURT refer to the one entity TURT and it is not necessary to reset any activities into new event notices to distinguish between arrivals and departures.

The Conversion of Biological Data into Steps in the Simulation

As an example of the technique for processing biological data and representing the biological structure as procedures within the program consider the turtle's digging activity on any one night.

This information was originally collected and made into a histogram showing the frequency of the first, second and third, etc. attempts to lay. The first step was to represent the information in this histogram by a branching process (Appendix 1) with conditional probabilities allocated to the various outcomes. This branching representation can then be readily coded into SIMSCRIPT using the decision commands available (SUBROUTINE SCRT(IAR)) which subroutine we will now consider.

The first statement puts a random decimal number between 0 and .999 into XYU. ABORT(IAR) contains the counter for the number of aborted attempts to lay and this gives the subscript for PLA the probability of laying at the ABORT(IAR)th attempt.

XYI is simply the probability of leaving after this dig. The first decision statement determines whether the random number is greater than the probability of laying plus staying in which case the turtle will leave after digging (ABORT set to 1).

If not, it is tested to see whether it is greater than the probability of laying in which case it is a stayer and the ABORT is incremented by 1 so that the next time it comes into the routine the correct values of the probabilities will be read.

The data for the conditional probabilities are automatically initialised at the beginning of the simulation from the initialisation deck (cards 20 and 21 at end of listing, figure 10). The card following 20 contains the probabilities of laying and the card following 21 contains the probabilities of staying.

Results

A series of simulation runs was carried out over a range of turtle populations representative of and extending the field situation.

Figure 11 is a plot of the percentage of nests destroyed against total turtle population. A linear regression was shown to fit these results with a correlation coefficient of .86.

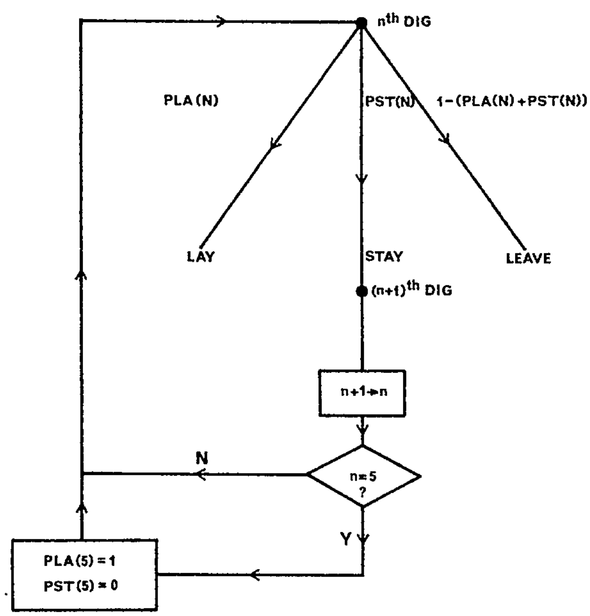
The regression line has the form fraction of nests destroyed = .005 turtle population + .024.

Figures 12 and 13 show typical output results from the simulation as plotted by the CALCOMP plotter.

Let us now examine the biological implications of this result. Consider a situation in which the population has been operating in a steady state for some time. By steady state is meant that the number of female adults at the time of nesting is constant as is the number of hatchlings. The probability of one of these hatchlings surviving to adulthood and returning to the beach for nesting (about 8 years) would be constant as would be the survival probability of the original population of females. Also the population of the next cycle of females and hatchlings would be the same as in the first cycle. Now assume that this process was subjected to a sudden fluctuation which resulted in the hatchlings being increased by a factor p. Then assume that the other factors regulating the population are constant and that the number of turtles returning to the beach as adults will be increased by a factor less than p (as the contribution from the surviving adults is unaffected). The number of hatchlings produced by this increased population of females would certainly be larger than before the fluctuation. However it would be subjected to the self induced mortality due to digging and this would apply to each cycle, being most pronounced on the first cycle and gradually diminishing, resulting in the peak due to the fluctuation being gradually flattened out. On this basis, digging mortality can be regarded as a self regulating mechanism in the control of the turtle population.

This simulation as it stands cannot say anything about the relative importance of the digging mortality compared with other limiting factors. However, if the nature of these other limiting factors were known this model could be readily extended to include these factors. As was mentioned earlier, a distinct advantage of a special simulation language such as SIMSCRIPT is that the program can be much more readily modified or extended in complexity compared with a program written in FORTRAN.

This model is now being extended to include the predation of the hatchlings as they swim across the reef to the open sea. Only a small amount of data and some speculation is available for this region. Across the reef and beyond is mostly speculation.



APPENDIX 1

BRANCHING PROCESS FOR DIGGING ACTIVITY

Appendix 2

Simple Formulation of Model

s = no. of sites for nesting

r = no. of nests containing eggs

j = no. of arrivals per night

It is assumed that each turtle digs once and lays. The laying site is selected from a uniform distribution. The site selection procedure for each turtle is independent of other turtles except that it will not disturb other turtles laying for that night i.e. sampling without replacement.

Thus the probability of k sites being destroyed is

$$P_k = \frac{\binom{r}{k}\binom{s-r}{j-k}}{\binom{s}{j}}$$

The distribution is hypergeometric. Consider the expectation for this distribution

$$= \frac{rj}{s}$$

Let EN_n = Expected number of nests after nth night

ED_n = Expected number of nests destroyed on nth night

$p = \frac{j}{s}$, $q = 1 - p$

TD_n = Expected total nests destroyed after n nights

First Night

$$EN_1 = j$$

$$ED_1 = 0$$

Second Night

$$ED_2 = EN_1 . j/s = pEN_1 = pj$$

$$EN_2 = j + EN_1 - pEN_1$$

$$= j\ (1 + q)$$

Thus $ED_n = pj\ (1 + q + \ldots q^{n-2})$

$$ED_n = j\ (1 - q^{n-1}) \qquad (1)$$

$$EN_n = j\ (1 + q + \ldots q^{n-1})$$

$$EN_n = s(1 - q^n) \qquad (2)$$

Also $TD_n = \sum_{k=1}^{n} ED_k$

$$= s(np + q^n - 1)$$

$$TD_n = nj - s(1 - q^n) \qquad (3)$$

Expected Fraction of Nests Destroyed

$$= \frac{TD_n}{nj}$$

$$= 1 - (1 - q^n)/pn \qquad (4)$$

As $n \to \infty$

$EN_n \to s$, $ED_n \to j$, $TD_n \to nj$,

fraction destroyed $\to 1$

If only a fraction f of the nests contain eggs, the above formula (1) to (3) are multiplied by f with $p = \frac{j}{s}$ unchanged.

Now for small p

$$1 - \frac{1-q^n}{pn} \sim 1 - \frac{1-(1-np+n(n-1)p/2-\ldots)}{pn} \sim \frac{(n-1)p}{2}$$

Thus the expected fraction of nests destroyed is directly proportional to the turtle population for small p and large n.

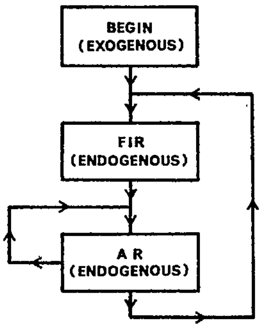


FIG. 1

GENERAL CAUSAL RELATIONSHIP OF EVENTS

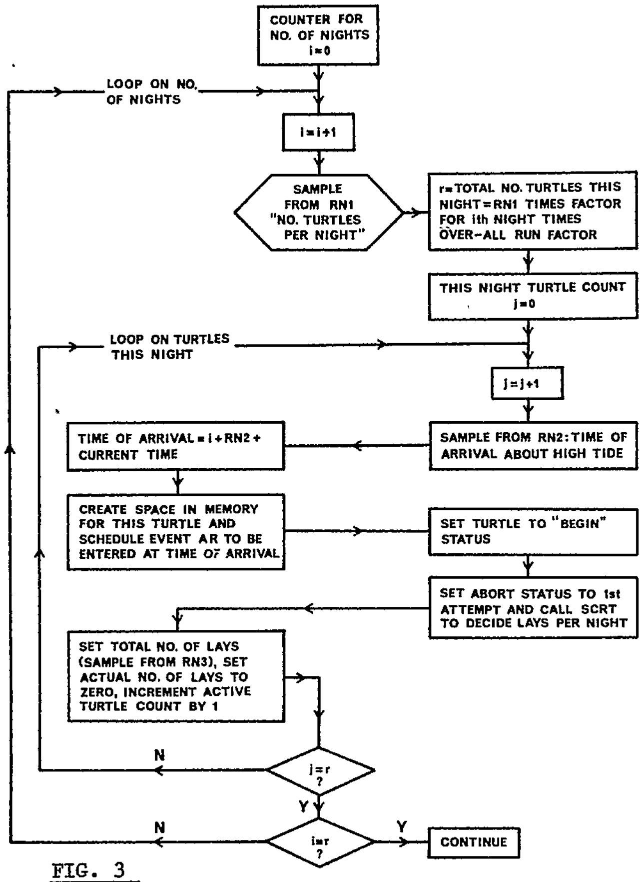


FIG. 3

SCHEDULE FOR TURTLE ARRIVALS (In FIR)

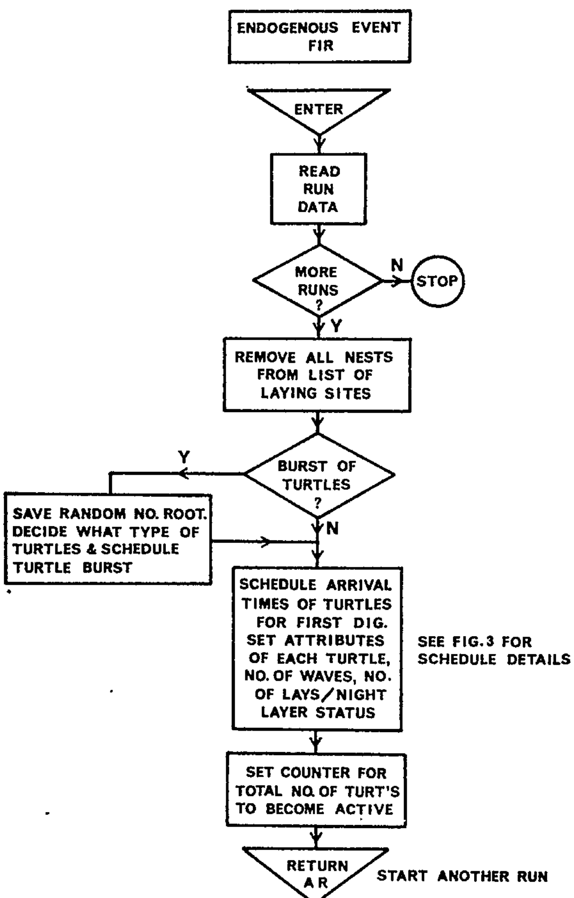


FIG. 2

ENDOGENOUS EVENT FIR

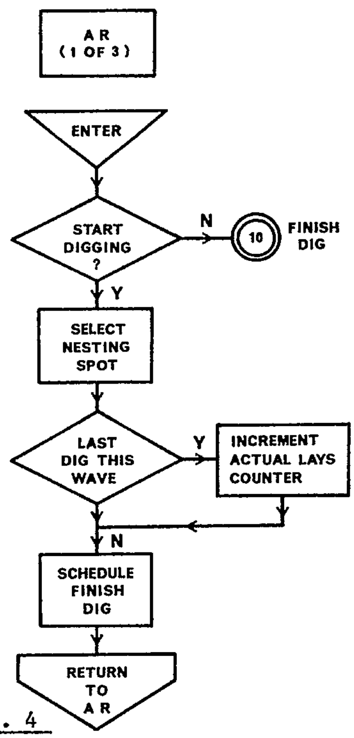


FIG. 4

ENDOGENOUS EVENT AR (1 of 3)

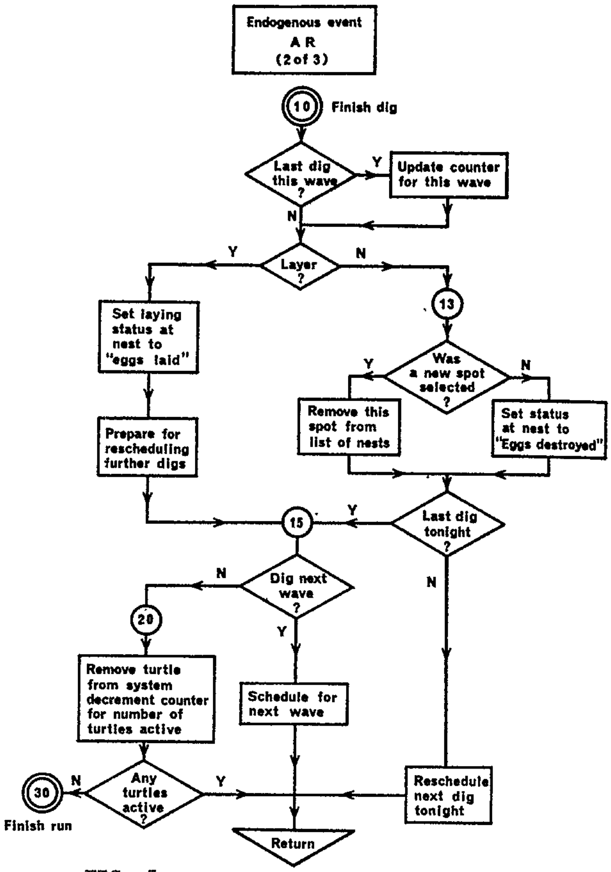
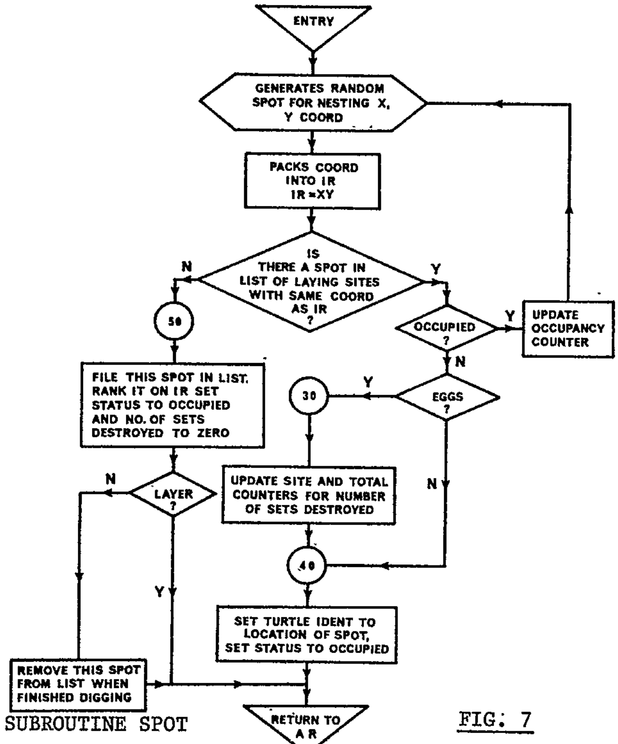


FIG. 5

ENDOGENOUS EVENT AR (2 of 3)



SUBROUTINE SPOT

FIG. 7

Positions turtle in laying zone, increment nests destroyed counter and laying status. Also tags turtle with location of the spot.

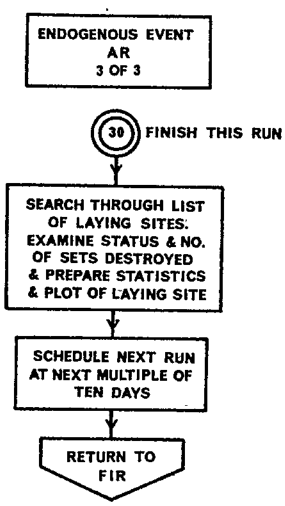


FIG. 6

ENDOGENOUS EVENT AR (3 of 3)

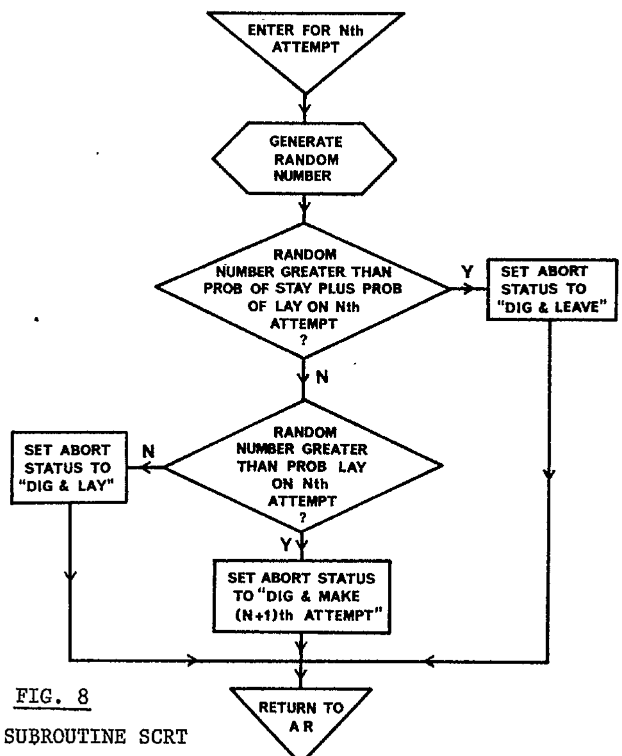


FIG. 8

SUBROUTINE SCRT

Determines laying sequence for one night. Returns detail of path according to the branching process detailed in Appendix 1.

FIGURE 9 DECISION TABLE FOR AR

		START		FINISH																					
DECISION	Last dig this wave	Y	N	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	N	N	N	N	N	N	N	N	N
	Layer?			Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	Y	N	N	N	N	N	N	N	N
	New spot selected?					Y	N	Y	N	Y	N	Y	N					Y	N	Y	N	Y	N	Y	N
	Last dig tonight?					Y	Y	N	N	Y	Y	Y	Y					Y	Y	N	N	Y	Y	Y	Y
	Dig next wave?			Y	Y	Y	Y			N	N	N	N	N	N	N	N	Y	Y			N	N	N	N
	Any turtles active?									Y	Y	N	N	Y	Y	N	N					Y	Y	N	N
ACTION	Select spot	*	*																						
	Increment actual lays counter	*																							
	Schedule finish laying	*	*																						
	Increment wave counter			*		*	*	*	*	*	*	*	*	*		*									
	Status = "Eggs Laid"			*	*									*	*	*	*								
	Reschedule next dig			*	*									*	*	*	*								
	Remove spot from list					*		*		*		*						*		*		*		*	
	Status = "Eggs Destroyed"						*		*		*		*						*		*		*		*
	Reschedule next dig tonight							*	*											*	*				
	Schedule for next wave			*	*	*	*																		
	Remove turtle from system, decrement active count									*	*	*	*	*	*	*	*					*	*	*	*
	Finish this run											*	*			*	*							*	*
	Return	*	*	*	*	*	*	*	*	*	*			*	*			*	*	*	*	*	*		

FIG. 10

SIMSCRIPT LISTING

```
C
C
C
C
C
C        TURTLE SIMULATION
C
C        LISTING OF A SIMSCRIPT PROGRAM AND DATA
C        FOR CONTROL DATA CORP. 3600
C
C        SIMSCRIPT VERSION 1.5
C
C        DEFINITION CARDS.....
*N AR    4          N NTYP   31/4 I   1RN1          I US QC      XIRNK  L
*N FIR   2          N LAYT   32/4 I   2RN2          F UL
*                   N LAYA   41/2 I   3RN3          I US
*                   N ID     42/2 I   4RN4          I US
*T COR   4          T FQC    11/2 I   5IOX          I
*                   T SQC    12/2 I   6IOY          I
*                   T IRNK   2    I   7FQC          I
*                   T ISTAT  3    I   8LQC          I
*                   T NSD    4    I   9LNQ          I
*                   N ABORT  33/4 I  10KFL          I
*                   N ISVR   34/4 I  11KF           I
*                                    12FKF          F
*                                    13ITSD         I
*                                    14TTL          F
*                                    15KPA          I
*                                    16NIT          I
*                                    17XNIT    1    F
*                                    18IOC          I
*                                    19LPL          I
*                                    20PLA     1    F
*                                    21PST     1    F
*                                    22TTD          F
*                                    23KOUT         I
*                                    24TAB          F
*                                    25INORR        I
*                                    26IBURR        I
      EVENTS
        1 EXOGENOUS
           BEGIN(1)
        2 ENDOGENOUS
           AR
           FIR
      END
      EXOGENOUS EVENT BEGIN
C       TEMPORARY  ATTRIBUTES,
C
C       NTYP =  ENTITY  TYPE      ( 1 = ARRIVE ) (2 = FINISH LAYING)


C       (3 = HAS DUG ONLY ON NEW SPOT,....DESTROY THAT SITE AT END)
C       LAYA =  ACTUAL NO, OF LAYS PER TURTLE ,SO FAR
C       LAYT =  TOTAL NO, OF LAYS PER TURTLE
C       ID   =  IDENT  OF  COR  ( SITE OCCUPIED)
C       IRNK = X,Y   COORDINATE  FOR  RANKING
C       ISTAT=  COORDINATE  STATUS (1 = OCCUPIED)( 2 = EGGS )
C                                  ( 3= EGGS DESTROYED )
C       NSD   = NO, OF SETS DESTROYED AT THIS POSITION
C       ABORT= STATUS FOR ABORTED LAYS
C         0   = WILL LAY
C         1   = WILL DIG AND LEAVE
C         2   = WILL DIG AND ENTER SECOND TEST
C         N   = WILL DIG AND ENTER N-TH TEST
C
```

```
C
C       PERMANENT ATTRIBUTES,
C       1  -  RN1   -  TURTLES  PER  NIGHT  (I)
C       2  -  RN2   -  TIME OF ARRIVAL ABCUT HIGH TIDE (F) HOURS
C       3  -  RN3   -  LAYS PER TURTLE (I)
C       4  -  RN4   -  DAYS  BETWEEN  LAYS (I)
C       5  -  IOX   -  XCOORD  ON  BEACH  (I,TRI)
C          (THE ABOVE IS NO LONGER READ IN AS A FUNCTION)
C       6  -  IOY   -  YCOORD  ON BEACH  (I,RECT)
C         (THE ABOVE IS NO LONGER READ IN AS A FUNCTION)
C       7  -  FQC   -  FIRST  POINTER     ( FOR QUEUE QC )
C       8  -  LQC   -  LAST    POINTER    ( FOR QUEUE QC )
C       9  -  LNG   -  LENGTH OF KFL  (10)
C         ABOVE IS THE LARGEST NUMBER OF WAVES OF TURTLES,
C       10  -  KFL(I)  -  COUNTER  FOR  I'TH WAVE
C       11  -  KF       -  COUNTER FOR TOTAL NO OF TURTS ACTIVE,
C       12  -  FKF      -  TOTAL NO OF NEW TURTLES
C       13  -  ITSD     -  TOTAL SETS DESTRCYED
C       14  -  TTL      -  TIME BETWEEN LAYINGS (IN DAYS )
C       15  -  KPA      -  ACTIVE NO,/TIME PLOT (0 OR BLANK - PLOT )
C       16  -  NIT      -  LENGTH OF XNIT
C         ABOVE IS THE NUMBER OF NIGHTS OF NEW ARRIVALS
C       17  -  XNIT(I) - MEAN FACTOR FOR THE NO, OF TURTLES ON I-TH NIGHT
C       18  -  IOC      - COUNT  OF THE NO, OF OCCUPIED SITES
C       19  -  LNL  - LENGTH OF PLA AND PST
C       20  -  PLA(I) - PROB OF LAYING AFTER I-1 TRIES
C       21  -  PST(I) - PROB OF STAYING AFTER I-1 TRIES
C
C       ,,HENCE (1 - (PLA(I)+PST(I))) IS THE PROBABILITY OF
C          LEAVING AFTER I-1 ATTEMPTS AT LAYING,
C       22  -  TTD   - TIME TO DIG(IN HOURS)
C       23  -  KOUT  - LAYING PLOT (0 OR BLANK - DO PLOT)
C       24 - TAB  -  TIME AT BEGINNING OF SIMULATION
C       25 - INORR-  INDICATES NORMAL RANDCM ROOT SEQUENCE
C       26 - IBURR-  INDICATES BURST TURTS RANDR SEQUENCE
C       ,,BURSTS REFER TO PROVISION FOR SCHEDULING A BURST OF
C         TURTLES AT ANY TIME TO DETERMINE THEIR DESTRUCTIVE WORTH,
C
C
C
C
C      LOCAL VARIABLES,,,,(IN EVENT FIR)
C       KBUR  - TYPE OF TURTLE BURST
C                     0 - DIGS ONCE AND GOES
C                     1 - NORMAL TURTLE
C       NBUR  - NUMBER OF TURTLES IN BURST
C       TBUR  - TIME OF BURST
C       RN  -  MULTIPLICATION FACTOR FOR RANDOM NO,OF TURTLES
C                 IF VALUE IS LESS THAN ZERC TERMINATE THE PROGRAM


C
C
C   CALLS TO SUBROUTINE PLOT REFER TO   A LIBRARY PLOT ROUTINE FOR
C   GRAPHICAL OUTPUT TO A CALCOMP PLOTTER
C
C
      WRITE ON 61,NIT
      FORMAT(* NIT =*,I8)
      DO,FOR I=(1)(NIT)
      WRITE ON 61,I,XNIT(I)
      FORMAT(* XNIT NO,*,I2,* = *,D6,6 )
      LOOP
      LET TTD = TTD/24,
      WRITE ON 61,TTD
      FORMAT(* TIME TO DIG = *,M4,3,3)
      WRITE ON 61
```

```
      FORMAT(S14,* PLA (I)      +     STA (I)   =    TOTAL *)
      DO,FOR I =(1)(LPL)
      LET XYZZ= PLA(I)+PST(I)
      WRITE ON 61,I,PLA(I),PST(I),XYZZ
      FORMAT(* NO,*,I4,* SUM   *, D6.4,* + *,D6.4,* = *,D6.4 )
      LOOP
C   SET ORIGINS AND SCALES FOR ANY REQUIRED PLOTS
      IF KOUT EQ 0, CALL PLOT(0.,0.,1)
      IF KPA EQ 0 , CALL PLOT(0.,0.,1,4)
      IF KOUT EQ 0, CALL PLOT(5.,10.,2)
      IF KPA EQ 0 , CALL PLOT(10.,50.,2,4)
      CREATE FIR
      CAUSE FIR AT TIME
      RETURN
      END
      ENDOGENOUS EVENT FIR
C
C   THIS FOLLOWING EVENT BEGINS THE SIMULATION.......
C
C     WRITE ON 61, TIME
C     FORMAT(*1IN FIR AT TIME *,M4,3,3)
C
C
C   N.B.....DATA CARDS MUST CONTAIN DUMMY EXOGENOUS EVENT CARD (1.E...15 )
C           THIS ALLOWS DATA TO BE READ IN FROM EVENT FIR
C
      LET TAB = TIME
      READ FROM 60,RN,KOUT,KPA,NBUR,TBUR,KBUR
      FORMAT(D4.5,I8,I8,I2,D3,1,I2 )
      WRITE ON 61,NBUR,TBUR,KBUR,RN,KOUT,KPA,TAB
      FORMAT(* NO, IN BURST =*,I8,*   TIME OF BURST = *,D3.1
     C,* KBUR = *,I8/* NEW RN = *,D6,4,*  KOUT = *,I4,* KPA = *,I4,
     C*   START AT TIME = *,M4,3,3)
C  SET PEN TO ORIGIN (IF REQUIRED)
      IF KOUT EQ 0, CALL PLOT(0.,0.,3)
      IF KPA EQ 0 , CALL PLOT(0.,0.,3,4)
C        IF VALUE OF RN  NEGATIVE ,INDICATES TERMINATION OF PROGRAM
      IF RN LS 0.0,STOP
      IF KOUT NE 0, GO TO 61
C   PLOT AXES
      CALL PLOT(0.,104.,3)
      CALL PLOT(0.,0.,1)
      CALL PLOT(0.,100.,4)
      CALL PLOT(30.,0.,  3)
      CALL PLOT(0.,0.,  4)
   61 IF KPA NE 0, GO TO 60
      CALL PLOT(0.,1000.,3,4)



      CALL PLOT(0.,0.,1,4)
   60 IF QC IS EMPTY,GO TO 50
      REMOVE FIRST IV FROM QC
      DESTROY COR CALLED IV
      GO TO 60
C  SCHEDULE BURSTS
   50 LET IRSAV = RANDR
C RESET ALL INDEX STORAGES
      DO, FOR I=(1)(LNG)
      LET KFL(I) = 0.0
      LOOP
      LET IOC= 0
      LET ITSD = 0
      LET KF=0
C  ADJUST ATTRIBUTES ETC., AND SCHEDULE BURST OF TURTLES
      DO,FOR I = (1)(NBUR)
      CREATE AR
      STORE 1 IN NTYP(AR)
```

```
      STORE 9 IN ISVR(AR)
      STORE 1 IN ABORT(AR)
      STORE 0 IN LAYA(AR)
      IF KBUR EQ 1 , GO TO 10
      STORE 1 IN LAYT(AR)
      GO TO 12
   10 LET LAYT(AR)= RN3
      CALL SCRT(AR)
   12 CAUSE AR AT TIME + TBUR + RN2/24,0
      LET KF = KF + 1
      LOOP
      LET IBURR = RANDR
      LET RANDR = IRSAV
C   LOOP FOR NO OF NIGHTS
      DO TO 4,FOR II=(1)(NIT)
      LET HOLD = RN1
C   RN1 IS NO OF TURTS PER NIGHT
C
C     WRITE ON 61,II,HOLD,XNIT(II)
C     FORMAT(* NO,*,I2,* ,RN1= *,D6,6,* ,XNIT = *,D6,6)
      LET IR=RN*HOLD*XNIT(II)
      WRITE ON 61, IR
      FORMAT(*    (TURTS PER NIGHT) *, I8)
      LET TTIM=II
      DO TO 3, FOR IJ=(1)(IR)
C   RN2 IS TIME OF ARRIV OF TURTS
      LET XY=RN2
      LET CH = XY/24,
      LET TY=TTIM+CH
      CREATE AR
      STORE 1 IN ABORT(AR)
      CALL SCRT(AR)
      STORE 1 IN NTYP(AR)
C   NTYP IS ENTITY TYPE 1 ARRIV TYPE 2 FIN LAYING
C   RN3 IS NO OF LAYS PER TURT
      LET KF=KF+1
      STORE RN3 IN LAYT(AR)
      STORE 0 IN LAYA(AR)
C     WRITE ON 61,AR,NTYP(AR),LAYT(AR),LAYA(AR),KF,CH,TY
C    C,ABORT(AR)
C     FORMAT(*  AR NTYP LAYT LAYA KF CH TY*, 5I8,D6,4
C    C,M4,3,3,* ABORT = *,I8)
      CAUSE AR AT TIME  + TY
    3 LOOP
    4 LOOP


      WRITE ON 61,KF
      FORMAT(* TOTAL NO, OF TURTLES,,,*,I8)
      LET FKF = KF
      LET INORR = RANDR
      RETURN
      END
      ENDOGENOUS EVENT AR
C  THIS EVENT CAUSES ARRIVALS OF TURTLES FOR LAYING ,,,,,,,
C
C
      DIMENSION AN(10)
C     WRITE ON 61,TIME,AR,NTYP(AR),LAYT(AR),ID(AR),LAYA(AR)
C    C,ABORT(AR)
C     FORMAT(*0IN AR,,, TIME ,AR,NTYP,LAYT,ID,LAYA,ABORT,,,*
C    C,M4,3,3,6I8)
      IF  NTYP(AR) NE 1,GO TO 10
C  TURTLE ARRIVES TO DIG OR LAY
      IF LAYA(AR) NE 0, GO TO 90
      IF ABORT(AR) GR 1 , GO TO 90
      LET TATA = TIME - TAB
```

```
      IF KPA EQ 0, CALL PROF(1,,TATA)
   90 STORE 2 IN NTYP(AR)
      IF ABORT(AR) LE 1 ,LET LAYA(AR)=LAYA(AR) + 1
      IF ISVR(AR) EQ 9 , GO TO 92
      LET RANDR = INORR
      CALL SPOT(AR)
      LET INORR = RANDR
      GO TO 93
   92 LET RANDR = IBURR
      CALL SPOT(AR)
      LET IBURR = RANDR
   93 IF ABORT(AR) GR 0 ,GO TO 12
C   RESCHEDULE AFTER LAYING
      LET XY = TIME + TTL
      CAUSE AR AT XY
      GO TO 70
C   RESCHEDULE AFTER DIGGING
   12 LET XY = TIME + TTD
      CAUSE AR AT XY
   70 LET LO = LO
C  70 WRITE ON 61,XY
C     FORMAT(* TIME TO FINISH DIG OR LAY *,M4,3,3)
      RETURN
C  TURTLE FINISHED DIGGING OR LAYING
   10 IF ABORT(AR) LS 2 , LET KFL(LAYA(AR))=KFL(LAYA(AR))+1
      IF ABORT(AR) GR 0 , GO TO 13
C     WRITE ON 61
C     FORMAT(* LAY AND FINISH *)
      STORE 2 IN ISTAT(ID(AR))
      STORE 1 IN ABORT(AR)
C  SCHEDULES  NEXT WAVE
   15 LET LO = LO
C  15 WRITE ON 61, KFL(LAYA(AR))
C     FORMAT(* NO,OF TURTS  IN LAYA WAVE     KFL *, I8)
      IF LAYT(AR) EQ LAYA(AR) , GO TO 20
      CALL SCRT(AR)
      LET  XY=RN4
      LET ROUND = TIME - TTL +,5
      LET INTEG= ROUND
      LET WAIT = INTEG
      LET TOA = RN2/24,0
      LET XZ = XY + WAIT * TOA
C     WRITE ON 61, XY, XZ,TOA


C     FORMAT(* TIME  TO  AND AT NEXT  LAY AND RANDOM TOA *,2M4,3,3
C    C,D6,4)
      STORE 1 IN NTYP(AR)
      CAUSE AR AT XZ
      RETURN
C
C     IF NEW SPOT REMOVE FROM QC
C
   13 IF NTYP(AR) EQ 3 , GO TO 35
      STORE 3 IN ISTAT(ID(AR))
   36 IF ABORT(AR) EQ 1 , GO TO 14
C     WRITE ON 61,ABORT(AR)
C     FORMAT(* DIG AND TEST,,,,,ABORT= *,I8)
C
C  TURTLE IS TESTED FOR (ABORT)-TH ATTEMPT
C
      CALL SCRT(AR)
      IF ABORT(AR) EQ 5 , LET ABORT(AR) = 0
      STORE 1 IN NTYP(AR)
      CAUSE AR AT TIME
```

```
C  NOTE,,, TIME DELAY IS NOT CONSIDERED,
      RETURN
   35 REMOVE ID(AR) FROM QC
      DESTROY COR CALLED ID(AR)
      GO TO 36
   14 LET LO = LO
C  14 WRITE ON 61
C     FORMAT(* DIG AND FINISH,*)
      GO TO 15
C THIS TURTLE FINISHED,,,,,,
   20 LET KF=KF-1
C     WRITE ON 61,KF
C     FORMAT(* THIS  TURT FINISHED, NO,ACTIVE= *, I8)
      IF KPA NE 0, GO TO 21
      LET TATA = TIME - TAB
      CALL PROF(-1,,TATA)
   21 DESTROY AR
      IF KF EQ 0 , GO TO 30
      RETURN
C
C   FINISH OF SIMULATION ,,,,STATISTICS  ETC,,
C
   30 LET XZ = TIME - TAB
      WRITE ON 61,TIME ,XZ
      FORMAT(*0 END OF SIMULATION,,,*,2M4,3,3)
      LET KK = 0
      LET AN(I) = 0 ,FOR I =(1)(10)
      WRITE ON 61
      FORMAT(* XY COORDINATE DESTROYED ,,,NO, OF SETS DESTROYED*)
      LET KNL = 0
      DO,FOR EACH I IN QC
      LET IYX = NSD(I)
C   IF KOUT = 0.,,THEN PLOT
      IF KOUT EQ 0 , CALL OUT(IRNK(I),IYX)
      IF IYX EQ 0 ,GO TO 80
C   ARRAY AN GIVES AGGREGATES OF THE NUMBER OF SETS DESTROYED ON
C     INDIVIDUAL SITES,,,,,
      LET AN(IYX) = AN(IYX) + 1,0
      WRITE ON 61 ,IRNK(I),ISTAT(I),IYX
      FORMAT(* *,I9,I7,S4,I8)
   80 LET KNL = KNL + IYX + 1
      IF ISTAT(I) EQ 3  ,LET KNL = KNL - 1
      LET KK = KK + 1


      LOOP
      LET XY = 0,0
      WRITE ON 61,AN(1),AN(2),AN(3),AN(4)
     C,AN(5),AN(6),AN(7),AN(8),AN(9),AN(10)
      FORMAT(*  VALUES OF AN(I)=*,10D4,0)
      LET XY = XY + AN(I) ,FOR I = (1)(10)
      LET KY4 = KNL - ITSD
      LET XY4 = KY4
      LET XY4 = XY4/FKF
      WRITE ON 61 , KY4,XY4
      FORMAT(*0 NO, OF SETS SURVIVING = *,I8,* ,  NESTING EFFICIENCY = *
     C,D5,4)
      LET XIOC = IOC
      LET P = ITSD
      LET FKNL = KNL
      LET COUNT = KK
      LET PDES =  P / COUNT
      LET POCC= XIOC/COUNT
      LET PDCLU= P/FKNL
      LET XY2  = P/XY
      WRITE ON 61,ITSD,PDES,POCC
      FORMAT(* TOTAL SETS DESTROYED = *,I8,*   PROBY OF SITE- DESTRUCTIO
```

```
      NN = *,D2,7,* RATIO OF OCCUPIED SITES =*,D2,6)
      WRITE ON 61,XY2,KNL,PDCLU
      FORMAT(* COND, MEAN *,
     LD2,6,* TOTAL LAYS *,I6,* PROBY OF CLUTCH DESTRUCTION *,D2,6)
C   WAVE DATA  FOLLOWS
      DO , FOR II=(1)(LNG)
      WRITE ON 61,II,KFL(II)
      FORMAT(* WAVE NO, *,I3,* ,CARRIED  *,I4,*  TURTLES,*)
      LOOP
C  PRINT THE TOTAL NUMBER OF TURTLES IN EACH WAVE,
      WRITE ON 61,KK
      FORMAT(* TOTAL NO, OF NESTS,,,*,I8)
      LET NEXDAY = (TIME + 10,0)/10,0
      LET HOUR = NEXDAY * 10
      CREATE FIR
      CAUSE FIR AT HOUR
      RETURN
      END
      SUBROUTINE  SPOT (IFL)
C       POSITIONS TURTLE  IN LAYING ZONE
C
C   SETS ISTAT(COR) TO 1 FOR OCCUPANCY
C   SETS SITE ADDRESS TO ID(AR)
C   UPDATES OCCUPANCY COUNTER IOC
C   UPDATES NO OF SITES DESTROYED NSD
C   UPDATES TOTAL SETS DESTROYED ITSD
C
C
C     WRITE ON 61
C     FORMAT(*   IN  SPOT *)
   25 LET IX=30.*SQRTF(RANDM)
      LET IY=100,*RANDM
      LET  IR=10000*IX +IY
C     WRITE ON 61,IX,IY,IR
C     FORMAT(* IX ,IY ,IR*, 3I9)
      FIND FIRST , FOR EACH I IN QC,              WITH  IR  EQ  IRNK(I),
     XWHERE IJ  SATISFIES THIS, IF NONE GO TO 50
      IF ISTAT(IJ) EQ 1, GO TO 20
      IF ISTAT(IJ) EQ 2, GO TO 30
C   OCCUPYING OLD EMPTY NEST
   40 STORE IJ IN ID(IFL)


C     WRITE ON 61,ISTAT(IJ), IJ,ABORT(IFL),IFL
C     FORMAT(* ISTAT,IJ,ABORT,IFL *, 4I8)
      STORE 1 IN ISTAT(IJ)
      RETURN
C   IF OCCUPIED SELECT ANOTHER RANDOM SITE
   20 LET LO = LO
C     WRITE ON 61
C     FORMAT(* OCCUPIED *)
      LET IOC = IOC + 1
      GO TO 25
   30 LET ITSD = ITSD + 1
C     WRITE ON 61
C     FORMAT(* LAYING OVER *)
      LET NSD(IJ)=NSD(IJ)+1
      GO TO 40
   50 CREATE COR
C  CREATE A NEW LAYING SITE
      STORE IR IN IRNK(COR)
      STORE 1 IN ISTAT(COR)
      STORE 0 IN NSD(COR)
      FILE COR IN QC
      STORE COR IN ID (IFL)
      IF ABORT(IFL) GR 0 , STORE 3 IN NTYP(IFL)
      RETURN
      END
```

```
      SUBROUTINE SCRT(IAR)
C   CHECKS FOR N-TH TEST AND,,
C   SETS ABORT(IAR) INDEX AS FOLLOWS,,
C         0   - IF TURTLE LAYS
C         1   - IF TURTLE DIGS AND LEAVES
C         N + 1 - IF TURTLE DIGS UNSUCCESSFULLY AND GOES ON TO N+1 -TH TEST
C
C
C   RESULTING ACTION DEPENDS ON A RANDOM NO, FALLING IN THESE RANGES,,,
C
C   RANDM < PLA ,,,LAY NOW
C   PLA < RANDM < PLA+STA ,,,DIG AND TRY AGAIN
C   RANDM > PLA+STA ,,, DIG AND LEAVE
C
      LET XYU = RANDM
      LET XYZ = PLA(ABORT(IAR))
      LET XYI = XYZ + PST(ABORT(IAR))
      IF XYU GR XYI , GO TO 10
      IF XYU GR XYZ , GO TO 12
      STORE 0 IN ABORT(IAR)
      GO TO 20
   10 STORE 1 IN ABORT(IAR)
      GO TO 20
   12 LET ABORT(IAR) = ABORT(IAR) + 1
   20 LET LO = LO
C 20  WRITE ON 61,IAR,XYU,XYI,ABORT(IAR)
C     FORMAT(*  IN SCRT ,,,IAR,RAN,SUM,ABORT *,I8,2D6,6,I8)
      RETURN
      END
C
C
C       FORTRAN SUBROUTINES
C
C
      SUBROUTINE OUT(I,J)
C   PLOTS NESTING PATTERN ,,,,,,
C   THE NO, PLACED ON THE GRAPH INDICATES THE NO, OF SETS
C   DESTROYED AT THAT LOCATION,


      IN = I/10000
      W = I - 10000*IN
      V = IN
      CALL PLOT(V, W,3)
      INK = SHIFT(J,42)
      CALL TEXT(INK,1,1)
      RETURN
      END
      SUBROUTINE PROF(X,R)
C   MAINTAINS A PLOT OF THE TOTAL NO, OF TURTLES ACTIVE
C   IN THE MODEL AT A GIVEN TIME
      DATA(WHERE = 0,0)
      WHERE = WHERE + X
      CALL PLOT(R,WHERE,4,4)
      RETURN
      END
C
C
C        INITIALIZATION CARDS,,,,,,,
C
C
C     NUMBER IN COLUMNS 3-4 REFERS TO PERAMNENT ATTRIBUTES
C     AS SPECIFIED ON DEFINITION CARDS,
C
123456789,123456789,123456789,123456789,123456789,123456789,123456789,123456789,
C
1        26
```

```
    1         R                                                U S I   4(D1,6,I2)
 0,025000 10,030000 20,032000 30,035000 4                                          C
 0,040     50,045    60,05     70,063    8                                         C
 0,08      90,1     100,1     110,08    12                                         C
 0,063    130,05    140,045   150,04    16                                         C
 0,035    170,032   180,03    190,025   20                                         4
    2         R                                                U   L C 6(D1,4,D2,2)
 0,0000-2,000,0057-1,750,0227-1,500,0511-1,250,0909-1,000,1420-0,75               C
 0,2045-0,500,2784-0,250,3636 0,000,4513 0,250,5325 0,500,6071 0,75               C
 0,6753 1,000,7370 1,250,7922 1,500,8409 1,750,8831 2,000,9188 2,25               C
 0,9481 2,500,9708 2,750,9870 3,000,9968 3,251,0000 3,50                          5
    3         R                                                U S I   5(D1,6,I1)
 0,40444410,27111120,24592630,06370440,0148155                                    5
    4         R                                                U S I   4(D1,6,I2)
 0,001054 80,004215 90,007376100,01369911                                          C
 0,035827120,130664130,219178140,22760815                                          C
 0,170706160,097998170,061117180,01369919                                          C
 0,010537200,00632221                                                              2
    5    8  0   Z
    9         R                                                           10
   10       1   Z   10     9
   11           Z
   12           Z
   13           Z
   14         R                                                           0,25
   15         R                                                                 0
   16         R                                                           35
   17       1 R      35   16                                        10(D1,5)
 0,1     0,2286  0,3571  0,4857  0,6143  0,7429  0,8     0,8     0,8     0,8       C
 0,8     0,8     0,8     0,8     0,8     0,8     0,8     0,8     0,8     0,8       C
 0,8     ,764710,705880,647060,588240,529410,470590,411770,352940,29412 C
 0,235290,176470,117650,058820,0000                                                5
   18           Z
   19         R                                                           4
   20       1 R       4   19                                         4(D1,5)
 0,660940,517860,444440,75000                                                      4


   21       1 R       4   19                                         4(D1,5)
 0,240340,160710,444440,25000                                                      4
   22         R                                                      1,0
   23         R                                                      0
   24           Z
   25   26  0   Z

   1    0
   15
    1,0              0          0  0   0,0  0
   -1,0              0          0
 C
 123456789,123456789,123456789,123456789,123456789,123456789,123456789,123456789,
 C
```

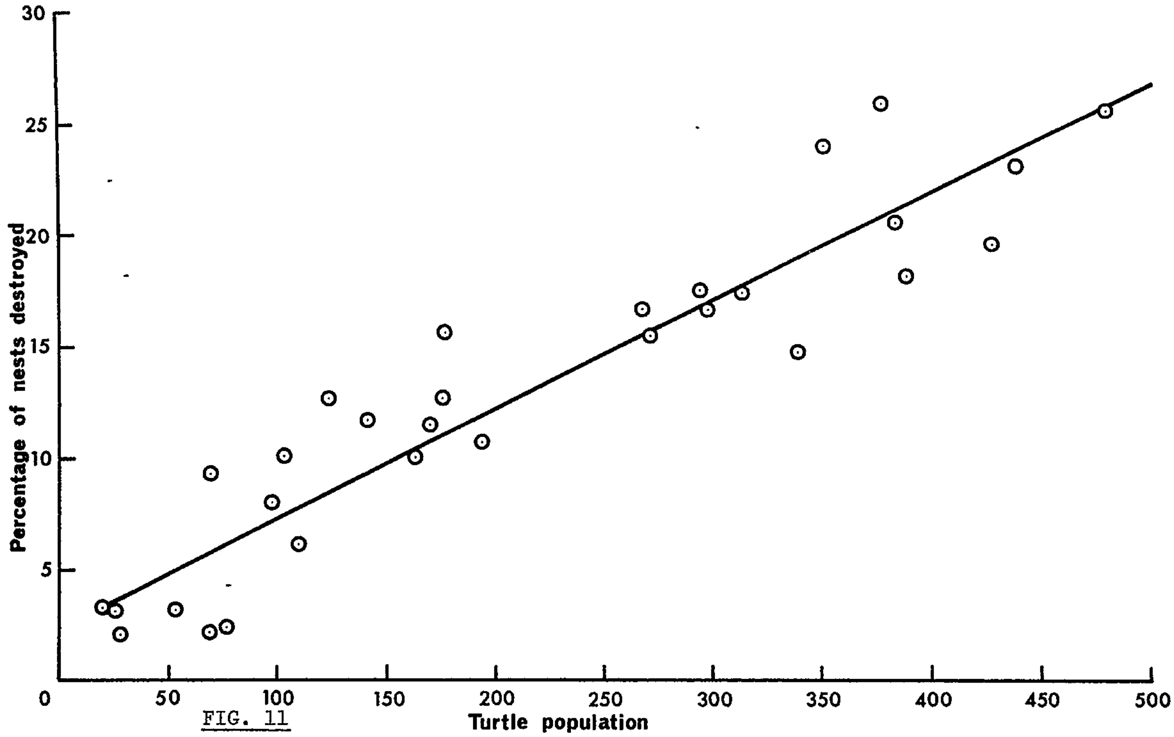


FIG. 11

Percentage of nests destroyed against total turtle population (linear regression).

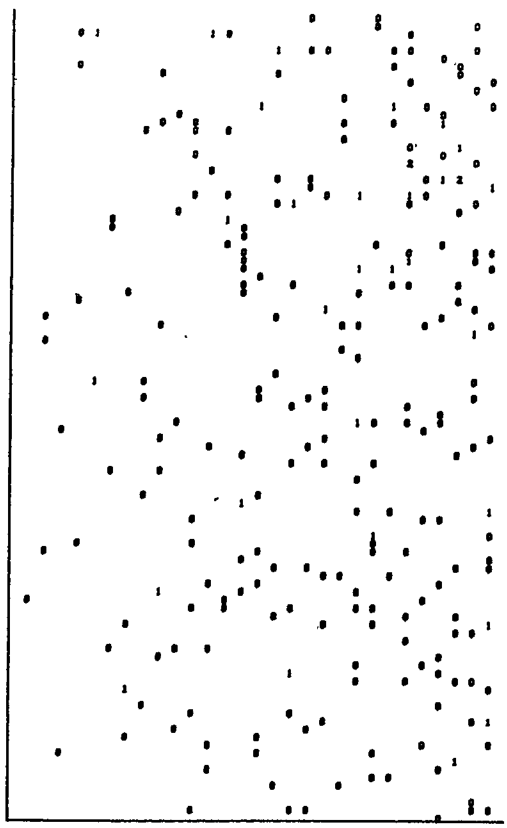


FIG. 12

Nesting pattern on beach site. Numbers indicate nests destroyed.

FIG. 13

Active turtles versus time.