NN 6024
422

REPRINTED FROM:
OECOLOGIA (BERL.) 4 (1970) 113-132

THE UTILITY OF A DIGITAL SIMULATION LANGUAGE FOR ECOLOGICAL MODELING

R. D. BRENNAN, C. T. DE WIT, W. A. WILLIAMS
and E. V. QUATTRIN

Bibliotheek
der
Landbouw Hogeschool
WAGENINGEN

Instituut voor Biologisch en Scheikundig Onderzoek van Landbouwgewassen, Wageningen
Mededeling 422

Oecologia (Berl.) 4, 113—132 (1970)

The Utility of a Digital Simulation Language for Ecological Modeling

R. D. Brennan, C. T. de Wit, W. A. Williams, and E. V. Quattrin

IBM Corporation, Palo Alto, California, U.S.A.; University of Wageningen, Wageningen, The Netherlands; University of California, Davis, California, U.S.A.; and Development Resources Corporation, Sacramento, California, U.S.A.

Received September 9, 1969

Summary. Dynamic modeling of ecological phenomena has been greatly facilitated by the recent development of continuous system simulator programs. This paper illustrates the application of one of these programs, S/360 Continuous System Modeling Program (S/360 CSMP), to four systems of graduated complexity. The first is a two species system, with one feeding on the other, using differential equations with constant coefficients. The second and third systems involve two competing plant species in which the coefficients of the differential equations are varying with time. The final example considers the management of a postulated buffalo herd in which the dynamics of the herd population and composition by sex and age is combined with various strategies to control its size and to optimize buffalo production.

Recently developed "continuous system simulator" languages make it possible for the ecologist to construct digital computer models of biological systems with a minimal input of effort and mathematical expertise. These languages permit him to concentrate on the biological phenomenon of interest rather than the intricacies of numerical analysis and digital computer programming (Brennan, 1968; Yates *et al.*, 1968). This paper provides an introduction to S/360 CSMP[1], perhaps the most powerful of these languages, and illustrates its application to several simple ecological systems.

Theory relating to the number and kinds of plants and animals found in nature and their order and interactions has developed considerable sophistication in recent years. Slobodkin (1961) has stated that such a general theory is a solvable problem and that the procedures involved in its solution are available at least in principle. In general, the solution must be sought in the mathematical expression of relationships. The most convenient mode for defining such processes is in systems of algebraic or differential equations. But most processes in nature are essentially non-linear, and the analytical solution of all

1 "S/360 CSMP", or more properly, "S/360 Continuous System Modeling Program", is a continuous system simulator developed by IBM for its System/360 Computer System operating under OS/360 on systems with 128K core memory or larger.

but the most trivial system of non-linear algebraic or differential systems of equations is impossible or at least extremely cumbersome (Franks, 1967). Watt (1966) and Patten (1966) among others have suggested using systems analysis techniques with general purpose computer languages (e.g., FORTRAN, ALGOL, PL/1). This makes possible the solution of many problems not amenable to analytical resolution, but requires considerable mathematical facility on the part of the researcher and usually the services of a professional programmer to handle the many details of program writing and debugging.

The utilization of a simulation language for work in this area has received little attention as yet by ecologists, although their usefulness has been indicated by Garfinkel and Sack (1964), Paulik and Greenough (1966), and de Wit and Brouwer (1968). Recently-developed simulation languages have several advantages for those with a limited mathematical and computer background. First, they are relatively simple to learn. Rather than worrying about the details of programming (which have been taken care of by the language developer), one can concentrate on relational concepts and problem conceptualization. Second, because of the dynamic aspects of the modeling permitted, one can observe relationships not apparent from steady state solutions, and thus can ascertain inadequacies and ambiguities in the postulated model. This may lead directly to the formulation of incisive experiments.

Description of S/360 CSMP

S/360 CSMP is a continuous system simulation language which enables the user to define the structure of a model starting from either a relational block diagram or a differential-equation representation of the model system (IBM, 1968). This language feature is an accommodation to the observation that people are of distinct cognitive types: those who conceptualize dynamic phenomena in pictures and those who do so via mathematical notation. To the ecologist this means he may use such simulation languages in whatever manner seems most convenient for the particular investigation. For example, he may view exponential smoothing as a process provided by a functional block into which he sends certain signals and from which is obtained an output, or he may consider it a mathematical operator of specific difference or differential equation form. The distinction is entirely conceptual; in each case the operation must be specified with precision. While simulation languages facilitate model building and verification, they contain no magic to transform muddled thinking into scientific investigation.

The program provides a complement of 34 functional blocks (also called functions) for modeling a continuous system. These functions

include such conventional analog computer components as integrators and relays plus many special purpose functions such as delay time, zero-order hold, dead space, and limiter functions. This complement is augmented by FORTRAN library functions such as sine, logarithm, and square root.

In addition, the user can define other functional blocks specially suited to his own system. This definition can be accomplished either through FORTRAN subprograms or, more simply, through "MACRO's" or "PROCEDURE's" which permit individual functions to be combined into larger functional blocks. By combining these functional blocks with FORTRAN algebraic and logic statements, the user may handle very complex non-linear and time-variant models.

Two important features of the program are statement sequencing and a choice of integration methods. With few exceptions, structure statements may be written in any order and are automatically sorted by the system to establish the correct order of information flow. Centralized integration is used to ensure that all integrator outputs are computed simultaneously at the end of each time interval. A choice may be made among the fifth-order Milne predictor-corrector, fourth-order Runge-Kutta, Simpson's, second-order Adam's, trapezoidal, and rectangular integration methods. The first two methods allow the integration interval to be automatically adjusted by the program to meet specified error limits.

A fixed format for data output is provided at selected increments of the independent variable for all output options, thereby freeing the user from the details of programming format. Output options include the printing of tables and print-plotting in graphs of the values for the variables used.

A simulation problem is programmed for solution by preparing the following three types of statements on punched cards or other input device:

1. Structure statements. These describe the functional relationships among the variables of the model; taken together, they define the system to be simulated.

2. Data statements. These assign numerical values to the initial conditions, parameters, and table entries used in the problem.

3. Control statements. These specify options relating to the translation, execution, and output phases of S/360 CSMP.

Two Species, One Feeding Upon The Other

In the first illustration we will use the classical two species system first described mathematically by Lotka (1925, reprinted 1956) and Volterra (1928, reprinted 1931). For a feasible analytical solution, the

```
TITLE    DYNAMIC INTERACTION IN A TWO SPECIES SYSTEM
INCON       S0 = 100.0   ,   L0 = 10.0
PARAMETERS K1 = 1.1      ,   K2 = 1.0E-5   ,   K3 = 1.0E-3 ,...
           K4 = 0.9      ,   K5 = 1.0E-4   ,   B = 0.02
            S = INTGRL( S0, K1*S -K2*S*S -K3*S*L    )
            L = INTGRL( L0, B*K3*S*L -K4*L -K5*L*L )
PRTPLOT     L
LABEL    LIMPET POPULATION PER HECTARE - RECOVERY FROM DISASTER
PRTPLOT     S
LABEL    SEAWEED AMOUNT PER HECTARE  - RECOVERY FROM DISASTER
TIMER     FINTIM = 30.0 , OUTDEL = 1.0
END
STOP
```

Fig. 1. S/360 CSMP statements for seaweed-limpet model

coefficients in the differential equations must be constant. A general form the equations can take is shown in two differential equations describing the time-rate of change in the amount of food as seaweed (S) and the number of limpets (L) in a system (Garfinkel, 1967):

$$dS/dT = K1 \cdot S - K2 \cdot S^2 - K3 \cdot S \cdot L.$$

$$dL/dT = B \cdot K3 \cdot S \cdot L - K4 \cdot L - K5 \cdot L^2.$$

The coefficients of proportionality are related to the associated processes as indicated:

$K1 = 1.1$ reproduction of seaweed;

$K2 = 1.0 \times 10^{-5}$ seaweed density factor;

$K3 = 1.0 \times 10^{-3}$ effect of limpet feeding;

$K4 = 0.9$ limpet mortality;

$K5 = 1.0 \times 10^{-4}$ limpet density factor;

$B = 0.02$ limpet birth rate;

with time expressed in days.

For an example, we will assume a disaster has reduced the seaweed to 100 units/hectare and the limpet population to 10 individuals/hectare and that we wish to simulate the population changes during the subsequent 30-day period. A suitable program in S/360 CSMP is shown in Fig. 1.

Initial conditions ($L0 = 10.0$, $S0 = 100.0$) are specified by the statement ICON, and the PARAMETERS are listed in a self-evident manner. The two differential equations are written in simplified integral (INTGRL) form with the initial conditions $S0$ and $L0$ (i.e., constants of integration) first, followed by the differential expressions to be integrated. Next are the output control statements for print-plotting (PRTPLT) and the results of the integrations with appropriate labels (LABEL). Finally, the duration of the simulation (FINTIM = 30.0) and the plotting interval (OUTDEL = 1.0) are specified in days. The program is terminated by END and STOP cards.

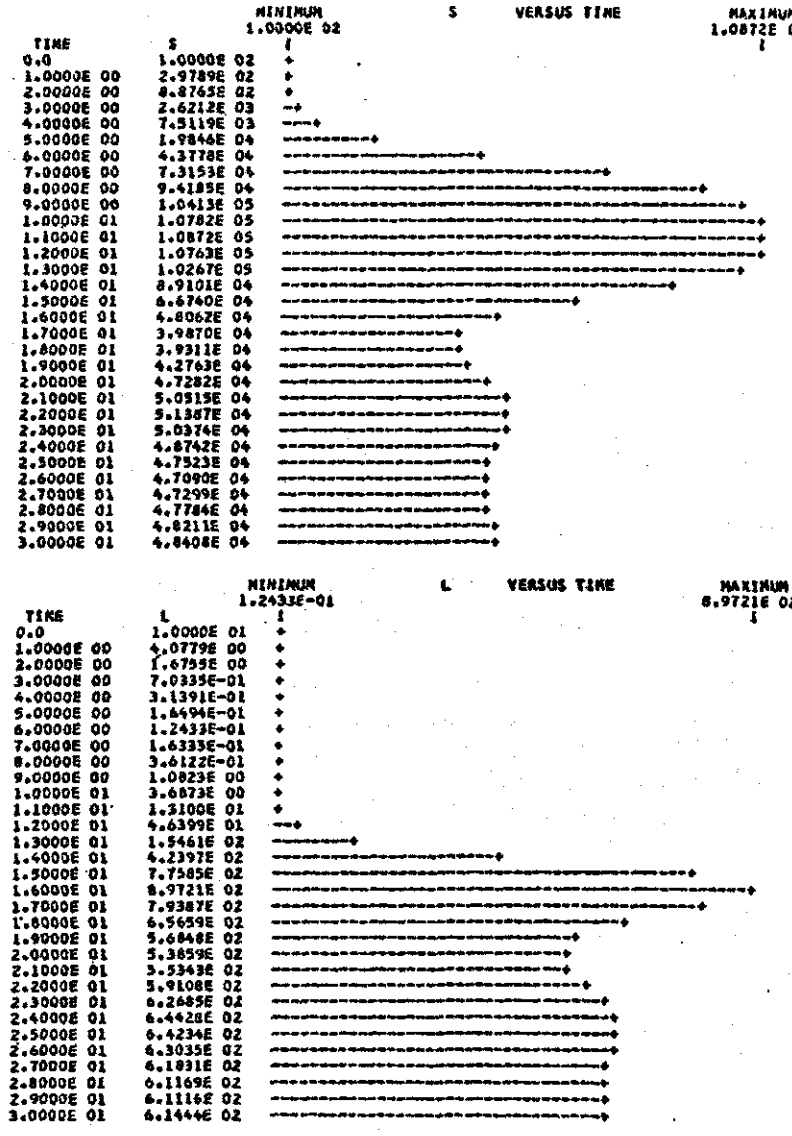

Fig. 2. Sample output (top—seaweed, bottom—limpets)

The resulting output is shown in Fig. 2 with the recovery of the organisms depicted as a function of time with both tabular and graphical data sets for the seaweed (top) and the limpets (bottom). The seaweed recovers first, and the limpet population then surges until it

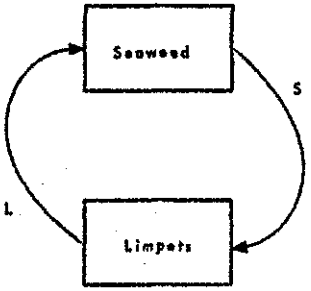

Fig. 3. Life processes as interconnected blocks

limits the seaweed. Cyclical fluctuation follows with decreasing amplitude in time exhibiting the well known behavior of this pair of equations.

Note the simple structure statements required to generate their solution. The two statements representing the integration of the differential equations are in the form

$$Y = \text{INTGRL}\ (IC, X)$$

which states that the output Y is obtained by integrating X, the differential function, with the initial condition that Y at time zero is equal to IC. The name INTGRL defines the operation to be performed on the function X, i.e., integration, and is one of the 34 functional blocks available. Also note the meaning of the operational symbols: (as in FORTRAN) = replaces, + addition, * multiplication, ** exponentiation, / division, and ( ) grouping.

The preceding formulation was based directly upon the differential equation representation of the seaweed-limpet system. Even for a model this simple, many different conceptualizations and formulations are permissible and equally valid — — "de gustibus non est disputandum". To illustrate an alternate approach, let us consider the life processes of the seaweed and the limpets as separate blocks, each involving the appropriate dynamics, and with the blocks connected as in Fig. 3 to emphasize their coupling or interaction.

For a more complex system, one might model the two blocks separately; but since both have similar factors — reproduction/mortality, density, and coupling — one can, in fact, define a general MACRO which is representative of both life processes. For example, the following set of statements define such a MACRO which has been arbitrarily named CYCLE:

```
MACRO        P1 = CYCLE (IC, A, B, C, P2)
          DP1DT = A*P1 + B*P1*P1 + C*P1*P2
             P1 = INTGRL (IC, DP1DT)
ENDMAC
```

The output of block CYCLE is $P1$ (population No.1); the initial value of $P1$ at time zero is assigned the quantity IC. The derivative of $P1$ involves linear and squared terms in $P1$ as well as a coupling term involving $P2$ (population No.2). Thus, the CYCLE block conceptually has a single input, $P2$, and associated parameters IC, A, B, and C. By invoking the MACRO twice, each time with the appropriate input and parameters, one may represent both processes as follows:

L=CYCLE ($L0$, $-K4$, $-K5$, $B3*K3$, S)

S=CYCLE ($S0$, $K1$, $-K2$, $K3$, L)

The S/360 CSMP translator automatically expands each of the invocations of the MACRO according to the "pattern" specified in the definition of CYCLE.

In the above problem an implicit solution is possible by analytical methods, and an explicit solution is more difficult. If some of the coefficients vary with time or any other element, analytical solution becomes impossible (Volterra, 1928). However, the use of an appropriate simulation language makes the solution of these differential equations with variable coefficients not only possible, but simple, as will be demonstrated in the next example involving competition in a mixture of two plant species.

Two Plant Species Competing for the Same Space

With but minor modification, the Lotka-Volterra equations may be used to represent the competition between plant species in a mixed stand. In contrast to the seaweed-limpets model in which the two populations were the variables of interest, in a mixed stand of an agricultural crop the number of plants of each species is determined at planting time. The competition between crop species may be expressed in terms of "relative space", RS, a dimensionless variable which characterizes the effect of crowding for available root and foliage space, nutrients, sunlight, and associated factors. Actual production of dry matter, if desired, can be obtained from the product of "relative space" and the maximum possible yield for very dense monoculture planting. It is important to realize that all three quantities are functions of time.

For example, barley and oats can be planted as a mixed stand. Simulation provides a convenient means for studying their competition. The differential equations, in which RS_b represents "relative space" for barley and RS_o represents that for oats, are as follows:

$$\frac{dRS_b}{dt} = RGR_b\,(RS_b - RS_b^2 - RS_b\,RS_o)$$

$$\frac{dRS_o}{dt} = RGR_o\,(RS_o - RS_o^2 - RS_b\,RS_o)$$

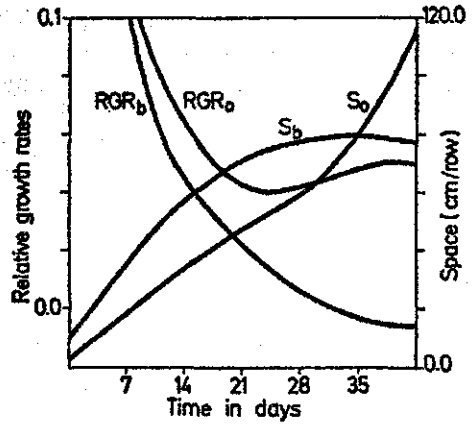

Fig. 4. Space (S_b and S_o) occupied by barley and oats versus time of harvesting and their relative growth rates (RGR_b and RGR_o) Note: $RGR = \frac{1}{S}\frac{dS}{dt}$

These have the same form as the seaweed-limpet example except for one important distinction — the coefficients RGR_b (relative growth rate for barley) and RGR_o (relative growth rate for oats) are both empirical functions of time. Both functions represent relative plant growth rates in absence of competition and have been determined from experimental plantings with very low seed densities. Fig. 4 illustrates the absolute growth, as well as the relative growth rates derived therefrom, for barley and oats harvested at intervals (Baeumer and de Wit, 1968). Note that the barley grows rapidly at first, then levels off; the reverse is true for the oats — its growth curve is concave upwards. Hence, if these two species are grown together, a disproportionate share of the available "space" is occupied by the barley at an early stage, and by the time the oats comes to claim its share there is little remaining.

Empirical functions such as the relative growth rates are readily modeled with a continuous system simulation language. To illustrate, the complete program to simulate the barley-oats competition is shown in Fig. 5. The data representing relative growth rate for the barley (RGRB) is specified as follows:

```
FUNCTION RGRB = (0.0, 0.4286), ...
                (7.0, 0.1071), ...
                etc.
```

Each couplet represents a data pair: the first, the value of the independent variable; the second the corresponding dependent variable. Thus, the second pair above gives the relative growth rate of the crop on the 7th day. These statements associated with the label

```
TITLE    BARLEY AND OATS -- COMPETING FOR THE SAME SPACE
INITIAL
                RSOI  = 3.0 * DO
                RSBI  = 9.0 * DB

DYNAMIC
*         EQUATIONS FOR OATS
                RSO   = INTGRL( RSOI , RATEO )
              RATEO   = RSO * ( 1. -RSO -RSB ) * NLFGEN( RGRO, TIME )
*         EQUATIONS FOR BARLEY
                RSB   = INTGRL( RSBI , RATEB )
              RATEB   = RSB * ( 1. -RSO -RSB ) * NLFGEN( RGRB , TIME )
FUNCTION    RGRB    = ( 0.0 ,  0.4286 ) ,...
                      ( 7.0 ,  0.1071 ) ,...
                      ( 14.0 ,  0.0441 ) ,...
                      ( 21.0 ,  0.0225 ) ,...
                      ( 28.0 ,  0.0064 ) ,...
                      ( 35.0 , -0.0036 ) ,...
                      ( 42.0 , -0.0065 )
FUNCTION    RGRO    = ( 0.0 ,  0.7143 ) ,...
                      ( 7.0 ,  0.1190 ) ,...
                      ( 14.0 ,  0.0634 ) ,...
                      ( 21.0 ,  0.0431 ) ,...
                      ( 28.0 ,  0.0242 ) ,...
                      ( 35.0 ,  0.0511 ) ,...
                      ( 42.0 ,  0.0491 )
*         DATA FROM BAEUMER AND DE WIT 1968

METHOD    MILNE
PRINT     RSB, RSO, DELT
TIMER       FINTIM = 42.0  ,  PRDEL = 7.0
*         DENSITY OF SOWING IN ROW/CM
PARAMETER      DB  = 0.02  ,    DO = 0.02
END

PARAMETER      DB  = 0.04  ,    DO = 0.04
END
STOP
```

Fig. 5. S/360 CSMP statements for barley-oats competition

"FUNCTION" merely enter the data, they do not specify its use. For example, to generate a time function with simple linear interpolation between these data points, one might use the statement

BARLEY = AFGEN (RGRB, TIME)

where AFGEN is one of the standard functional blocks or elements of the S/360 CSMP language. If instead one wished a quadratic interpolation between these data points, it is only necessary to use instead the standard element NLFGEN rather than AFGEN.

Fig. 5 includes several other features not used in the seaweed-limpet simulation. Note that a "METHOD" statement is used to specify the desired integration algorithm — here a variable-step Milne 5th-order predictor-corrector method. If a method is not specified, the program automatically chooses a variable-step Runga Kutta method which usually gives excellent results. The first occurrence of the "END" statement signals the completion of the model description and the para-

BARLEY AND OATS, -- COMPETING FOR THE SAME SPACE

TIME	RSB	RSO	DELT
0.0	1.8000E-01	6.0000E-02	4.3750E-01
7.0000E 00	4.6805E-01	2.7305E-01	4.3750E-01
1.4000E 01	5.0987E-01	2.9388E-01	8.7500E-01
2.1000E 01	5.2911E-01	3.1270E-01	1.7500E 00
2.8000E 01	5.3690E-01	3.2377E-01	1.7500E 00
3.5000E 01	5.3740E-01	3.3417E-01	1.7500E 00
4.2000E 01	5.3485E-01	3.4948E-01	1.7500E 00

Fig. 6. PRINT output for barley-oats competition

meters for the first run. Note that following the first use of "END" there is another parameter specification for DB and DO, the planting densities for the two crops. By this means one may order a lengthy series of simulation runs, varying parameters and control variables from run to run. It is perhaps worth noting that the S/360 CSMP data and control statements are executed "interpretively"; thus given some convenient terminal device, it is possible to achieve effective man-machine interaction while experimenting with a simulation.

For this example simple tabular or "PRINT" output was selected rather than the "print-plot" shown in Fig. 2. A portion of this output is shown in Fig. 6. Although both species were sown in a 1:1 ratio, it appears that barley obtains a much larger relative space than the oats — because it uses much of the space while the oats are yet growing slowly. Comparison with actual competition experiments (Baeumer and deWit, 1968) shows that the assumptions made regarding the interference between species are valid. Fig. 6 also shows the variation of the integration step DELT during the simulation run; since a variable-step integration method was selected, the program automatically adjusts DELT to satisfy relative and absolute accuracy requirements (which could have been explicitly specified, if the default values routinely used by the program were not satisfactory).

Size Distribution of Plants in a Stand

After sowing seeds in a field, some may germinate in one day, more after two days, the bulk perhaps between the 5th and the 10th day, and some others still later. The plants from the earlier germinating seeds claim most of the available space so that little is left for those plants resulting from later germination. This may be an important reason why plants in a cultivated crop often have great differences in size. This simulation study was initiated to study this effect; except for time of germination, it is assumed here that all plants are governed by the same growth factors.

```
TITLE   PLANT SIZE DISTRIBUTION IN A CROP DUE TO GERMINATION TIMES
TITLE       S IN KG/HA,  S IN HA/KG, TIME IN DAYS
STORAGE   RFGM(15)
TABLE     RFGM(1-15) = .048,5*.095,.084,.072,.067,.058,.053, ...
                       .048,.038,.033,.024

MACRO            RS  = EMCL( N, FACTOR )
 PROCEDURE  RATE,RSIC,EN = TEST( N )
          IF( TIME ) 1,1,2
       1              RSIC  = S * B * RFGM(N)
                      EN = N
       2  IF ( TIME - EN ) 3,4,4
       3              RATE  = 0.0
          GO TO 5
       4              RATE  = NLRGEN( RGR, TIME - EN )
       5  CONTINUE
 ENDPRO
                  RS  = RSIC + INTGRL( 0.0, RATE * RS * FACTOR )
ENDMAC
INITIAL
                   S  = 0.002
                   B  = 20.0
DYNAMIC
*    RS1 TO RS15 ARE THE RELATIVE SPACES OCCUPIED BY THE PLANTS
*      IN EACH EMERGENCE CLASS -- THEIR SUM IS SRS
                RS1   = EMCL(  1, FACTOR )
                RS2   = EMCL(  2, FACTOR )
                RS3   = EMCL(  3, FACTOR )
                RS4   = EMCL(  4, FACTOR )
                RS5   = EMCL(  5, FACTOR )
                RS6   = EMCL(  6, FACTOR )
                RS7   = EMCL(  7, FACTOR )
                RS8   = EMCL(  8, FACTOR )
                RS9   = EMCL(  9, FACTOR )
                RS10  = EMCL( 10, FACTOR )
                RS11  = EMCL( 11, FACTOR )
                RS12  = EMCL( 12, FACTOR )
                RS13  = EMCL( 13, FACTOR )
                RS14  = EMCL( 14, FACTOR )
                RS15  = EMCL( 15, FACTOR )
                SRS   = RS1 + RS2 + RS3 + RS4 + RS5 + RS6 + RS7 + ...
                        RS8 + RS9 + RS10 + RS11 + RS12 + RS13  + ...
                        RS14 + RS15
             FACTOR  = 1.0 - SRS

FUNCTION  RGR = (  0.0 ,   0.55000 ) ,...
                (  7.0 ,   0.11346 ) ,...
                ( 14.0 ,   0.06322 ) ,...
                ( 21.0 ,   0.04262 ) ,...
                ( 28.0 ,   0.03254 ) ,...
                ( 35.0 ,   0.05528 ) ,...
                ( 42.0 ,   0.03249 ) ,...
                ( 49.0 ,   0.04759 ) ,...
                ( 56.0 ,   0.05747 ) ,...
                ( 63.0 ,   0.06248 ) ,...
                ( 70.0 ,   0.05606 )
LABEL  PLANT SIZE DISTRIBUTION IN A CROP DUE TO GERMINATION TIMES
PREPARE  RS1, RS4, RS7, RS10, RS13, RS15, SRS
METHOD   ADAMS
TIMER   DELT = 1.0 ,  FINTIM = 70.0 ,  OUTDEL = 1.0
END
STOP
ENDJOB
```

Fig. 7. S/360 CSMP statements for size distribution of plants in a stand

The model assumes 15 emergence classes, representing the plants resulting from germination on each of 15 successive days after sowing. Since each class must obey the same growth function, a "MACRO"

has been defined to specify the relationship; it is then invoked 15 times to represent each of the classes. The relative frequency of germination for each class is communicated to the respective use of the MACRO by means of a data array. The "STORAGE" statement shown in Fig. 7 informs the S/360 CSMP translator that the symbolic name RFGM represents an array of 15 elements. The associated "TABLE" statement is used to assign relative frequency distribution values to each of the 15 classes. Thus, the statement

TABLE RFGM (1—15) = 0.048, 5 * 0.095, 0.086, ...

specifies that 4.8% will germinate the first day, 9.5% the next 5 days, 8.6% on the seventh day, etc.

The MACRO shown in Fig. 7, while representing more complex use of the simulation language than required for the preceeding examples, still shows only modest degree of sophistication compared to what a knowledgeable modeler might attempt. The statements between "MACRO" and "ENDMAC" specify the structural pattern which the S/360 CSMP translator should follow each time the MACRO is subsequently invoked in the DYNAMIC portion of the model. Note that within the definition of the MACRO is contained a set of 10 ordinary FORTRAN statements which collectively comprise a "PROCEDURE" which is conceptually a single functional element. This particular element is defined as having three outputs (RATE, RSIC, and EN) and a single input (N). At the beginning of each run, that is, when TIME equals zero, the initial value for the Nth class is computed as the product of S, D, and RFGM(N) where S is the initial relative space for the particular species, D is the density of sowing, and RFGM(N) is the relative frequency of germination for the class. Until TIME equals N days, the RATE of growth is set to zero; thereafter it is determined by a function generator using the FUNCTION data array named RGR.

Except for two statements required to compute SRS and FACTOR, the DYNAMIC segment of the model's structure shown in Fig. 7 consists of "invocations" of the MACRO "EMCL" previously defined. It is this portion of the model, describing the dynamics of the model, which the program must exercise in conjunction with the selected integration subroutine at each discrete advance in TIME. While the DYNAMIC segment appears concise, it should be remembered that each of the 15 invocations of "ENCL" results in generation of a set of FORTRAN statements corresponding to those specified when initially defining this MACRO; thus, the subroutine actually generated by the S/360 CSMP translator may be quite lengthy. Since MACROs may actually be embedded within MACROs, themselves perhaps within other MACROs, this feature of continuous system simulation languages offers significant

power to the modeler; this is especially true with biological systems which characteristically are hierarchical in nature.

For this simulation study, the PREPARE output option was selected. This causes the variables included with the statement — here RS1, RS4, RS7, etc. — to be included in a magnetic tape which includes their values every OUTDEL time units. This tape can then be used for preparation of an X-Y plot of the variables on a Calcomp plotter.

Buffalo Herd Management

Most changes in ecological systems occur continuously with time. However, when demographic aspects of systems are under consideration, it is clear that discrete events must be adequately accounted for in the modeling process. Although S/360 CSMP is a continuous system simulator, this type of problem may be conveniently handled by means of this language as illustrated here.

The dynamics of buffalo herd population and composition is combined with various strategies to control its size and to optimize buffalo production. Population dynamics, when approached from the more traditional stochastic approach, involves complex statistical formulations that are difficult to apply (Bartlett, 1960). Our simulation of a herd of buffalo shows how situations containing such complex relationships may be dealt with rather simply.

The program elaborates on Watt's (1968) buffalo example and uses data from buffalo (American bison) studies by Fuller (1962), McHugh (1958), and Roe (1951). The objectives of this simulation are to generate a realistic life process and to view the results of various management practices on the population.

The buffalo example was chosen for it typifies questions that are being asked as wildlife areas become more restrictive and the demand for a representive natural wildlife state increases. Our hypothetical wildlife area is restricted to an area on which only a herd of buffalo composed of a typical cross-section of ages and sexes is maintained. All the other animals that would naturally share the range could be represented also, but since this is only a simple illustration we concerned ourselves with the dynamics of the buffalo only.

The first objective was to simulate the population of buffalo over time while weather and the carrying capacity of the range resource are varied cyclically according to the season, but in a random manner from year to year. The condition of the animals is a specified function of the carrying capacity of the range. Fecundity is introduced as a function of animal condition and age and the proportion of adult males to adult females. The mortality is calculated from the condition and age factors. Within the model various harvesting strategies are

```
TITLE     SIMULATION STUDY OF HERD MANAGEMENT POLICY
*    EACH SEX DIVIDED INTO CALVES, YEARLINGS, AND 5 FIVE YEAR AGE GROUPS
*    PLUS A SINGLE 'AGED' GROUP CONTAINING MALES AND FEMALES OF 27+ YEARS

*    AVCAP = AVERAGE CAPACITY OF RANGE
*    CAP   = CAPACITY THAT CAN BE CARRIED IN A SPECIFIC MONTH
*    COND  = CONDITION OF THE ANIMALS
*    FERTA = FERTILITY RATE DUE TO AGE
*    FERTC = FERTILITY FACTOR DUE TO CONDITION
*    FRSHT = FRACTION HARVESTED IN A SPECIFIC AGE GROUP
*    LEVEL = CENSUS LEVEL OF CONTROL IMPOSED BY RANGERS
*    MORTA = MORTALITY RATE IN PERCENT PER MONTH AS A FUNCTION OF AGE
*    MORTC = MORTALITY FACTOR DUE TO CONDITION
*    NBORN = NUMBER BORN
*    NSHOT = NUMBER HARVESTED
*    TOTFF = TOTAL NUMBER OF FERTILE FEMALES
*    TOTFM = TOTAL NUMBER OF FERTILE MALES

FIXED     MONTH
MACRO  TOTAL, TFERTL, OLD, SHOT, SHTYET = GROUP (IC,NIN,ZAP,MORTA,FERTA)
            TOTAL = AMAX1 ( 0.0, N1 + N2 + N3 + N4 + N5)
              OLD = AMAX1 ( 0.0, N5 )
           TFERTL = TOTAL * FERTA * FERTC
     PROCEDURE SHOT, SHTYET, FRSHT = HUNT ( ZAP,  TOTAL )
             SHOT= AMIN1 ( ZAP,  TOTAL * 0.8 )
          SHTYET = ZAP - SHOT
           FRSHT= 0.0
       IF ( TOTAL .GT. 0.0 ) FRSHT = SHOT / TOTAL
     ENDPRO
            MORTAL = AMIN1 ( MORTA * MORTC + FRSHT,  1.0 )
                F1 = ADVNCE * ( 1.0 - MORTAL )
                F2 = AMAX1 ( ADVNCE, MORTAL )
                N1 = INTGRL ( IC, F1 * NIN - F2 * N1 )
                N2= INTGRL ( IC, F1 * N1  - F2 * N2 )
                N3= INTGRL ( IC, F1 * N2  - F2 * N3 )
                N4= INTGRL ( IC, F1 * N3  - F2 * N4 )
                N5= INTGRL ( IC, F1 * N4  - F2 * N5 )
ENDMAC
***** END OF MACRO DEFINITIONS, BEGINNING OF MODEL STRUCTURE
INITIAL
             RATIO = 1.0
               W2 = 1.0
DYNAMIC
            MONTH = 1.0 + AMOD( TIME , 12.0 )

PROCEDURE    FBORN,MBORN,ADVNCE = APRIL( MONTH, TOTFF, RATIO )
       IF( MONTH .NE. 4 )  GO TO 10
            NBORN = TOTFF * AMIN1( 1.0, 0.0*RATIO )
            FBORN = 0.47 * NBORN
            MBORN = 0.53 * NBORN
            ADVNCE= 1.0
       GO TO 20
    10      FBORN = 0.0
            MBORN = 0.0
            ADVNCE= 0.0
    20 CONTINUE
ENDPRO

PROCEDURE     CENSUS, ESTLD,RATIO = JULY(MONTH,CALVES,YRLING,COWS,BULLS,...
                                    AGED,TOTFF,TOTFM,LOAD)
       IF( MONTH .NE. 7 ) GO TO 40
       RANGER TAKES YEARLY CENSUS AND ESTIMATES LOAD ON GRAZING AREA
          ACTUAL = CALVES + YRLING + BULLS + COWS + AGED
          CENSUS = ACTUAL * GAUSS( 13, 1.0, 0.05 )
           ESTLD = LOAD . * GAUSS( 13, 1.0, 0.08 )
           RATIO = TOTFM/TOTFF
       RATIO OF FERTILE BULLS TO COWS DETERMINES IMPREGNATION
    40 CONTINUE
  ENDPRO

PROCEDURE    NSHOOT  = OCTOBE( MONTH, CENSUS, ESTLD )
         IF( MONTH .NE. 10 )  GO TO 70
            NSHOOT  = AMAX1( 0.0, CENSUS - LEVEL )
       GO TO 80
    70      NSHOOT  = 0.0
    80 CONTINUE
  ENDPRO
```

Fig 8 S/360 CSMP t er e ts for buffalo herd mar agemer t st d

```
                CM = INTGRL( 250.0 , MBORN -AMAX1(ADVANCE,0.02*MORTC)*CM)
                CF = INTGRL( 250.0 , FBORN -AMAX1(ADVANCE,0.02*MORTC)*CF)
                YM = INTGRL( 140.0 , ADVNCE*(1.0 - 0.01*MORTC)*CM      ...
                          -AMAX1( ADVNCE, 0.01*MORTC)*YM )
                YF = INTGRL( 140.0 , ADVNCE*(1.0 - 0.01*MORTC)*CF      ...
                          -AMAX1( ADVNCE, 0.01*MORTC)*YF )
ADM1,ADFM1, AGE7M,MSHOT1,SYETM1 = GROUP(110.0 , YM ,SYETF1,0.01,1.0 )
ADF1,ADFF1, AGE7F,FSHOT1, SYETF1 = GROUP(110.0 , YF ,SYETM1, 0.01,1.0)
ADM2,ADFM2, AGE12M,MSHOT2,SYETM2 = GROUP( 90.0 ,AGE7M,SYETF2,0.01,0.90)
ADF2,ADFF2, AGE12F,FSHOT2, SYETF2 = GROUP( 90.0 ,AGE7F,SYETM2,0.01,0.85)
ADM3,ADFM3,AGE17M,MSHOT3,SYETM3 = GROUP( 70.0 ,AGE12M,SYETF3,0.015,0.8)
ADF3,ADFF3, AGE17F,FSHOT3, SYETF3 = GROUP( 70.0 , AGE12F,SYETM3,0.02,0.78)
ADM4,ADFM4,AGE22M,MSHOT4,SYETM4= GROUP( 50.0 ,AGE17M,SYETF4,0.03,0.60)
ADF4,ADFF4,AGE22F,FSHOT4,SYETF4 = GROUP( 50.0 , AGE17F,SYETM3.0.05,0.58)
ADM5,ADFM5,AGE27M,MSHOT5,SYETM5= GROUP( 30.0 , AGE22M,SYETF5,0.10,0.45)
ADF5,ADFF5,AGE27F,FSHOT5, SYETF5 = GROUP( 30.0,AGE22F,MSHOOT,0.10,0.30)
              AGED = INTGRL( 20.0, ADVNCE*(AGE27M + AGE27F)*          ...
                        AMAX1 ( 0.0,1.0 - .5 * MORTC )                ...
                        -FCNSW(NSHOOT,0., AGED*AMIN1(1.,0.2*MORTC),AGED ))
        CALVES = CF + CM
        YRLING = YF + YM
          BULLS = ADM1 + ADM2 + ADM3 + ADM4 + ADM5
          COWS= ADF1 + ADF2 + ADF3 + ADF4 + ADF5
         TOTFF = ADFF1 +ADFF2 + ADFF3 + ADFF4 + ADFF5
         TOTFM = ADFM1 + ADFM2 + ADFM3 + ADFM4 + ADFM5
*       RELATIVE CONDITION, FERTILITY AND MORTALITY FACTOR OF THE HERD
*       DEPEND UPON THE CARRYING CAPACITY OF THE GRASSLAND AND THE LOAD
PROCEDURE      CAP = HERBGE( MONTH )
        IF ( MONTH .NE. 1 ) TO TO 100
            W1 = W2
            W2 = LIMIT( 0.8, 1.2, GAUSS(12,1.0, 0.1))
            DW = ( W2 - W1 ) / 12.0
        100 YRVAR = W1 + DW * FLOAT( MONTH )
            CYCLE = NLFGEN( SEASON, FLOAT( MONTH ) )
FUNCTION   SEASON= ( 1.0 , 0.6 )                  , ...
                   ( 3.0 , 1.0 )                  , ...
                   ( 6.0 , 1.4 )                  , ...
                   ( 9.0 , 1.0 )                  , ...
                   (12.0 , 0.6 )
              CAP= YRVAR * CYCLE * AVCAP

    ENDPRO
          LOAD = 0.24*CF + 0.4*CM + 0.6* (YRLING +COWS +AGED) + BULLS
          COND = NLFGEN( CONDTB, AMIN1(C1.0, CAP/LOAD ) )
  FUNCTION CONDTB= ( 0.0 , 0.0 )  ,...
                   ( 0.3 , 0.5 )  ,...
                   ( 0.6 , 0.8 )  ,...
                   ( 1.0 , 1.0 )
          FERTC  = NLFGEN( FERTBL, COND )
  FUNCTION FERTBL = ( 0.0 , 0.0 )  ,...
                    ( 0.8 , 0.1 )  ,...
                    ( 1.0 , 1.0 )
          MORTC  = NLFGEN( MORTCT, COND )
  FUNCTION MORTCT= ( 0.0 , 20.0 )   ,...
                   ( 0.3 , 15.0 )   ,...
                   ( 0.6 , 2.5 )    ,...
                   ( 1.0 , 1.3 )

  FINISH    LOAD  = 180.0
  METHOD    RECT
  TIMER     DELT  = 1.0     , FINTM = 3200.0 ,  PRDEL = 12.0 ,...
            OUTDEL  = 12.0
  PETPLT  BULLS(0.0,.CENSUS,LOAD,COND) ,COWS(0.0,.CENSUS,LOAD,COND)
  PRINT   CALVES, YRLING, COWS,BULLS,AGED,AGE7M,AGE12M,AGE17M
  LABEL   NATURAL LIMITATION OF HERD -- NO SHOOTING POLICY
  PARAMETER AVCAP=7500.0 ,  LEVEL = 100000.0

  END
  PARAMETER    LEVEL = 10000.0
  RESET  LABEL
  LABEL  CONTROLLED SHOOTING OF HERD
  END
  STOP
```

Fig. 8 continued

inserted to keep the herd numbers under control without adversely affecting the health or age and sex composition of the herd. The time interval for computation is the month. Births occur in April, a census is taken in July (mating season), and hunting takes place in October of each year.

The resulting model has been made somewhat more complex than the previous examples in order to illustrate some additional features of the simulation program, yet it is still a relatively simple model. It could be easily made more sophisticated by including other factors of the environment and the buffalo's life processes in the program.

In the model, animals of each sex are divided into calves, yearlings, and five 5-year age groups, plus a single "AGED" group containing both males and females 27+ years old (Fig. 8). The number of animals in each group (TOTAL) is subdivided into the five yearly age classes N1, N2, N3, N4, N5. The AMAX1 function in the first operational statement of the MACRO is used as a convenient way to sum the five yearly age classes while insuring that TOTAL never becomes negative from numerical "round off". OLD is the number of animals in the year class N5 and which will be advanced to the next 5-year age group (or AGED) at their next birthday (APRIL). The total number of fertile animals (TFERTL) in the 5-year age group is the product of the fertility factors due to age and condition and the TOTAL.

A PROCEDURE is inserted into the MACRO to calculate the number of animals harvested in the 5-year age group (SHOT), the number yet to be harvested from other groups (SHTYET), and the fraction of the group harvested (FRSHT). The AMIN1 function limits the number harvested to not more than 80% of the total. The PROCEDURE also calculates the remainder to be harvested from the rest of the adult age groups (SHTYET) and the fraction of this age group harvested (FRSHT). The PROCEDURE is sorted as a unit, but its contents remain unsorted to permit the power of FORTRAN logic statements and other FORTRAN routines to be exploited.

The rest of the MACRO calculates monthly values for each yearly age class N1, N2, N3, N4, and N5 by integrating from an initial number of animals (10) over the number advanced from the previous yearly age class (F1 * NIN) minus the number advanced to the next yearly age class (F2 * N1); F1 and F2 are calculated from ADVNCE which takes the value 1.0 on the birth month, April, and 0.0 for all other months and from MORTA1, the number of animals dying as a function of age and range condition factors (MORTA and MORTC) and of the fraction of the age group harvested (FRSHT). Note that the MORTA and FERTA factors are inputs for which actual values are substituted when the MACRO is called for below, whereas MORTC

and FERTC are factors obtained from structure statements near the end of the program in function generator blocks dependent on condition of the animals (COND).

The initial values of two parameters, the RATIO of fertile bulls to cows and the relative range condition in the first January (W2), are established in the INITIAL segment of the program.

The DYNAMIC segment comes next with the monthly time interval (MONTH) defined numerically from 1.0 to 12.0 and reset at 1.0 at the beginning of each yearly cycle by use of the AMOD function. This is followed by three PROCEDURES labeled APRIL, JULY, and OCTOBR and they relate to the activity identified with each of these months, calving, census taking, and harvesting, respectively. In each a FORTRAN "IF" statement is used to activate calculations done for that month.

The PROCEDURE called APRIL calculates the number of animals born, NBORN, from the product of the total fertile females, TOTFF, and the minimum of 1.0 and (8.0 * RATIO) of fertile bulls to fertile cows (calculated in the JULY PROCEDURE). The proportion of female calves is 0.47 and of male calves is 0.53.

The PROCEDURE called JULY makes the annual herd count ACTUAL = CALVES + YRLING + BULLS + COWS + AGED. Since real censuses involve errors of estimation the CENSUS is weighted by a random number generator with a normal distribution GAUSS (71, 1.0, 0.05) which uses any odd number as a starting point (71 in this example), a mean of 1.0, and a chosen standard deviation of 0.05. Estimated load ESTLD on the range resource is also weighted by a random number generator. The LOAD of the herd on the forage resources, from which EDTLD is derived, is calculated below from weightings based on the feed required for the various classes of buffalo with bulls weighted as 1.0.

The PROCEDURE called OCTOBR determines the number of animals to be harvested based on the CENSUS and on the desired minimum size of herd LEVEL, an input parameter that can be adjusted to implement various harvesting policies.

Next, the numbers of male calves CM, female calves CF, yearling males YM, and yearling females YF are calculated using the integration function INTGRL. Initial numbers are specified, and then the procedure for summing subsequent time intervals is presented as functions of the number entering the age class being considered and the number leaving it due to mortality or due to advancing to the next age class during the birth month (when ADVNCE = 1.0 in APRIL).

Following this are 10 statements each calling on the MACRO defined initially (GROUP). Outputs and inputs are respecified for each use of

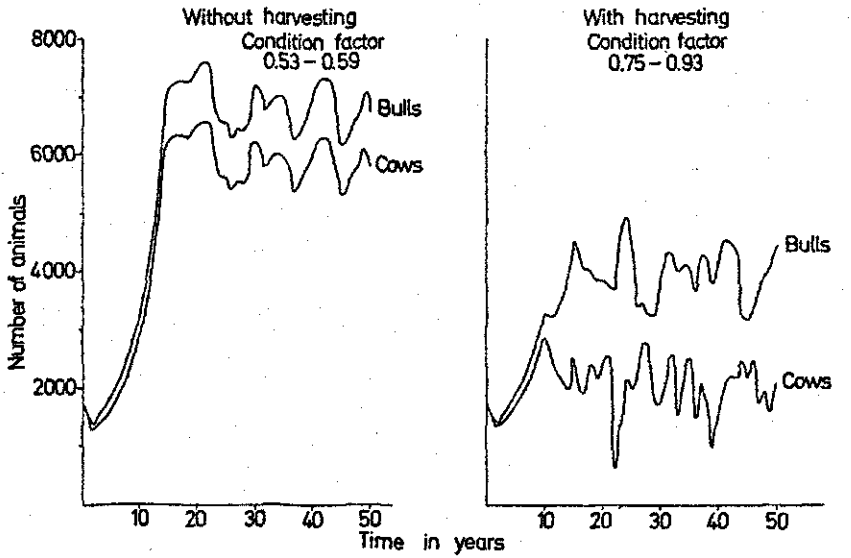

Fig. 9. Plot of herd size without and with harvesting

the MACRO, and the inputs that are calculated of necessity before using the MACRO are YM, YF, and NSHOOT. Appropriate ordering of the FORTRAN program is attained automatically by the sorting algorithm. Inability of the CSMP program to sort because of an algebraic loop will result in a diagnostic message which indicates the failure of the sorting algorithm and lists the variables in the loop. However, such is not the case here.

The last PROCEDURE computes the grazing capacity CAP of the range resource as a function of a generator of normally distributed random numbers (GAUSS) and a seasonal CYCLE based on a table called SEASON from which values are obtained by quadratic interpolation using the arbitrary function generator NLFGEN. Average capacity AVCAP is a parameter input and should be chosen in conjunction with appropriate values of LEVEL.

The statements from FINISH to STOP constitute the terminal segment of the program and control the operation and the output of the program. For the first run the parameter LEVEL is set so high (100,000 animals) that no animals are harvested, allowing "natural causes" to limit herd population dynamics. The print-plot shows both the numbers of bulls and cows falling slightly the first two years and then rising to a plateau at 17 years. After that, fluctuations occur in response to range condition changes. In the second run a harvesting

policy is implemented by reducing the parameter LEVEL to 10,000 animals. This results in lower average numbers of bulls and cows and much greater variability from year to year (Fig. 9).

Results of this simulation illustrate how the herd size and composition vary in a natural setting with weather and range condition as relevant variables and under various harvesting strategies. Methods of this type should help ecologists to understand natural phenomena and wildlife managers to select the best strategy to achieve game management goals.

It is readily seen that many complex life processes can be simulated which would be difficult or impossible using conventional statistical methods. Increasing the complexity of the model to make it more realistic still is readily achieved with S/360 CSMP. Consideration of the interrelationship of all animals can be handled in a similar manner as the limpet-seaweed or barley-oats example. An automated search algorithm could be inserted in the TERMINAL segment to optimize a stated objective in finite values.

The flexibility and adaptability of simulation languages, which these examples illustrate, make them powerful tools that can be easily adapted to ecological problems.

References

Baeumer, K., Wit, C. T. de: Competitive interference of plant species in monoculture and mixed stands. Neth. J. Agr. Sci. 16, 103—122 (1968).

Bartlett, M. S.: Stochastic population models in ecology and epidemiology. New York: Wiley 1960.

Brennan, R. D.: Continuous system modeling programs, state-of-the-art and prospectus for development. In: Simulation programming languages. Amsterdam: North Holland 1968.

Franks, R. G. E.: Mathematical modeling in chemical engineering. New York: Wiley 1967.

Fuller, W. A.: The biology and management of the bison of Wood Buffalo National Park. Can. Dept. Northern Affairs National Resources, Wildlife Management Bull. Ser. 1, No 16 (1962).

Garfinkel, D.: A simulation study of the effect on simple ecological systems of making rate of increase of population density-dependent. Simulation 8, 111—122 (1967).

— Sack, R.: Digital computer simulation of an ecological system, based on a modified mass action law. Ecology 45, 502—507 (1964).

IBM S/360 Continuous system modeling program (360 A-CX-16X), Program Reference Manual, Form No H20-0367-2 (1968).

Lotka, A. J.: Elements of mathematical biology. New York: Dover 1925, reprinted 1956.

McHugh, T. C.: Social behavior of the american buffalo (Bison bison bison). Zoologica 43, 1—40 (1958).

Patten, B. C.: System ecology: a course sequence in mathematical ecology. BioScience 16, 593—598 (1966).

Paulik, G. J., Greenough, J. W., Jr.: Management analysis for a salmon resource system. In: K. E. F. Watt, Systems analysis in ecology. New York: Academic Press 1966.

Roe, F. G.: The North American Buffalo. Toronto, Canada: University of Toronto Press 1951.

Slobodkin, L. B.: Growth and regulation of animal populations. New York: Holt, Rinehart and Winston, 1961.

Volterra, V.: Variations and fluctuations of the number of individuals in animal species living together (1928). Reprinted in R. N. Chapman, Animal ecology. New York: McGraw-Hill 1931.

Watt, K. E. F.: System analysis in ecology. New York: Academic Press 1966.
— Ecology and resource management. New York: Mc Graw Hill 1968.

Wit, C. T. de, Brouwer, R.: Über ein dynamisches Modell des vegetativen Wachstums von Pflanzenbeständen. Angew. Bot. 42, 1—12 (1968).

Yates, F. E., Brennan, R. D., Urquhart, J., Dallman, M. F., Li, C. C., Halpern, W.: A continuous system model of adrenocortical function. In: M. D. Mesarovic, ed., Systems theory and biology. New York: Springer 1968.

W. A. Wililams
Agronomy and Range Science
University of California
Davis, Ca. 95616