Technical Report No. 64

PWNEE: A Grassland Ecosystem Model

L. J. Bledsoe, R. C. Francis,
G. L. Swartzman, and J. D. Gustafson
Colorado State University

GRASSLAND BIOME
U. S. International Biological Program
January 1971

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

ABSTRACT

The primary objective of Grassland Biome Project modelling efforts this year (1970) has been to produce a mechanistic total system model of a grassland ecosystem. The result has been a model which is mechanistic to the extent that, wherever possible, the mathematical formulations are analogous, at some level of resolution, to the functional mechanisms operating within the system.

The model is primarily designed to describe the shortgrass prairie ecosystem of the Pawnee National Grassland. It is designed as a highly modular system for two reasons:

i. So that individual processes or mechanisms may be changed as information becomes increasingly available, and

ii. So that the model can be used in situations having greater (Pawnee Site) or lesser (Comprehensive Sites) detail in data and information.

The current version of the model is in a first-pass condition, and has not been subjected to extensive scientific debugging (i.e., the mechanisms have not been closely reexamined by biologists).

The model is structured in the following way:

i. It is a time-dependent biomass model. No spatial aspects are taken into consideration at present.

ii. The primary equations to be solved make up a series of first-order differential equations. Thus, the equations for the principal system variables express the rate of change of biomass with respect to time.

iii. The total model is made up of trophic level submodels. Within each trophic level various functional relations describe the processes.

A set of 40 first-order differential equations has been developed to describe the abiotic, producer, consumer, and decomposer components of the ecosystem. The abiotic section involves driving forces of solar energy, air temperature, wind speed and precipitation, and driven variables of micro-climatic temperature, soil temperature, and soil moisture. The producer variables consist of biomass density of aboveground live biomass for four plant functional groups, plant standing dead, plant litter, and plant live roots. The consumer biomass is compartmentalized as animal live material, animal dead material, and animal fecal material. The animal live biomass is further subdivided into five functional groups (wild primary consumers-mammal, domestic primary consumers-mammal, secondary consumers-mammal, birds, and insects). The decomposer compartments are mediated by microbial functional groups whose activity is in turn controlled by their biotic and abiotic environment.

ACKNOWLEDGEMENTS

The authors wish to express their appreciation to all of the scientific, administrative, and clerical personnel and graduate research assistants of the Grassland Biome study for help and encouragement provided in the development of the PWNEE model as well as the manuscript for this report. A research project of this type must be accomplished by a team effort and the authorship of specific reports is frequently a poor guide to the source of ideas and distribution of effort. The team effort for this project included, but was not limited to the following persons: Norm French, Larry Harris, George Innis, Don Jameson, Leonard Paur, Freeman M. Smith, and George M. Van Dyne.

1. INTRODUCTION

The purpose of this report is to provide the basic mathematical documentation for a Grassland Ecosystem Model developed by the modelling group of the Grassland Biome Study of the US IBP. It's primary purpose is to communicate with persons capable of reading and understanding conventional mathematical notations. A second purpose is to communicate the model mechanisms to the general biological scientist who may not be conversant in mathematical notation. A third purpose is to indicate the directions which further research on high-resolution models of whole ecosystems will take in the Grassland Biome Study.

The purpose of the model, entitled PWNEE, MOD 1, is to mathematically represent a shortgrass prairie ecosystem, specifically at the Pawnee Site where an intensive research effort is being undertaken. In its attempt to cover all trophic levels of the ecosystem, the model involves a great deal of systems complexity. On the other hand, the mechanisms for individual processes are made as simple as possible so as to make the initial system organization job not overly complex. The primary purpose of this version of PWNEE is to stimulate discussion and feedback on all phases of this modelling approach--the biological mechanisms, the mathematical notation, the computer implementation, and this report--as a method of communicating modelling ideas.

The mechanisms in the current version of the model have not been thoroughly examined by experimental scientists; some of the mechanisms were derived from the literature or by consultation with biologists. A great many mechanisms fall into the category of "intuitive guesses" where the intuition is that of the authors of this report who are not biological scientists. The model was implemented on the computer in a highly modular fashion, i.e., most of the processes or mechanisms were implemented as separate FORTRAN functions or subroutines. This was done to facilitate changes in future versions of the model. This also allows for a set of different mechanisms to be substituted for a particular process of the system. In this way, the same general structure may be used with high resolution mechanisms to compare with Pawnee Site data and lower resolution mechanisms to apply to Comprehensive Site data. Thus, the purpose of this first version of a complete ecosystem model is to provide a skeletal structure within which modifications may be made to eventually provide a realistic description of grassland ecosystem functioning.

In the light of the above stated purpose of this model, explanations of mechanisms contained in this report will frequently be without reference to the open literature. Wherever such a reference is appropriate, it will be included. Generally, the rationale which goes into the construction of the mechanisms of the model is sufficiently simple that it will be transparent upon study of the mechanism explanation. When faced with a plethora of possible alternative causes for a known phenomenon, the selection of a simple mechanism involving only one or a few of these causes is arbitrary. Justification of the mechanism selected over other mechanisms is of little use in this case, since it is only one of a set of equally acceptable hypotheses.

When the reader encounters mechanisms in the model which are presented without explicit written justification or literature citation, he may assume the above situation to be the case. Some of the mechanisms have been previously reported in Bledsoe and Jameson (1969). These will be indicated.

A particular problem in the writing of this report has been the diversity of the audience to which it is directed. As stated in the beginning of this section, we are primarily concerned with a documentation of mathematics. We are also concerned with a description for the non-mathematically inclined reader and a documentation of the rationale which went into the construction of the model. Later reports concerning other versions of this model could have a different purpose since the objective of higher realism will demand more detailed justification of mechanisms. As a result of our varied objectives in this report, some mathematically inclined readers may be frustrated by the detail which goes into the description of mathematics which, to them, is trivial, or they may be confused by some conventions of biological terminology. On the other hand, the biologically inclined readers may be confused by the use of nomenclature, units, and equation conventions which are common among mathematically oriented scientists. Clearly, the detailed explanation of conventions common to various scientific disciplines is beyond the scope of this report. We can only ask our readers to bear with us and, perhaps, to read selectively, skipping over sections which are either confusing or seem trivial. We have revised this manuscript repeatedly at the suggestions of proofreaders from divergent backgrounds.

The next section contains a discussion of the organization of the model and of the nomenclatural and notational conventions used. Following is a presentation of the algebraic form of the mechanisms in the driving functions, abiotic, producer, consumer, and decomposer sections of PWNEE, and a brief discussion of possible future areas of modifications for each section.

Appendix material includes: (A) a summary of algebraic and differential equations in the model; (B) a glossary of symbols used for variables and parameters in the model, including a cross reference between algebraic and computer notation; (C) a description of the computer implementation of the model including a listing of the FORTRAN code; (D) a description of input and output formats for the computer program together with a listing of parameter values used in (E), a final appendix section with a set of graphs containing results of typical computer simulation of the system.

2. DESCRIPTION OF THE MODEL

2.1 General Description

PWNEE is a system of 40 principal variables, 33 of which are represented by first order ordinary differential equations, generally nonlinear. The remaining seven are represented by algebraic equations. The equations are designed to describe the general functioning of a grassland ecosystem; the variables are somewhat arbitrarily categorized into 4 groups: abiotic, producer, consumer, and decomposer. Fig. 2.1 shows in a diagrammatic way the approximate chain of causal relations among these four groups and subdivisions within the groups. However, the reader should not attempt to draw conclusions about the operation of the model solely by reference to the diagram. The model is designed to be deterministic; however, this distinction is somewhat trivial since the parameters could be drawn randomly from emperically determined distribution functions in a series of system simulations to provide a measure of the stochastic nature of the system variables. The time resolution of the model is such that the derivatives of the principal variables will have, in some cases, significant changes over a 3 to 5 minute period. At the other end of the scale, the model should evenutally depict ecosystem changes over a longer time, e.g., several years. Currently, simulations are being made only on an intra-year basis.

2.2 Notation and Nomenclature

In order to enhance communication about the ecosystem model, it is necessary to adopt a set of conventions concerning the names for the different variables which occur in the mathematical system description. Every discipline

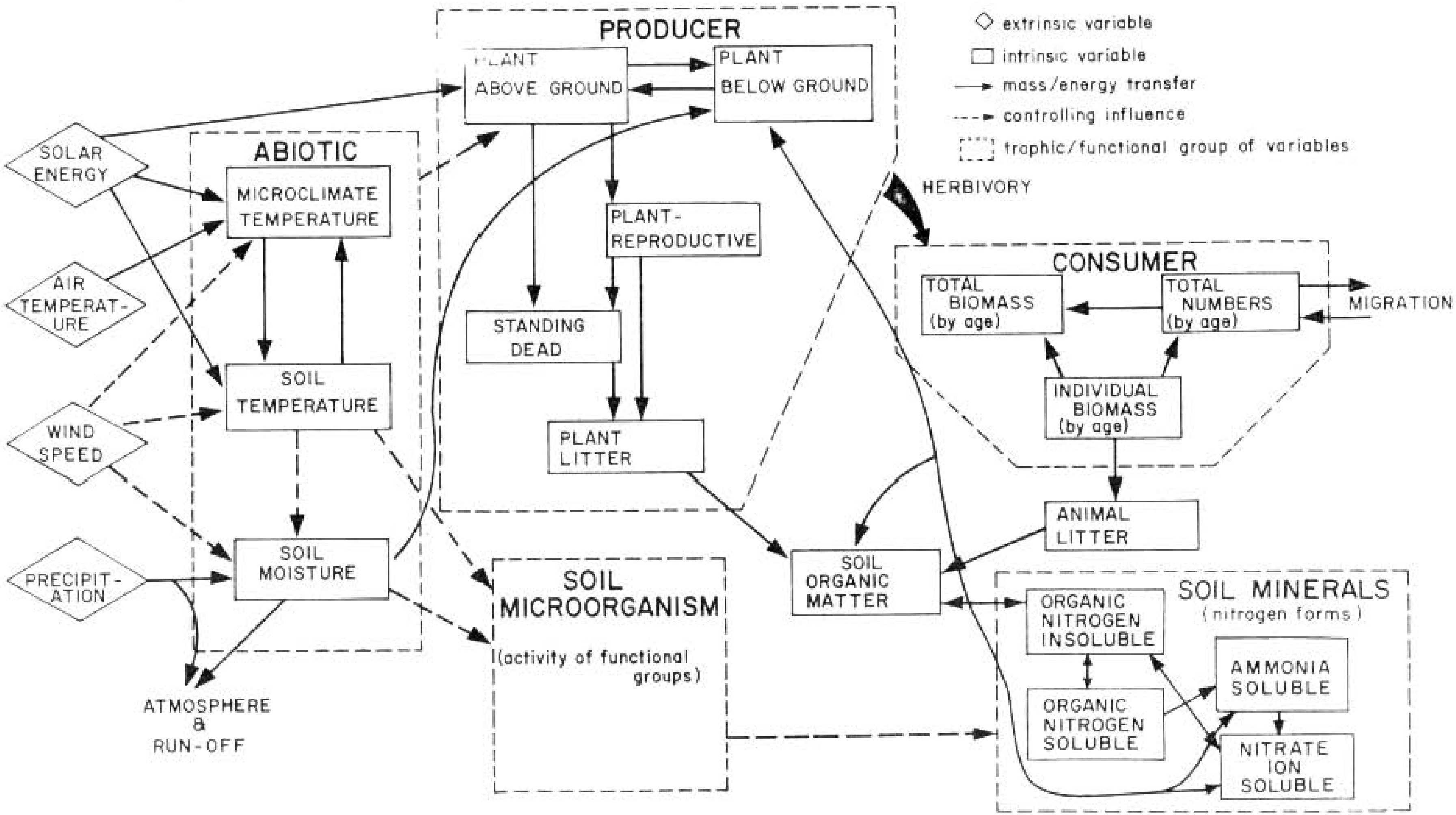


Fig. 2.1. Generalized diagram of the grassland ecosystem showing biomass, energy, and nutrient flows as represented by equations in this text. Species enumeration and process names have been omitted for clarity.

which utilizes mathematical models has, to some degree, adopted their own conventions. However, there is little agreement across disciplines and, even within disciplines the conventions are rarely well defined. This subsection describes the nomenclature which has been decided upon to be used in this report and tentatively for use in future Grassland Biome modelling reports. This nomenclature has evolved after a great deal of thought and discussion. The biological problems of the model are of greater importance than the nomenclature problems and we will allocate our time accordingly. The important aspect of the nomenclature is that a system be adopted and adhered to. Thus, we expect our nomenclature to be modified as our experience with the system increases, but we shall not be overly concerned with minor inadequacies.

Analogous to the nomenclature problem is the problem of selecting an algebraic notation for written representation of the mathematical equations which comprise PWNEE. Again, various disciplines have adopted their own conventions and we have abstracted some of these for our own use. The general criteria for selection of a notational system were: 1) that the notation must clearly discriminate among the types of variables described by the nomenclatural system; 2) that it must do as much as possible to enhance the understanding of the mathematical systems by the reader who is not familiar with the model; and 3) that it must be mathematically convenient to work with. The decision as to what distinctions are to be made among the variables was made primarily upon the basis of biological criteria rather than the type of mathematics associated with the variables. Thus, we do not distinguish between whether the variables are defined by differential equations as opposed to algebraic equations but rather between the variables which are more or less important to the biological description of the ecosystem.

There are six general types of variables utilized in the model, defined as follows:

Driving variables. A driving variable is one which is necessary for the elaboration of other variables of the system, but with whose mechanistic description we are not concerned. For example, it is necessary to know the precipitation rate in order to drive the soil moisture differential equation. However, we are not concerned with the mechanisms responsible for the occurrence of precipitation. We are satisfied with a non-mechanistic description of precipitation such as would be given by a tabulation of actual values or a stochastic rainfall generator.

The algebraic notation for driving variables is a small letter "z" followed by a single subscript. See Table 2.1 for examples.

Principal system variables. The principal system variables (PSV's) are the primary, dependent variables of the model. These are variables such as aboveground live plant biomass or densities of small mammals which comprise the principal output of the model, i.e., a graph of the PSV's as a function of time can be thought of as the solution to the set of equations which comprise the model. The purpose of the computer program which implements the model is to produce such a set of graphs. The algebraic notation for a PSV consists of one or two English letters, possibly subscripted, the first of which is a capital and second of which is a small letter. See Table 2.1 for examples.

Table 2.1. Examples of variable notation and nomenclature used in this report.

Type of Variable	Example--name of variable	Example--symbol for variable
Driving variable	Air temperature	z_2
	Precipitation rate	z_3
Principal system variable (PSV)	Aboveground biomass density of plant group i	Al_i
	Biomass density of consumer	C_i
Intermediate system variable (ISV)	Translocation rate above- to belowground, plant group i	bt_i
	Ingestion rate per unit body weight, ith consumer group	fc_i
Independent variable (time)	Time from Jan. 1, seconds	t
	Time from Jan. 1, hours, modulo 24	thm
Dummy Function argument	--	x
		x_1
		x_2
Parameters	Threshold of positive photosynthesis response to temperature, plant group i (first parameter used in formulation of Tc)	pTc_{i1}
	Upper threshold for density of consumed material (j) at which food consumption will be equal to $pfc1_i$ (second parameter used in formulation of the ISV fc_i)	pfc_{i2j}

Intermediate system variables. Intermediate system variables (ISV's) are the values of mathematical functions which are necessary in the calculation of the PSV's. The distinction between PSV's and ISV's is based upon the following biological (as opposed to mathematical) criterion: a PSV is decided upon a priori as a desired output variable of the model; an ISV arises a posteriori in the process of writing equations to relate the PSV's. As shown in the example in Table 2.1, ISV's are frequently rates, i.e., they usually give the change per unit time or flow of a biomass or energy density variable. Thus, they are frequently in the same units as the derivative of a principle system variable. The differential equation for the principal system variables frequently consists of a sum of terms, each term containing an ISV, a PSV, and a parameter (explained below). For example, see the first term in equation 5.1 of Appendix A.

The algebraic notation utilized for ISV's consists of two small letters, sometimes with a subscript. Table 2.1 contains some examples. The first of these two letters may not be a p, t, z, or x as these are reserved for other types of variables in the model as explained above and below.

Independent variables. Since this model does not have explicit spatial variability, the only independent variable is time. It is frequently convenient to have time available in different systems of units for the formulation of equations in different parts of the model. For example, in formulating a mechanism which fluctuates diurnally, such as windspeed, the time of day in hours might be useful; for a seasonal fluctuation, the time of year in months might be useful. We have utilized separate symbols to indicate time in the various units which might be utilized for the independent variable. These are summarized in Table 2.2 together with the symbolic notation of the computer implementation described in Appendix C.

Dummy function arguments. In the simulation of an ecosystem model, it is frequently necessary to utilize mathematical functions purely for their mathematical, as opposed to biological, properties. For example, the usual trigonometric functions such as sine and cosine have been used. In addition, there are less common functions which are defined in the text of the model documentations. For example, we have found it necessary to utilize a function which asymptotes to zero as its' argument decreases and to one as its' argument increases, with a smooth transition in between (at(x_1, x_2, x_3), see equation 7.18). We have called these functions intermediate system variables. When they are defined, it is necessary to have variables to use for their arguments. Conventional mathematical usage labels these arguments "dummy variables," and we will utilize a small letter "x," sometimes subscripted, as shown in Table 2.1, for a symbol. For an example of the use of a dummy variable see equation 7.18.

Parameters. In any mathematical model there are variables whose derivative, with respect to all other variables of the system, is zero. These are frequently termed coefficients, constants, or parameters. For various reasons, some of them arbitrary, we have decided to uniformly call such variables "parameters." The model uses a great number of parameters and we have adopted a notational system to make it somewhat easier to relate the parameters to the ISV's and PSV's with which they are associated. The parameters always begin with a small letter p and the next one or two letters correspond

Table 2.2. Symbolic notation for time in different units.

Algebraic Symbol	Description	Program Symbol
t	Time, seconds	T
td	Time, days	TD
tdm	Time, days, modulo 365	TDM
th	Time, hours	TH
thm	Time, hours, modulo 24	THM
tw	Time, weeks	TW
twm	Time, weeks, modulo 52	TWM
tm	Time, months (Julian)	TM
tmm	Time, months, modulo 12	TMM
ty	Time, years	TY

to the PSV or ISV which the parameter is used to formulate. In the event that a parameter is used in formulation of more than one PSV or ISV, this selection is arbitrary. If the system variable referred to has subscripts, they will appear as the first subscript in the parameter. Additional subscripts may appear as well. These additional subscripts may be used simply to enumerate the various parameters involved in formulation of the ISV or PSV. They might also refer to other enumerated variables of the model. The example in Table 2.1 should help to clarify the usage. The parameter pTc_{i1} is the first parameter used in the formulation of the principal system variable Tc. The subscript i indicates that it refers to the *ith* plant group. The parameter pfc_{i2j} is the second parameter used to formulate the intermediate system variable fc_i (food consumption rate of *ith* consumer); the subscript j refers to the *jth* group being consumed.

A uniform system of physical units facilitates the construction of an ecosystem model by avoiding the use of parameters whose sole purpose is to convert other variables to the proper system of units. We have decided to utilize centimeters for length, grams for weight, and seconds for time for nearly all of the variables of the model. When exceptions occur, they will be noted in the text. Appendix B contains a glossary of all of the system variables, together with their units, their algebraic notation, and the corresponding notation utilized in the computer implementation. The notation used in the computer implementation differs substantially from the algebraic notation, because the two have evolved at different rates during model development. Table 2.3 contains a summary of the numbers of different

Table 2.3. Number of variables of different types used in various sections of PWNEE, MOD 1.

Type of Variable	Section					
	Driving Variables	Abiotic	Producer	Consumer	Decomposer	Total
Driving	4	-	-	-	-	4
PSV	-	3	7	17	13	40
ISV	1	0	51	249	21	321
Parameters	5	2	116	217	59	340

types of variables used in the various sections of PWNEE. This should give the reader some idea of the size of the PWNEE model.

2.3 Parameter Values

The text will frequently show numerical values inserted in equations where a symbol for a parameter would seem more logical. This is a reflection of the fact that at the time of preparation of this manuscript, the model had not been "cleaned up" thoroughly and contained a number of relatively minor inconsistencies. The numerical values were not replaced with symbols, primarily because the computer implementation contained the same values rather than symbols for storage locations where a value could be input from a parameter card (see Appendix D). Though insertion of a symbol for a number in the text of this report is a minor task, modification of the computer program is more involved. These and other modifications are being made continuously, both to the model and its computer implementation, and will be reflected in the documentation for PWNEE, MOD 2.

In order to experiment with the computer implementations of a model, it is necessary to have values for all model parameters. Ideally, the parameter values should be chosen to make the intermediate system variables conform to empirically determined results. In point of fact, the empirical results are not available for most of the ISV's, and it is necessary to use a combination of several heuristic techniques to obtain initial estimates of parameter values. One such technique is to assume a certain percentage growth rate under optimal or normal conditions for principle system variables and adjust the parameters to provide that rate. For example, one might assume that, under optimal conditions of moisture, temperature, sunlight, and nutrients, the warm season grasses will have a net production of 5% per day. This would dictate the value for the parameter pAl_1 of $.05(\text{day})^{-1}$ or $.058 \times 10^{-5}$ $(\text{seconds})^{-1}$. Parameter values for the translocation function might be calculated by making crude assumptions concerning relative above- and belowground growth rates. The justification for the assumptions are not based upon any literature or experimental values, but rather by the intuition and experience of the biological scientist or the modeller. The numerical estimates *per se* are made by a modeller (who may or may not be also a biologist), however his information comes through consultation with biological scientists. Frequently, the consultation may involve examination of data concerning similar species. As stated earlier, at the stage of model building represented by this report, the fact that the assumptions made are very crude does not affect the overall objectives of the model. Appendix D contains a set of numerical values for the parameters utilized to produce the output in Appendix E. The same parameters were utilized to produce the graphs of intermediate system variables which are found in later sections of the text. Time does not permit giving a detailed explanation of the rationale for the determination of each parameter; as model development proceeds these parameter values change so rapidly that the rationale is of little use in any event. (The fact that the parameter values are so rapidly modified during model development is one reason why they are not called constants.)

2.4 Updating and Validating the Model

Since a purpose of this model is to stimulate critisism to improve future versions, it is appropriate to comment on the manner in which PWNEE might be updated. Basically, there are four types of modifications which can be made. These correspond to the steps used in producing the model originally. In order of complexity of the modification, most complex first, these areas are as follows:

(i) addition and deletion of PSV's,

(ii) addition and deletion of terms in the equations for the PSV's or their rates of change,

(iii) changes in the formulation of functional curves which relate the ISV's to other system variables, and

(iv) changes in the values of model parameters.

A change in the value of a model parameter is a very easy modification; additions and deletions of PSV's affect the model much more drastically and are more difficult to implement and document. Modifications of type two and three change the basic mathematics of the model. It is here where we expect most activity in updating the model to take place. Criticisms of any type will be useful and interesting. Only those which fall under the category of specific changes of one of the above types will result in any immediate improvement in the model. Suggestions concerning a change in basic methodology have been initiated in other project phases. The differential equation approach is supported by years of experience in other scientific disciplines and a growing literature in ecology. A great deal of thought has gone into the selection of this approach. Alternate methods are being explored as separate projects, but we are committed to a major effort with the differential equation approach for high resolution ecosystem models.

In order to achieve a smooth and orderly transition from our initial modelling efforts into a reputable and realistic ecosystem model, it is necessary to make changes a small amount at a time. For example, biological scientists may feel that the entire range of principle system variables is fundamentally incorrect and must be changed. However, in order to avoid redoing the work which has gone into the current version of the model, we would suggest adding and deleting PSV's one at a time with constant reevaluation of model output. It is only in this way that future results can be built upon past effort and that we can profit by our mistakes as we proceed.

Validation of a whole ecosystem model is a topic beyond the scope of this report, however a few comments are in order. Ideally, validation involves comparison of all PSV's and ISV's of the model with experimental data. ISV's would be measured over the range of all environmental parameters of which they are a function (the "process studies" of the Grassland Biome), and PSV's would be monitored for a sufficient number of years to provide a wide cross section of environmental conditions. If the PSV's of the model, when graphed as a function of time, do not agree with the corresponding empirically determined values, the modeller must reformulate parts of the system and try again (assuming the system has been driven with the appropriate data). It is impractical of course to determine the complete range of all ISV's as a function of all variables and to measure all ecosystem variables corresponding to PSV's. The Grassland Biome study is endeavoring to do as much of this type of experimental measurement as possible. Fortunately, model validation is not an absolute thing but rather a continuum. It can be defined, in a practical sense, as any technique which helps the modeller to gain confidence in the efficacy of his model as a description of reality. As defined in Bledsoe and Jameson (1969), a model is an abstraction of reality. The level of the abstraction is determined by the resolution of the model. The concept of model resolution is discussed in Grassland Biome Technical Report No. 32 (Swartzman 1970). A great many of the principle system variables of the PWNEE model have been measured in the 1970 field season. Some of those which were not measured in 1970 will be measured in 1971 and succeeding years. Consider some examples of model validation in areas where other than complete data are available. If a total predicted standing crop of all plant functional groups in the model corresponds with the total standing crop as measured in the field, this would be an aid to determining the reality of the model mechanisms, even if comparison by species or functional groups could not be done, and thus, is a method of validation. Similarly, if project scientists feel that the rate of change of a principle system variable, such as animal density, has been completely formulated to a realistic degree, they may have a great deal of confidence in the prediction of animal numbers in spite of the fact that field technique might not be adequate to provide a check. Such a situation could result from laboratory studies or literature data providing appropriate ISV information. It is true that in some areas where it is very difficult to make field measurements of model PSV's, it may be desirable to reformulate the model and redefine the principle system variables to correspond to those field values which can be measured. However, this need not always be the only alternative as pointed out above.

An objective of research in the Grassland Biome is to provide the most complete and realistic ecosystem models at several levels of resolution which are possible with the current state of ecological knowledge and experimental techniques, subject to a reasonable monetary constraint. Exactly how complete and realistic these models will be remains to be seen. We believe that they will represent a considerable enhancement of the knowledge of the functioning of grassland ecosystems. This enhancement will probably result from the integration of existing knowledge rather than from the discovery of new fundamental truths in separate subdivisions of the ecosystem.

3. DRIVING VARIABLES

Driving variables are those ecosystem variables which are not produced via mechanistically formulated equations but are nevertheless needed as input to the rest of the model. For example, we are not interested in a mechanistic formulation to predict air temperature (at 2 m) but we do need to know what air temperature is at any time in order to predict, say, photosynthesis rate. The driving functions of the current PWNEE model are z_1--net shortwave radiation [cal/(cm^2·sec)], z_2--air temperature at 2 m (°C), z_3--precipitation rate (cm/sec), and z_4--wind speed at 2 m (cm/sec). Our current notational scheme utilizes a small letter "z" with a single subscript for a driving variable.

There are three basic methods of inputing driving variables to the computer implementation of the model:

i. an actual record of the variables as recorded on an experimental site for some time period. The program must provide an interpolation scheme to provide information between the recorded time points since any data entered into a digital computer must be discrete. This scheme can be a simple linear interpolation if time points are close, relative to the time resolution of the model (say, every five minutes), or it must be more complex if there is a lengthy spacing between time points (say, twice a day);

ii. a deterministically generated variable. This might be used if data were unavailable for some needed driving variable for purposes of checking out the model. For example, one might assume a simple sinusoidal variation in air temperature or solar radiation;

iii. a stochastically generated variable. Here one would employ a numerical scheme which would produce a set of pseudo-random numbers whose statistical characteristics match the characteristics of the variable as measured in the field.

The computer implementation of PWNEE, MOD 1, contains a mixture of methods (*i*) and (*ii*) to produce the needed driving variables.

PWNEE, MOD 1, utilizes the very simplest possible mechanisms to give continuous driving variables. At the time of construction of this version, the most readily available data consisted of records of daily precipitation and maximum and minimum temperatures. The wind speed and solar radiation functions contain only diurnal variability; seasonal changes are reflected only through the precipitation and air temperature variables. The reason is the priority placed on production of a whole system model as a target for discussion and framework for modification.

3.1 Net Shortwave Radiation

Method (*ii*) is utilized to give a figure for z_1 [cal/(cm^2·sec)], at any time. Solar radiation is assumed to be a truncated sinusoid with a peak value given by parameter pz_1 as shown by equation (3.1):

$$z_1 = \begin{cases} pz_1 \cdot \sin\left[\frac{2\pi}{24.}(thm - 6.)\right] & \text{if} \quad 6 \le thm \le 18 \\ 0 & \text{otherwise} \end{cases} \qquad (3.1)$$

Time of day in hours is given by thm.

3.2 Air Temperature

Method (*i*) is utilized here (z_2, °C) and a complex interpolation scheme has been employed since the recorded temperatures consist of daily maximum and minimum. Since the time of day at which the optimal temperatures occurred was

64

not recorded, the assumption is made that the maximum temperature always occurs at 2:00 PM (thm = 14.0) and the minimum at 5:00 AM (thm = 5.0). If the input daily values are regarded as parameters, $pz_{2,1,i}$ and $pz_{2,2,i}$, i = 1, . . ., 730 days, for maximum and minimum, respectively, for each day for two years, then the formulation is a cosine interpolation as follows:

$$z_2 = \begin{cases} (pz_{2,1,i-1} - pz_{2,2,i})\, cx[(thm + 10.) \cdot \frac{\pi}{15.}] + pz_{2,2,i} \\ \qquad \text{if} \quad 0 \le thm \le 5.0 \\ (pz_{2,1,i} - pz_{2,2,i})(1. - cx[(thm - 5.) \frac{\pi}{9}]) + pz_{2,2,i} \\ \qquad \text{if} \quad 5. < thm \le 14. \\ (pz_{2,1,i} - pz_{2,2,i+1})\, cx[(thm - 14.) \frac{\pi}{15.}] + pz_{2,2,i+1} \\ \qquad \text{if} \quad 14. < thm < 24. \end{cases} \tag{3.2}$$

where

$$cx(x_1) = \tfrac{1}{2}[1 + \cos(x_1)] \tag{3.3}$$

and i = td gives the time index.

3.3 Precipitation

Method (*i*) is utilized to give precipitation (z_3, cm/sec) rates for any time of day. The basic data record is the amount of rainfall which occurs in any one day without any information about when during the day it occurred, so some assumptions are required. We have assumed that the daily rainfall occurs between 3:00 AM and 8:00 AM (thm = 3.0 to thm = 8.0) during the months October through June. During the summer months, all precipitation occurs between 3:00 PM and 7:00 PM (thm = 15. to thm = 19.) to correspond with the afternoon thunderstorm typical of the Great Plains in late summer. The following algebraic equation summarizes this mechanism:

$$z_3 = \begin{cases} 1.39 \cdot 10^{-4}\ pz_{3,i} & \text{if} \quad (6. \leq \text{tmm} < 9.) \ \text{and} \ (15. \leq \text{thm} < 17.) \\ 5.56 \cdot 10^{-5}\ pz_{3,i} & \text{if} \quad (\text{tmm} < 6. \ \text{or} \ \text{tmm} \geq 9) \ \text{and} \ (3. \leq \text{thm} \leq 8.) \\ 0. & \text{otherwise} \end{cases} \qquad (3.4)$$

where i = td gives the time index and $pz_{3,i}$ is the amount of precipitation, in cm, occurring on the *ith* day. The numerical constants are for conversion of precipitation rate in cm/day to the appropriate time period, either two hours or five hours.

3.4 Wind Speed

Method (*ii*) is utilized here. We assume that the wind blows constantly at 5 mph during the hours 12:00 noon to 6:00 PM (thm = 12. to thm = 18.), at 0 mph between 9:00 PM and 9:00 AM (thm = 21. to thm = 9.), and changes linearly in the interim periods (thm = 9. - 12. and thm = 18. - 21.). The following equation summarizes this:

$$z_4 = \begin{cases} 447. & \text{if} \quad 12. \leq thm < 18. \\ (thm - 9.) \cdot 149. & \text{if} \quad 9. \leq thm < 12. \\ -(thm - 18.) \cdot 149. + 447. & \text{if} \quad 18. \leq thm < 21. \\ 0 & \text{if} \quad 21. \leq thm < 24. \end{cases} \tag{3.5}$$

3.5 Future Expansion

For air temperature and precipitation rate, near-continuous recording devices are operating on the Pawnee Site. A simple linear interpolation scheme will eventually provide a very realistic driving function for these two variables. Solar radiation and wind speed are also being continuously recorded for periods of several weeks at a time. Correlation of these data with periodic measurements of wind speed (e.g., total miles/day) and other meteorological parameters will allow generation of realistic values for z_1 and z_4. This would be a combination of methods (*i*) and (*ii*) or (*i*) and (*iii*).

For simulation of locations other than the Pawnee Site, a more realistic extension of the methods described herein might be used. These would be based on analysis of data from nearby stations of the U.S. Weather Bureau and utilization of existing functions in the meteorological and climatological literature. For example, the summer thunder storm intensity might be given by the Gamma distribution function employed by Amaracho and Broadstetter (1967). This would allow the storm intensity to vary within the storm period instead of assuming a square distribution as in MOD 1.

4. DRIVEN ABIOTIC VARIABLES

The abiotic section of PWNEE is relatively simple compared to other sections. This is because of the greater amount of knowledge already in existence concerning mathematical forms in this area. Rather than attempt to duplicate other research, we basically have sought to formulate mechanisms which will provide the rest of the system with needed variables. We expect that the hydrologists and agricultural engineers in the Grassland Biome study will be able to supply correct formulations in future modifications of PWNEE. The task of mathematical modelling is met with much less skepticism and much greater confidence by scientists of this area than in the biologically oriented areas. Thus, it is necessary to place a different emphasis on abiotic mechanism formulation in construction of a model designed to encourage mathematical statement of mechanisms.

The PSV's of this section are soil temperature, Ts, soil moisture, Ms, and soil surface temperature, Tc. There are no ISV's associated with the abiotic PSV's. Such obvious associated variables as soil moisture tension or nutrient status are included in later sections because of their use in formulating PSV's of those sections.

4.1 Principal System Variables

The soil surface temperature, Tc, is utilized synonymously with plant canopy temperature and is used in other sections of the model as the characteristic temperature for mediating aboveground plant physiological mechanisms. In a completely mechanistic approach, the soil would be broken down into thin horizontal layers with a PSV to describe the temperature of each layer. This would be done by developing heat budget equations for each layer to give rate of change of heat content as a function of other system variables (Whitman, 1969). For the current level of resolution of PWNEE we have regarded this as too complex and detailed an approach. We have adopted the following expedient in which Tc is formulated in an algebraic equation, in contrast to most PSV's of PWNEE.

Tc is regarded as being equal to air temperature, z_2, under conditions of high wind or zero solar radiation since radiative heating due to insolation can be removed by convection under these conditions. Otherwise, Tc can rise to as much as 10°C (an order of magnitude estimate) above z_2. The following equation summarizes this relation:

$$Tc = \begin{cases} z_2 + 10.\left(\frac{z_1}{pz_1}\right) & \text{if} \quad (z_4 \le 223.5) \text{ and } (z_1 > .0002) \\ z_2 & \text{otherwise} \end{cases} \tag{4.1}$$

Notice that the breakpoint for an increase in Tc over z_2 is a wind speed less than 5 mph (223.5 cm/sec).

Soil temperature, Ts, is calculated by a linear differential equation which causes a first order lag behind Tc. The time constant of the lag is pTs. We define Ts as an average for the top 20 cm of the soil.

$$\dot{T}s = pTs(Tc - Ts) \tag{4.2}$$

Fig. 4.1 gives a graph of z_2, Tc and Ts for the first few days of January as calculated in PWNEE. The temperature minima and maxima used in Appendix E were taken in 1967 near Akron, Colorado on an area of sandhills prairie (courtesy of USDA).

Soil moisture, Ms, is given in cm of water in the top 20 cm of the soil. It is characterized by a linear differential equation with precipitation rate as an input and a linear loss rate as the only driving mechanism. The rate constant for loss is pMs.

$$\dot{Ms} = z_3 - pMs \cdot Ms \tag{4.3}$$

The constant-coefficient linear loss term, $pMs \cdot Ms$, is used to account for all sources of moisture loss from the ground. With the precipitation driving functions used, the soil never becomes saturated so it was not necessary to put a saturation limiting mechanism into the model. Bledsoe and Jameson (1969) contains a description of such a mechanism, but it was not implemented for the reason given above. Fig. 4.2 gives a graph of Ms as a function of time for precipitation data taken concomitantly with the temperature data in Fig. 4.1.

The parameter values reported in Appendix D were all order of magnitude values with the exception of pMs. This parameter was chosen by trial and error to give soil moisture values to correspond with measurements made concomitantly with the precipitation values, $pz_{3,i}$, $i = 1, \ldots, 730$.

4.2 Future Expansion

PWNEE, MOD 2, will probably separate the soil surface and canopy temperatures into two PSV's with a heat budget formulation including net radiative heating and evaporative cooling for each. The information necessary to validate such mechanisms is currently being made (on the Pawnee Site) with automated digital recording devices. A separate modelling project on prediction of soil temperature as a function of depth is being carried out independently of Grassland Biome support by the Agricultural Engineering Department of the

University of Wyoming. Improvements in the soil temperature PSV will be based on this project. A soil moisture model is being developed by the personnel responsible for the microwatershed experimental project at the Pawnee Site (Department of Watershed Resources, Colorado State University). This model would consider such phenomena as runoff, evaporation, transpiration, surface roughness, slope, and aspect.

To give a better seasonal distinction in the model, PSV's for snow depth and hail might be added. The modifications mentioned above will probably require additional driving forces such as cloud cover and dew point depression. Modifications in other sections of the model may also require elaboration of the abiotic section to provide the necessary environmental data.

5. PRODUCER VARIABLES

5.1 Principal System Variables

There are seven producer principal system variables. The first four of these are the aboveground live biomass densities of four plant functional groups representing four of the major groups on the Pawnee Site. Al_1 denotes the biomass per square centimeter of warm season grasses, as typified by *Bouteloua gracilis*; Al_2 denotes the biomass of cool season grasses, as typified by *Agropyron smithii*; Al_3 represents forbs, as typified by *Sphaeralcia coccinea*; Al_4 represents *Opuntia polyacantha*, which makes up over 95% of the cactus species. Each of the PSV's, though typified by a single plant species, stands for the total biomass in each of the four functional plant groups. These four groups make up a large percentage of the plant biomass on the experimental pastures at the Pawnee Site. They also are the groups which are most important in consideration of grazing patterns for domestic herbivores on the site.

The rates of change of each of these four producer PSV's are represented by the differential equation given below.

$$\dot{Al}_i = pAl_i \cdot hp_i \cdot Al_i - (bt_i + da_i + ha_i) \text{ , } i = 1, \ldots, 4 \qquad (5.1)$$

where the processes of photosynthesis, translocation of photosynthate to belowground parts, death of aboveground parts, and harvest of aboveground parts are represented by hp_i, bt_i, da_i, and ha_i, respectively, for the *ith* plant group. All units are $g/(cm^2 \cdot sec)$. These ISV's vary with time depending on the states of various system variables. These relationships are discussed below. pAl_i is a conversion factor for CO_2 fixation rate to photosynthate accumulation rate for species i.

The last three producer system variables are the belowground (i.e., top 20 cm of soil) live plant biomass, Bl, the biomass of plant standing dead, Ad, and the biomass of plant litter, Lt. These last three variables are totals for all plant functional groups represented because species separation was not done (1970 field season) in the collection of field data on them. The rates of change for these variables are given in the model by the following three differential equations.

$$\dot{Bl} = \sum_{i=1}^{4} bt_i - (db + rb \cdot Bl + hb) \tag{5.2}$$

$$\dot{Ad} = \sum_{i=1}^{4} da_i - (sa \cdot Ad + hs) \tag{5.3}$$

$$\dot{Lt} = sa \cdot Ad - (hl + hm + lc \cdot Lt) \tag{5.4}$$

There are several intermediate system variables in these equations. The respiration rate of belowground plant parts is rb; hb, hs, and hl are the harvest rates by macroconsumers (as opposed to microbial harvest) of belowground, aboveground standing dead, and litter biomasses, respectively. The rate of shattering of plant standing dead material is sa, and lc is the rate of leaching of plant litter into the soil. The microbial harvest rate of plant litter is hm. All these rates are given in grams per square centimeter per second [gm/(cm$^2\cdot$sec)] or $(\text{sec})^{-1}$ if the ISV is multiplied by a biomass density.

The mechanisms used to represent each of these processes in the model will be discussed in the following sections.

5.2 Intermediate System Variables

The rate of net photosynthesis in this model is based on a graph of a generalized photosynthesis function given by Gates (1968). Photosynthesis (per unit leaf area) depends, in his work, on canopy temperature and intensity of photosynthetically active radiation. Qualitative curves are given by Gates for relative net photosynthesis as a function of these two abiotic variables. Equation 5.5 is an algebraic translation of the general shape of the curves as given by Gates. Fig. 5.1 is a graph of equation 5.5. The function gives net photosynthesis, and thus may be negative under environmental conditions in which respiration exceeds gross photosynthesis. A moisture stress and nutrient stress index, ms_i and ns_i, respectively, were added to include stresses specifically due to soil moisture and nutrient deficiencies. These ISV's are discussed below. Their functional range is from zero to one. Since Gate's curve is given for a unit leaf area, the leaf area index, la_i, is used to convert to a soil surface basis. The ISV, cr_i, fractional cover, takes into account the fact that the ground is not uniformly covered with vegetation of group i.

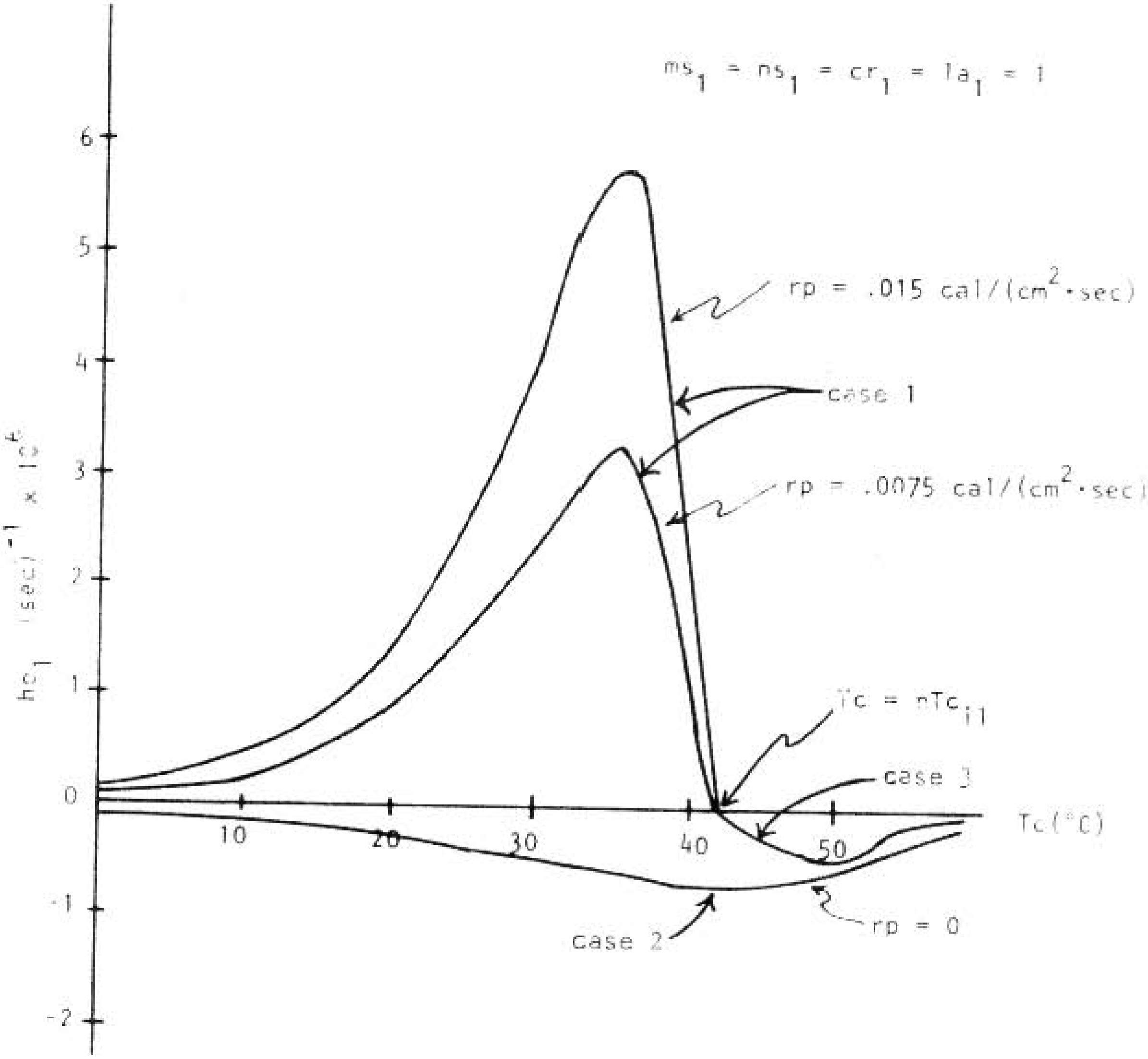


Fig. 5.1. Net photosynthesis (hp_1) vs. canopy temperature (Tc) for warm season grasses.

$$hp = \begin{cases} ms_i \cdot ns_i \cdot cr_i \cdot la_i \cdot php_{i1} \left[\dfrac{php_{i2} \cdot rp}{(1 + php_{i2} \cdot rp)} - \dfrac{php_{i2} \cdot prp_{2i}}{(1 + php_{i2} \cdot prp_{2i})} \right] \\ \quad [(pTc_{i1} - Tc)\exp(-php_{i3}[pTc_{i1} - Tc])] \\ \qquad \text{if } rp > prp_{2i} \text{ and } (0 < Tc < pTc_{i1}) \qquad [\text{case 1}] \\ -ms_i \cdot ns_i \cdot cr_i \cdot la_i \cdot php_{i4} \left[\dfrac{(prp_{2i} - rp)}{prp_{2i}} \right] \left(\dfrac{Tc}{pTc_{i2}} \right)^{php_{i5}} \\ \quad \exp \left\{ \left(\dfrac{php_{i5}}{php_{i6}} \right) \left[1 - \left(\dfrac{Tc}{pTc_{i2}} \right)^{php_{i6}} \right] \right\} \\ \qquad \text{if } rp < prp_{2i} \text{ and } Tc > 0 \qquad [\text{case 2}] \\ -ms_i \cdot ns_i \cdot cr_i \cdot php_{i7} [(Tc - pTc_{i1})\exp\{-php_{i8}(Tc - pTc_{i1})\}] \\ \qquad \text{if } rp > prp_{2i} \text{ and } Tc > pTc_{i1} \qquad [\text{case 3}] \\ 0 \qquad \text{if } Tc \le 0 \qquad [\text{case 4}] \end{cases} \tag{5.5}$$

The four cases in equation (5.5) correspond to a categorization of the sunlight and canopy temperature status. Case 1 is for normal daytime and moderate temperatures; case 2 is for low sunlight or darkness and moderate temperatures; case 3 is for excessively high temperature with normal daylight; and case 4 is for freezing temperatures. Fig. 5.1 has the corresponding curves marked. This mechanism is also explained in Bledsoe and Jameson (1969).

The parameters in equation (5.5) consist of unit interchange and scaling coefficients and three threshold values (compensation points) for temperature and sunlight (prp_{2i}, pTc_{i1}, and pTc_{i2}). The compensation point for a variable is the level of that variable at which gross photosynthesis exactly equals respiration, giving a net photosynthesis rate of zero. Tc denotes canopy temperature (see above, section 4) and rp denotes photosynthetically active radiation. Percent cover and leaf area index were taken as unity for the results given in Appendix E, except for *Opuntia* where $la_4 = .15$.

The ISV rp, photosynthetically active radiation, is calculated solely as a function of the driving variable solar radiation, z_1. The idea is that the geometry of the grass leaves is such that a greater amount of radiation will be intercepted per unit soil surface than a horizontal plane would intercept for the solar position at other than the zenith. The driving variable z_1 describes the positive portion of a sine curve during the daylight hours. However, rp will describe a curve given by a fractional power of a sine curve as shown in equation 5.6 and Fig. 5.2. This mechanism is explained in Bledsoe and Jameson (1969).

$$rp = pz_1 \cdot \left(\frac{z_1}{pz_1}\right)^{prp_1} \qquad (5.6)$$

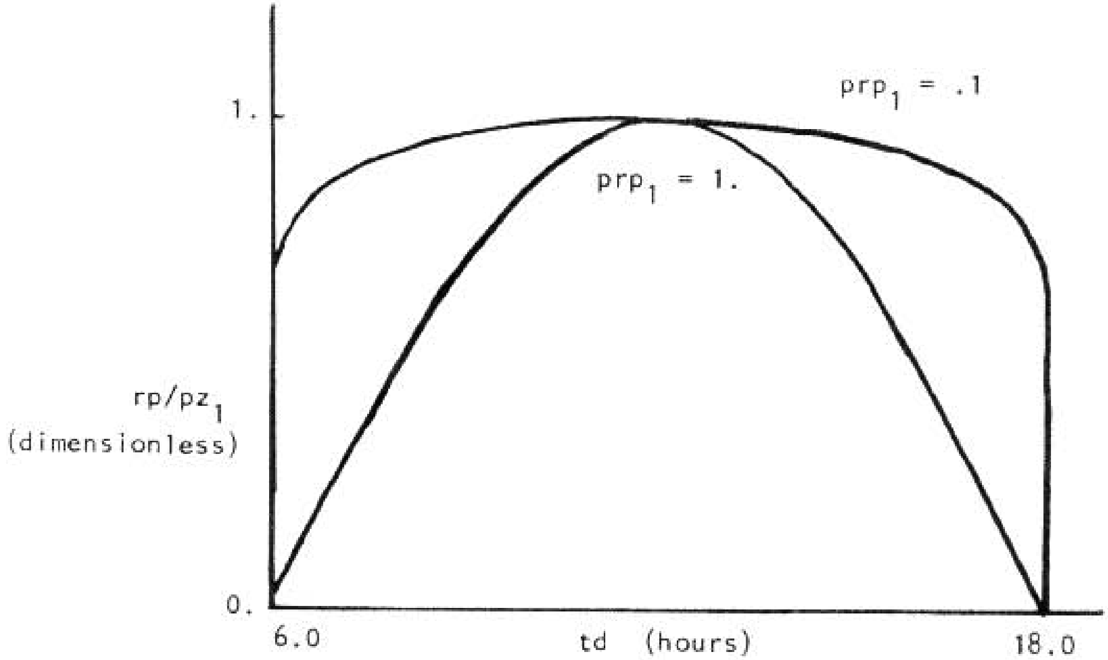


Fig. 5.2. Relative photosynthetically active radiation (rp/pz_1) as a function of time of day (td).

Growth response by the *ith* plant group to moisture stress is given by ms_i which is a function of the soil moisture by volume, Ms. It is formulated to imitate the growth response which might be observed as a function of soil moisture tension which would vary in soils of different types with a given volume fraction of water. It is also designed to take into account differential plant tolerances to high moisture tension. The formulation is designed to exhibit a semi-threshold response of varying degrees of sharpness. Equation 5.7 and 5.8 and Fig. 5.3 show the mechanism.

$$ms_i = at(pms_{i2}, pms_{i1}, fw) \tag{5.7}$$

where

$$fw = \frac{(\frac{Ms}{20.})}{1.4} \tag{5.8}$$

the ISV, fw, is soil moisture converted to fraction by weight (in top 20 cm of soil) assuming a bulk density of 1.4. The function (ISV) $at(x_1, x_2, x_3)$ gives an arc tangent response to its independent variable as follows.

$$at(x_1, x_2, x_3) = (\frac{1}{\pi}) \tan^{-1}[c_1(x_3 - x_1)] + .5 \tag{5.9}$$

$$c_1 = \tan(.4 \cdot \frac{\pi}{x_2}) \tag{5.10}$$

This ISV is a mechanism for calculating an index between zero and one which is dependent upon a semi-threshold effect utilizing an arc tangent function. Notice that the function is designed so that if the first argument, x_1, is the inflection point, the second argument, x_2, is a measure of spread or the distance from the inflection point to accomplish a rise of the index from .5 to .9, then the third argument is the variable for which the index is to be computed.

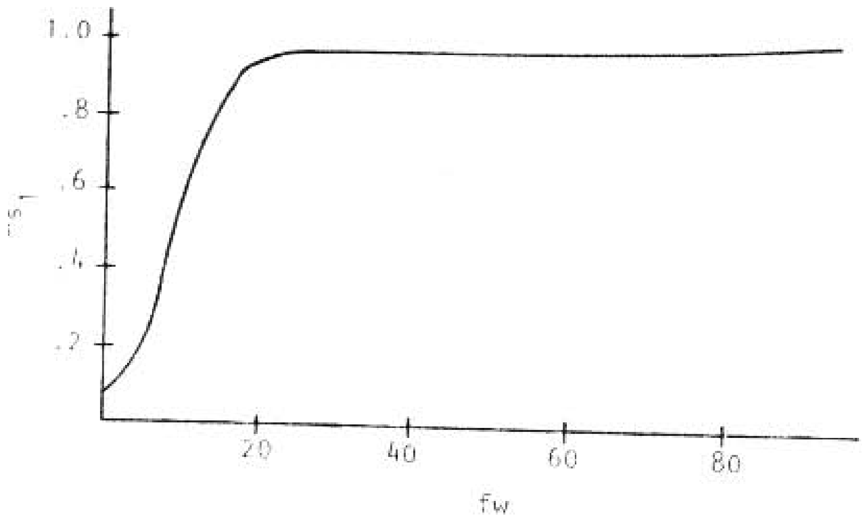


Fig. 5.3. Moisture stress index (ms_1) for the warm season grasses as a function of percent soil moisture by weight (fw) in the top 20 cm of soil.

The nutrient stress index, ns_i, also utilizes the arc tangent relation with a weighted sum of decomposer PSV's as its argument. Details of this ISV are given in section 7.

The respiration of belowground plant parts, rb, is assumed to be a function of soil temperature. Fig. 5.4 shows this relationship which is formulated in equation (5.10).

$$rb = \begin{cases} prb_1(44 - Ts)\exp\left(\frac{-(44 - Ts)}{8}\right) & \text{if} \quad Ts \leq 44 \\ 0 \quad \text{otherwise} & \end{cases} \tag{5.11}$$

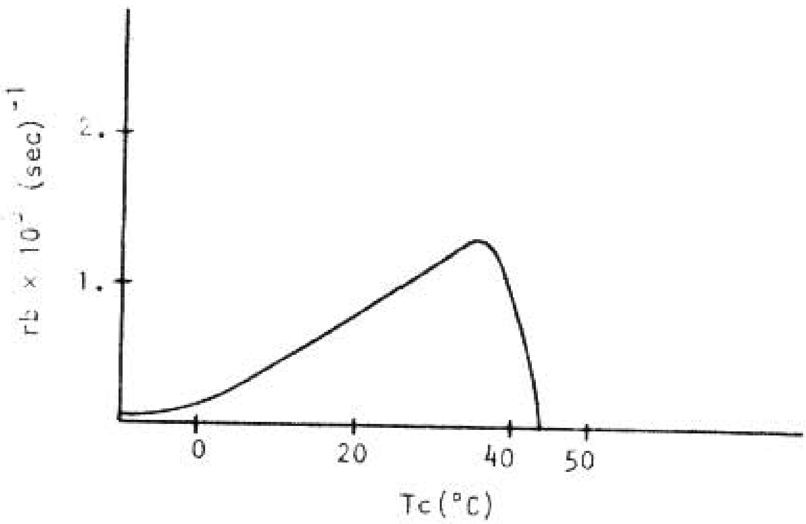


Fig. 5.4. Rate of respiration of belowground plant parts (rb) vs. soil temperature (Tc).

It is assumed that 36° is a temperature value above which the belowground plant respiration decreases precipitously. The rate, rb, is given in $g/(cm^2 \cdot sec)$ per g/cm^2 of belowground live plant material (thus rb has units sec^{-1}). The maximum respiration rate under this assumption is about 5% of the mean root biomass (10^4 g/cm^2) per day.

The death rate of belowground plant material, db, is taken to be half the respiration rate as an initial assumption.

$$db = \frac{rb}{2} \tag{5.12}$$

This rate is in $g/(cm^2 \cdot sec)$ per g/cm^2 of belowground live plant tissue.

The ISV, da_i, death rate of aboveground parts, is calculated as a function of the moisture stress index, ms_i, and the canopy temperature, Tc, as in equation 5.11.

$$da_i = (1 - ms_i)pda_{i1} \cdot at(pda_{i3}, pda_{i2}, Tc) \cdot Al_i \tag{5.13}$$

Fig. 5.5 shows da_1 as a function of canopy temperature for limiting soil moisture (ms_1 = 0.). For higher levels of soil moisture the curve would have similar shapes but proportionally lower values.

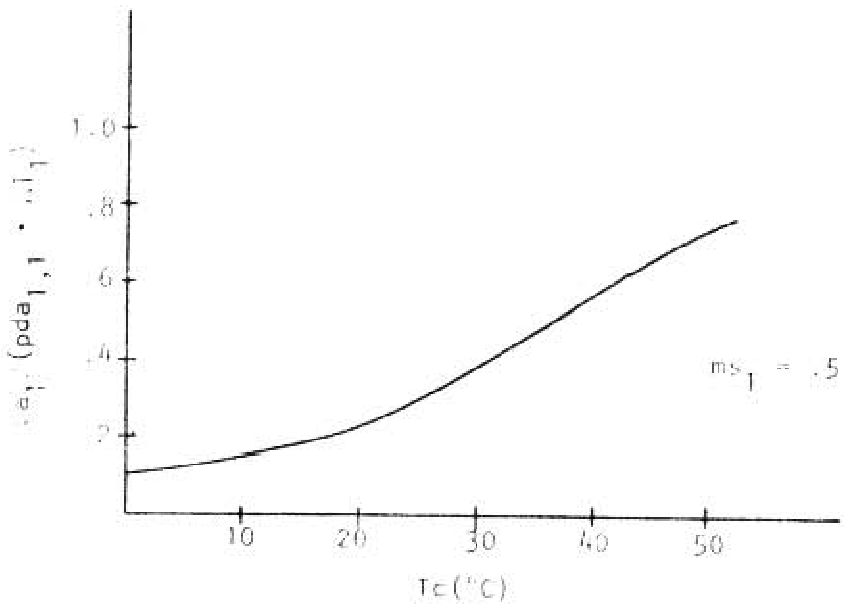


Fig. 5.5. Death rate of warm season grasses (Al_1) as a function of canopy temperature (Tc).

The translocation from above- to belowground plant parts, bt_i, is given in this model as a function of both the above- and belowground plant biomasses. The total specific aboveground plant biomass is denoted by ab, and is equal to the sum over i of Al_i. Then the translocation rate in g/(cm^2·sec) is given in equation (5.13).

$$bt_i = ms_i \cdot \begin{cases} 0. & \text{if } Ts \le 0 \\ \max[0., (pbt_{i1} - pbt_{i2} \cdot bl_i)Al_i] & \text{if } (Al_i > pbt_{i3}) \text{ and } (Ts > 0) \\ -\min[0., pbt_{i4}(bl_i - pbt_{i5}) & \text{if } (Al_i \le pbt_{i3}) \text{ and } (Ts > 0) \end{cases} \qquad (5.14)$$

where

$$bl_i = Bl \cdot \frac{Al_i}{ab} \qquad (5.15)$$

and

$$ab = \sum_{i=1}^{4} Al_i \qquad (5.16)$$

This appears graphed in Fig. 5.6. The ISV bl_i is a device for estimation of the belowground biomass plant group. Since only the total belowground plant biomass is known (Bl), it is assumed that the belowground plant parts are in the same ratio as aboveground parts.

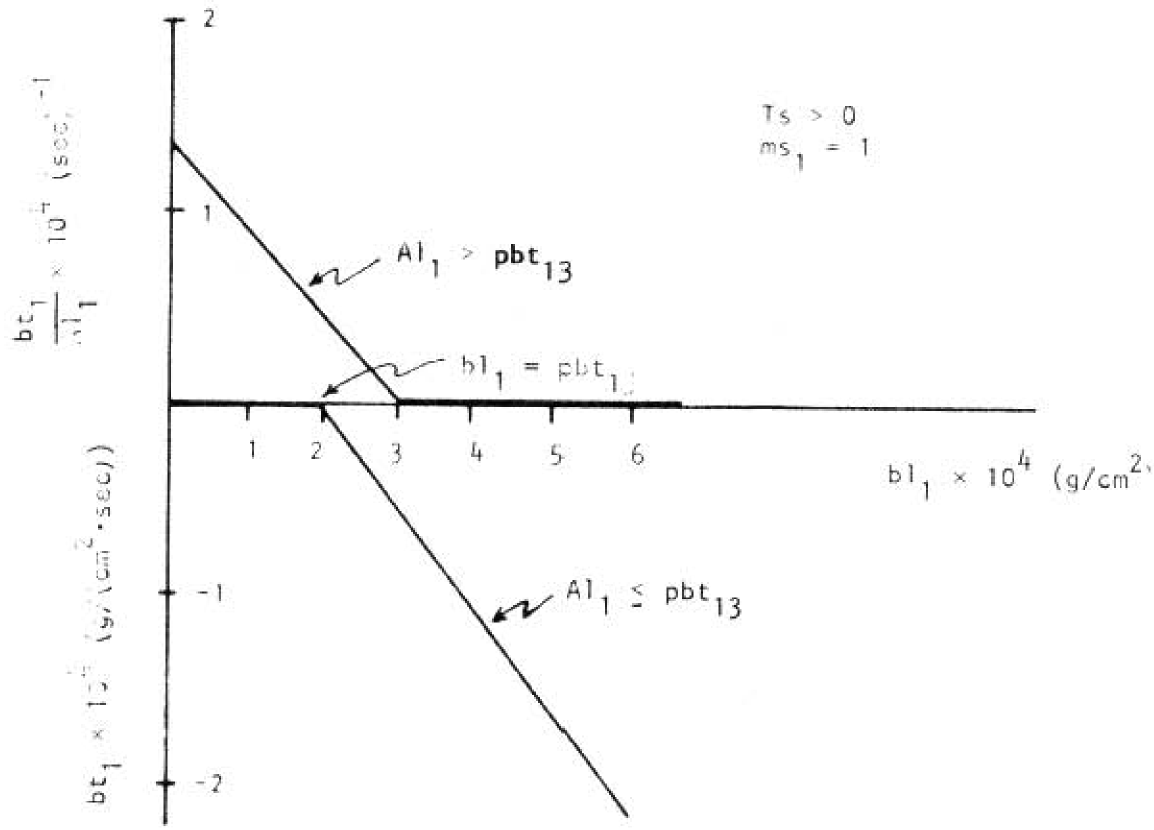


Fig. 5.6. Translocation (bt_i) of photosynthate from aboveground to below-ground plant parts for warm season grasses. In this graph, pbt_{i1} – pbt_{i5} have values 1.3×10^{-4}, $.43 \times 10^{-4}$, 3.0×10^{-4}, $.27 \times 10^{-4}$, and 2×10^{-4}, respectively.

The shatter rate, sa, is given as a piecewise linear function of wind speed. For wind speed less than 5 mph (223.5 cm/sec) the shatter rate is constant at $.005(\text{sec})^{-1}$. Above this wind speed the shatter rate increases linearly.

$$sa = \begin{cases} .005 & \text{if } z_4 \leq 223.5 \text{ cm/sec} \\ (.005 + 2.21 \cdot 10^{-4} \cdot z_4) & \text{if } z_4 > 223.5 \end{cases} \tag{5.17}$$

The values used are based on an assumed shatter rate of .5% per day with a 5 mph wind. The shatter rate is graphed in Fig. 5.7.

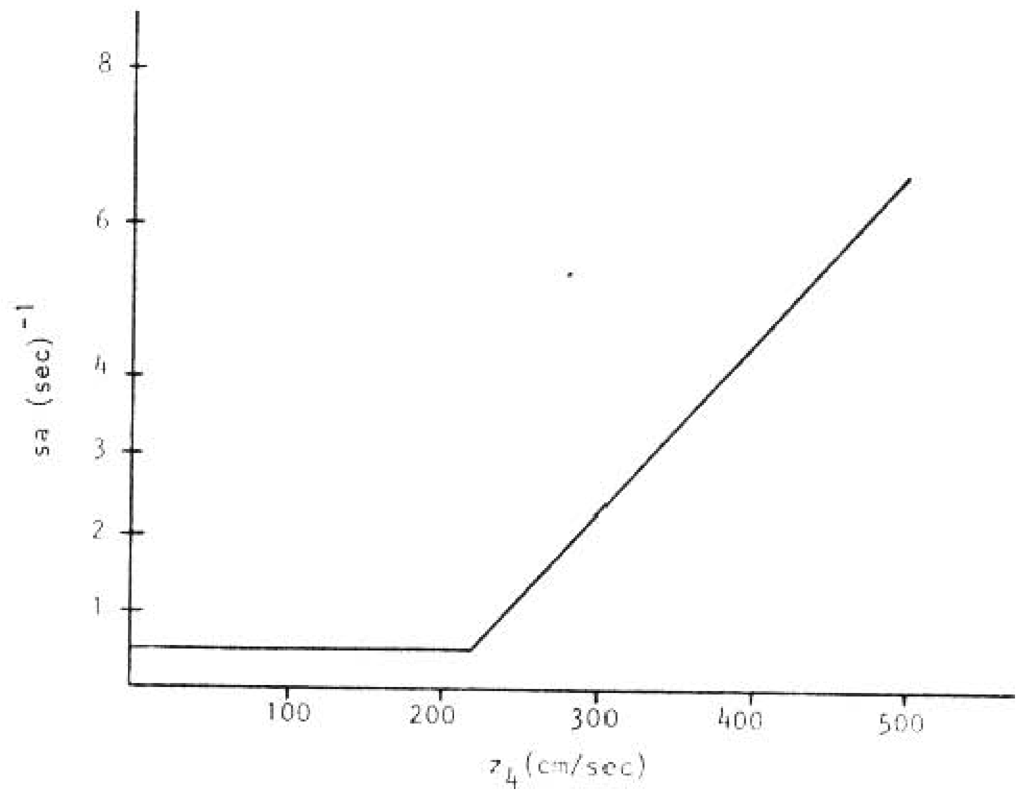


Fig. 5.7. Shatter rate of standing dead plant material (sa) as a function of wind speed.

The leaching rate, lc, is assumed to depend on the precipitation rate.

$$lc = .035 \cdot z_2 \tag{5.18}$$

where z_2 is the precipitation rate in g/(cm^2·sec). A graph of lc vs. z_2 appears in Fig. 5.8. (A rainfall rate of 5×10^{-5} cm/sec corresponds to a rain of about 1 cm/day.)

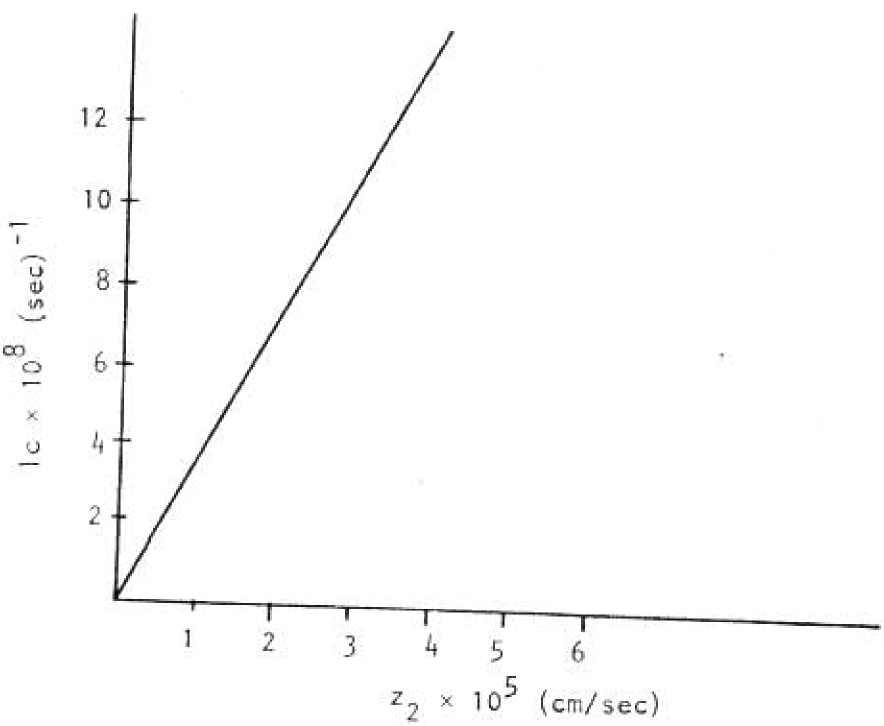


Fig. 5.8. Leaching rate of plant litter, lc, into soil organic matter as a function of precipitation (z_2). The leaching rate with .5 cm rain in a day would be about .5% per day of the total litter biomass. Harder rains would give proportionately more leaching.

Appendix B contains a summary of parameter definitions; Appendix E contains typical solution curves; parameter values are given in Appendix D.

5.3 Future Expansion

The possible areas for modification and expansion are so numerous that only a cursory catalog of the most obvious will be attempted. The first modifications will probably involve redefinition of the PSV's so that the major plant species have their own functional group. The belowground and dead biomasses will probably also be subdivided by functional group. A dead root compartment might be added. Separate PSV's for reproductive parts of plants might be added. Additional ISV's will include a microbial harvest term to account for in situ decay from each PSV, a trampling loss term for all aboveground biomasses, and a shattering term for live plants.

Modifications of the formulation of ISV's might include addition of the photosynthesis rate as an argument of the translocation function, as indicated by an experimental project in the Botany Department of the University of Wyoming. Belowground respiration might be made a function of other soil variables than temperature; photosynthesis might be broken into gross photosynthesis minus respiration; an actual soil moisture tension ISV might be added; shatter rates might be made to depend on snow cover and storm intensities; leaf area index should be made a function of solar angle and biomass density with concomitant changes in photosynthetically active radiation; aboveground death rate might be modified to consider phenological data; leaching rates should consider the previous occurences of rainfall; nutrient and moisture stress indices should be formulated to give more specific responses than a simple reduction in rate of occurrence of physiological functions; nutrient stress should give differential responses for shortage to different nutrient forms; allowances should be made for soil toxicities, such as salinity. Obviously, this list could be extended. The modification plan, as explained in section 2, will involve the gradual inclusion of the above factors rather than an attempt to rewrite the entire section and include everything. The decision as to what changes to make first will be a difficult one and the choice will necessarily be arbitrary sometimes.

For a longer view, the model will need to be a spatial rather than point system. The ISV, cr_i, percent cover of the *ith* plant group contains the beginnings of this extension. Currently, cr_i has a constant value and properly should be renamed as a parameter. Changing cr_i to a PSV and representing it by a differential equation will allow spatial growth of plant groups. Constraints can be placed on the sum of the cr_i values so that they do not exceed 100% cover. Some mechanisms of this type can be found in Clymer and Bledsoe (1970). The spatial model can be made more realistic by superimposing it on the plant communities of a fine scale vegetation map of an experimental area. Such maps have been and are being produced for various areas at the Pawnee Site by members of the Range Management Department of the University of Wyoming.

6. CONSUMER VARIABLES

6.1 Principal System Variables

The biomass density of seven functional groups is represented in the consumer section of the total ecosystem model. Associated with each of the first five functional groups (live material) is the name of a representative species in that group, as shown in Table 6.1.

The units for variables in the consumer section differ slightly from other sections of the model in that a "day" rather than a "second" is the basic time unit. The computer implementation allows for this by multiplying all derivatives calculated by the number of days/second $(.864 \cdot 10^5)^{-1}$. Similarly, harvest rates calculated in other program sections and supplied to the consumer section are multiplied by $.864 \cdot 10^5$.

Let

N_i = density of individuals in group i $(cm^2)^{-1}$

W_i = average weight of an individual in group i (g)

C_i = biomass density of group i (g/cm^2)

Then in order to compute the biomass of the first five functional groups, representing live biomass, at any time, differential equations are solved for N_i and W_i. Then the algebraic formulation in equation (6.1) gives total biomass densities.

$$C_i = N_i W_i; \quad i = 1, \ldots, 5 \tag{6.1}$$

Table 6.1. Consumer functional groups.

Functional Group	Representative Species	Population Numbers	Avg. Weight	Biomass
Primary consumer--wild	Black-tailed jackrabbit	N_1	W_1	C_1
Primary consumer--domestic	Cow	N_2	W_2	C_2
Secondary consumer	Coyote	N_3	W_3	C_3
Omnivore--migratory	Lark bunting	N_4	W_4	C_4
Insect	Grasshopper	N_5	W_5	C_5
Total animal feces	--	--	--	Fe
Total animal dead material	--	--	--	Cd

The differential equation for the rate of change of animal density with respect to time is written in the following form:

$$\dot{N}_i = N_i[nb_i - nd_i] - \frac{ha_{i+7}}{W_i} + im_i - em_i; \quad i = 1, \ldots, 5 \qquad (6.2)$$

where

nb_i = instantaneous birth rate of consumer group i $(day)^{-1}$

nd_i = instantaneous natural mortality rate of consumer group i $(day)^{-1}$

ha_{i+7} = harvest rate $(g/(cm^2 \cdot day)$ of consumer group i (in terms of biomass density)

im_i = immigration rate of consumer group i $(cm^2 \cdot day)^{-1}$

em_i = emigration rate of consumer group i $(cm^2 \cdot day)^{-1}$

Thus, the number of individuals in consumer group i is increased by birth and immigration and is decreased by natural mortality, harvest (predation), and emigration. In this model, all of the above mentioned rates are assumed to be functions of time only and are not influenced by such factors as the numbers of individuals in other consumer functional groups or the biomass composition of the various primary producer functional groups. This simplification was for the priority reasons mentioned in other sections and at the beginning of the report.

The differential equation for the rate of change of the average weight of an individual in consumer group i is written in the following form:

$$\dot{W}_i = W_i \cdot as_i \cdot fc_i - re_i; \quad i = 1, \ldots, 5 \qquad (6.3)$$

where

as_i = assimilation efficiency of the food ingested by an individual in consumer group i

fc_i = food intake rate $(day)^{-1}$ for an individual in consumer group i

re_i = respiration rate (g/day) for an individual in consumer group i

The differential equation for the rate of change of the biomass of animal feces with respect to time is written in the following form:

$$\dot{Fe} = \left[\sum_{i=1}^{5} C_i \ (1 - as_i) fc_i\right] - ha_{13} \tag{6.4}$$

where

ha_{13} = harvest rate $(g/(cm^2 \cdot day))$ of animal feces.

Thus, the animal feces compartment is simply an accumulation of material ingested but not assimilated by the five consumer groups.

The differential equation for the rate of change of animal dead material with respect to time is written as

$$\dot{Cd} = \left[\sum_{i=1}^{5} C_i \cdot nd_i\right] - ha_{14} \tag{6.5}$$

where

ha_{14} = harvest rate $(g/(cm^2 \cdot day))$ of animal dead material.

Thus, the animal dead compartment is simply an accumulation of the biomass of the animals dying from natural causes (non-predation) in the five consumer groups.

6.2 Intermediate System Variables

Birth, death, immigration, emigration. The instantaneous birth rates (nb_i), natural mortality rates (nd_i), immigration rates (im_i), and emigration rates (em_i) are determined for the first five consumer groups in the following manner. Note that immigration and emigration are modelled only in consumer group 4 (birds), i.e., $im_i = em_i = 0.$ for $i = 1, \ldots, 5$, except $i \neq 4$.

i. The percent of functional group 1 (primary consumer--wild) reproducing during the year decreases linearly from 100% at day 58 (start of the breeding season for a 365-day year) to 10% at the end of the breeding season. This mechanism is to lump all restrictions on breeding (e.g., sex ratio, fecundity) into a single factor. The instantaneous natural mortality rate is restricted so that a net increase in the population size occurs from day 58 to day 210, and a net decrease in the population size occurs after day 210. The reason for changing (decreasing) the proportion of the population reproducing during the breeding season is that young individuals recruited into the population due to birth do not reproduce.

Thus,

$$nb_1 = \begin{cases} 0 & \text{if } (tdm < 58) \text{ or } (tdm > 252) \\ .007[1 - .0046(tdm - 58)] & \text{if } 58 \leq tdm \leq 252 \end{cases} \tag{6.6}$$

$$nd_1 = .0022 \tag{6.7}$$

ii. Consumer group 2 (primary consumer--domestic) is kept at a fixed density, as if stocking rates were not changed and animals neither died nor gave birth on the range.

$$nb_2 = nd_2 = 0 \tag{6.8}$$

iii. Due to a lack of information on secondary consumer population dynamics, consumer group 3 is kept at a fixed density, representing an estimated average value.

$$nb_3 = nd_3 = 0 \tag{6.9}$$

iv. The birth, natural mortality, immigration, and emigration rates of consumer group 4 (birds) mimic those simulated by Swartzman (1969) for lark buntings in Technical Report No. 3. Thus, hatching occurs during weeks 23, 24, 27, and 28. Birds immigrate at a rate of 26 birds/100 ha/day during weeks 17 and 18 (days 113 through 126) and emigrate at a rate of 21 birds/100 ha/day after week 31 (day 218).

Thus,

$$nb_4 = \begin{cases} .0308 & \text{if} \quad 155 \leq tdm \leq 168 \\ .0102 & \text{if} \quad 190 \leq tdm \leq 203 \\ 0 & \quad \text{otherwise} \end{cases} \tag{6.10}$$

$$nd_4 = \begin{cases} .0021 & \text{if} \quad tdm \leq 154 \\ .0035 & \text{if} \quad tdm > 154 \end{cases} \tag{6.11}$$

$$im_4 = \begin{cases} .26 \cdot 10^{-8} & \text{if } 113 \le tdm \le 126 \\ 0 & \text{otherwise} \end{cases} \tag{6.12}$$

$$em_4 = \begin{cases} .21 \cdot 10^{-8} & \text{if } 218 \le tdm \le 235 \\ 0 & \text{otherwise} \end{cases} \tag{6.13}$$

v. The percent of consumer group 5 (insects) reproducing during the year decreases linearly from 100% at day 162 (start of the breeding season) to 10% at day 224 (end of the breeding season). Natural mortality occurs at a constant instantaneous rate from day 162 to 259. The instantaneous natural mortality rate is restricted so that a net increase in the population size occurs from day 162 to day 208, and a net decrease in the population size occurs after day 208 and until day 259. This information was modified from preliminary reports of grasshopper population data taken by Biology Department personnel at the University of Colorado, Colorado Springs campus.

Thus,

$$nb_5 = \begin{cases} .0214[1 - .0145(tdm - 162)] & \text{if } 162 \le tdm \le 224 \\ 0 & \text{otherwise} \end{cases} \tag{6.14}$$

$$nd_5 = \begin{cases} .0071 & \text{if } 162 \le tdm \le 259 \\ 0 & \text{otherwise} \end{cases} \tag{6.15}$$

Enumeration of food sources. In order for computations of the above rates to be made, all possible sources of food for any consumer group must be enumerated. These sources of food are referred to as consumed groups and are listed in Table 6.2.

The first seven consumed groups refer to the seven principal system variables in the producer section of the model, and the last five consumed groups refer to the first five principal system variables in the consumer section of the model. Let cb_j = biomass of consumed group j; $j = 1, \ldots, 12$. Thus, $cb_8 = C_1$, $cb_5 = Bl$, etc.

In general, when a variable or parameter must be subscripted doubly to refer both to a consumer group and a consumed group, the first subscript will denote the consumer and the second will denote the consumed group.

Food intake rates. Ingestion involves calculation of two basic quantities relating to the quantity and quality of food eaten. The two basic ISV's are food intake, fc_i, and food preference, fp_{ij}, where i indicates the consumer group and j indicates the consumed group. We will treat these two interrelated quantities next.

The food intake rate (fc_i) for consumer group i, at any time, is controlled by two factors:

i. The average weight (W_i) of an individual in that group, and

ii. The amount of available food.

Let

pfc_{i1} = food intake rate $(\text{day})^{-1}$ for consumer group i under ideal food availability conditions

Table 6.2. Numbering of biomass PSV's for purposes of consumer section.

Index (j) of Consumed Group	Functional Name	PSV
1	Warm season grass	Al_1
2	Cool season grass	Al_2
3	Forb	Al_3
4	Cactus	Al_4
5	Belowground plant parts	Bl
6	Standing dead (plant)	Ad
7	Litter	Lt
8	Primary consumer--wild	C_1
9	Primary consumer--domestic	C_2
10	Secondary consumer	C_3
11	Bird	C_4
12	Insect	C_5

mp_i = index of most preferred food source for consumer group i under ideal food availability conditions

The former is an input parameter to the model; the latter is a model ISV which is explained below. mp_i is used to indicate an element of the array $[cb_j]$; thus it must be equated to a subscript.

The food intake rate (fc_i) for consumer group i, at a particular time, will be less than pfc_{i1} if any of the following conditions occur:

i. The average weight of an individual in consumer group i (W_i) is above a threshold weight (pdl_{i1}) for that group. Thus, if an individual gets too heavy, it consumes less food than normal and its weight is consequently reduced to within a specified range.

ii. The biomass of the most preferred source of food (consumed group k) under ideal food availability conditions, (cb_k), drops below a threshold (pb_{1k}). Thus, the biomass of the most preferred source of food has a direct effect on the amount of food consumed by a particular consumer group. The food intake rate for consumer group i (fc_i) will be greater than pfc_{i1} if the average weight (W_i) of an individual in consumer group i is below a second threshold weight (pwl_i) for that group, and more food is available for consumption. Thus, if an individual gets too light, it will consume more food, if available, than normal and its weight will increase to within a specified range.

Thus, if

k = most preferred source of food for consumer species i

= mp_i

$$d1_i = \begin{cases} pfc_{i1}\left(\dfrac{pdl_{i1}}{W_i}\right);\ W_i > pdl_{i1} \\ pfc_{i1};\ pdl_{i2} \le W_i \le pdl_{i1} \\ pfc_{i1}\left(\dfrac{pdl_{i2}}{W_i}\right);\ W_i < pdl_{i2} \end{cases} \qquad (6.16)$$

= dummy food intake variable

$$fc_i = \begin{cases} d1_i\ ;\ cb_k > pfc_{i2k} \\ d1_i\left(\dfrac{cb_k}{pb_{1k}}\right)\ ;\ pfc_{i3k} \le cb_k \le pfc_{i2k} \\ d1_i\left(\dfrac{pfc_{i3k}}{pfc_{i2k}}\right)\ ;\ cb_k < pfc_{i3k} \end{cases} \qquad (6.17)$$

pfc_{i3k} is a threshold which is explained in the next section.

The selection of the qualitative aspects of the diet, or functional groups consumed, are described below.

Let

pfp_{ij} = amount of the diet of consumer species i made up of consumed species j under ideal food availability conditions

Then

k = j such that $pmp_{ij} = \max\{pmp_{ij};\ j = 1, \ldots, 12\}$

Then

i. If the biomass of any consumed species j (cb_j) is below a fixed threshold (pfc_{i3k}), that consumed species is eliminated from the diet of all consumer species.

ii. If the biomass of the most preferred source of food (k) is below a fixed threshold (pfc_{i2k}), then the amount of the diet of consumer species i made up of consumed species k is reduced from what it would be under ideal food availability conditions.

Thus,

$$d2_{ik} = \begin{cases} pfp_{ik}; \quad cb_k > pfc_{i2k} \\ \\ pfp_{ik}\left(\dfrac{cb_k}{pfc_{i3k}}\right); \quad pfc_{i3k} \leq cb_k \leq pfc_{i2k} \\ \\ 0; \quad cb_k < pfc_{i3k} \end{cases} \tag{6.18}$$

= dummy food preference factor

$$d2_{ij} = \begin{cases} pfp_{ij}; \quad cb_j \geq pfc_{i3j} \\ \\ 0; \quad cb_j < pfc_{i3j} \end{cases} \qquad j \neq k \tag{6.19}$$

Finally, we normalize the proportions of consumed material,

$$fp_{ij} = \frac{d2_{ij}}{\sum\limits_{j=1}^{12} d2_{ij}} \quad , \tag{6.20}$$

So that

$$\sum_{j=1}^{12} fp_{ij} = 1.0. \tag{6.21}$$

The use of fp_{ij} in formulating differential equations for the PSV's comes in calculating the harvest ISV's. If ha_j is the harvest of the *jth* biomass group, then we have

$$ha_j = \sum_{i=1}^{5} hp_{ij} \tag{6.22}$$

where $$hp_{ij} = fc_i \cdot fp_{ij} \cdot C_i \tag{6.23}$$

Table 6.3 gives typical values for hp_{ij} corresponding to the simulation run explained in section 6.3.

Example

We shall calculate food preference and food consumption variables for consumer group 1 (Black-tailed jackrabbit) as an example.

i. pfc_1 = .200g/g body weight/day

j	pfp(1,j)
1	.05
2	.12
3	.70
4	.07
5	0.0
6	.059
7	.001
8	0.0
9	0.0
10	0.0
11	0.0
12	0.0

Table 6.3. Typical harvest rate values using parameters of Table 6.5 [g/(cm^2·day)].

hp_{ij}		j											
		1	2	3	4	5	6	7	8	9	10	11	12
	1	.134E-5	.321E-5	-	.187E-5	-	.158E-5	.241E-7	-	-	-	-	-
	2	.132E-5	.132E-5	-	.783E-7	-	.156E-6	.232E-7	-	-	-	-	-
i	3	-	-	-	-	-	-	-	.153E-8	-	-	.900E-10	.180E-8
	4	-	-	-	-	-	-	-	-	-	-	-	.378E-9
	5	.271E-5	.648E-5	-	.377E-5	-	.319E-5	.486E-7	-	-	-	-	-
ha_j		.537E-5	.110E-4	-	.572E-5	-	.493E-5	.959E-7	.153E-8	-	-	.900E-10	.218E-8

Equating this notation to the notation of the producer section of the model:

$$ha_5 = hb$$

$$ha_6 = hs$$

$$ha_7 = hl$$

ii. $k = 3$

Thus, under ideal food availability conditions, forbs make up 70% of the diet of black-tailed jackrabbits and are the most preferred source of food for that species.

iii. Assume $W_1 = 75. \leq pdl_{i2} = 100.$

Since the jackrabbit population is underweight, on the average, each individual will consume a larger than normal amount. Thus,

$$d1_1 = pfc_{11}\left(\frac{pdl_{i2}}{W_1}\right) = (.200)\left(\frac{100.}{75.}\right)$$

$$= .267 \text{ g/g body weight/day}$$

iv. The only source of food j whose biomass is below pfc_{i3j} is consumed species 3. Thus, $d2_{1,3}$ is set equal to zero.

j	$d2_{1j}$
1	.05
2	.12
3	0.0
4	.07
5	0.0
6	.059
7	.001
8	0.0
9	0.0
10	0.0
11	0.0
12	0.0

v. $cb_3 = .100E\text{-}3 \leq pfc_{1,3,3} = .200E\text{-}3$

Since the primary source of food for black-tailed jackrabbits is below both critical thresholds, the amount of food consumed by jackrabbits must now be reduced. Thus,

$$fc_1 = dl_1 \left(\frac{pfc_{i3,3}}{pfc_{i2,3}} \right) = (.267) \left(\frac{.200E-3}{.500E-3} \right)$$

$$= .107 \text{ g/g body weight/day}$$

Therefore, food consumption is ultimately reduced due to the inadequate supply of the primary source of food.

vi. The actual dietary composition is now computed.

j	fp_{1j}
1	.167
2	.400
3	0.0
4	.233
5	0.0
6	.197
7	.003
8	0.0
9	0.0
10	0.0
11	0.0
12	0.0

Assimilation

Each consumed group (j) is assumed to have a constant assimilation efficiency (pas_j) regardless of the consumer group (as given in Table 6.2). Thus, for any consumer group, i, the assimilation efficiency (as_i) of the food ingested at any time is simply the sum of the products of the assimilation efficiencies of the consumed groups and the fraction of the diet made up of the consumed groups at that time.

Thus,

$$as_i = \sum_{j=1}^{12} pas_j \cdot fp_{ij}; \quad i = 1, \ldots, 5 \tag{6.24}$$

Once the assimilation efficiencies have been computed, the food consumption rates can be calculated as

$$as_i \cdot fc_i \cdot W_i \quad .$$

Typical values are given in Table 6.4.

Respiration

The respiration equations are written in the following form:

$$re_i = pre_{i1} \cdot (W_i)^{pre_{i2}}, \quad i = 1, \ldots, 5 \tag{6.25}$$

The form of the equation is mechanistic to the extent that it is thought that the rate of respiration (basal metabolism) for an animal is approximately proportional to the surface area of that animal. The parameters for the respiration equations are given below in Table 6.5.

An estimate of $pre_{1,2}$ was taken from Hansen and Cavender (1970), of $pr_{2,2}$ from Kleiber (1961), of $pre_{4,2}$ from Swartzman (1970), and of $pre_{5,2}$ from Wiegert (1965). The value of $pre_{3,2}$ was guessed.

Table 6.4. Typical values of assimilation and intake rates.

i	as_i	$as_i \cdot fc_i \cdot W_i$ = intake
1	.382	3.07
2	.389	5630
3	.938	1690
4	.776	3.72
5	.382	0.012

Table 6.5. Typical values for respiration parameters.

i	pre_{i1}	pre_{i2}
1	0.54	0.67
2	2.10	0.75
3	3.80	0.67
4	0.28	0.75
5	0.13	0.84

The given values of pre_{i1} were determined so that under optimal food availability conditions

i. An individual in consumer group 1 will increase from 100 g to 125 g in 30 days,

ii. An individual in consumer group 2 (cow) will increase from 400 lb. (181,200 g) to 500 lb. (226,500 g) in 30 days,

iii. An individual in consumer group 3 will increase from 20 lb. (9,060 g) to 25 lb. (11,325 g) in 30 days,

iv. An individual in consumer group 4 will increase from 30 g to 35 g in 30 days, and

v. An individual in consumer group 5 will increase from .30 g to .35 g in 30 days.

6.3 Summary and Example

The consumer section of the model was simulated independently of the model by providing a record of forb aboveground biomass (Al_3 or cb_3) as an artificial driving variable. The graph of this driver is shown in Fig. 6.1(a). Table 6.6 summarizes the initial conditions and parameter values used in the simulation. It was designed to allow investigation of the consumer response to a limiting food supply. Accordingly, the food preference factors (pfp_{ij}) were arranged so that forbs are the most preferred food source for all the herbivorous consumers. Table 6.7 shows the rate of change of the individual average weights (W_i, i = 1, . . ., 5) for the initial conditions of the simulation. For the parameter and initial conditions

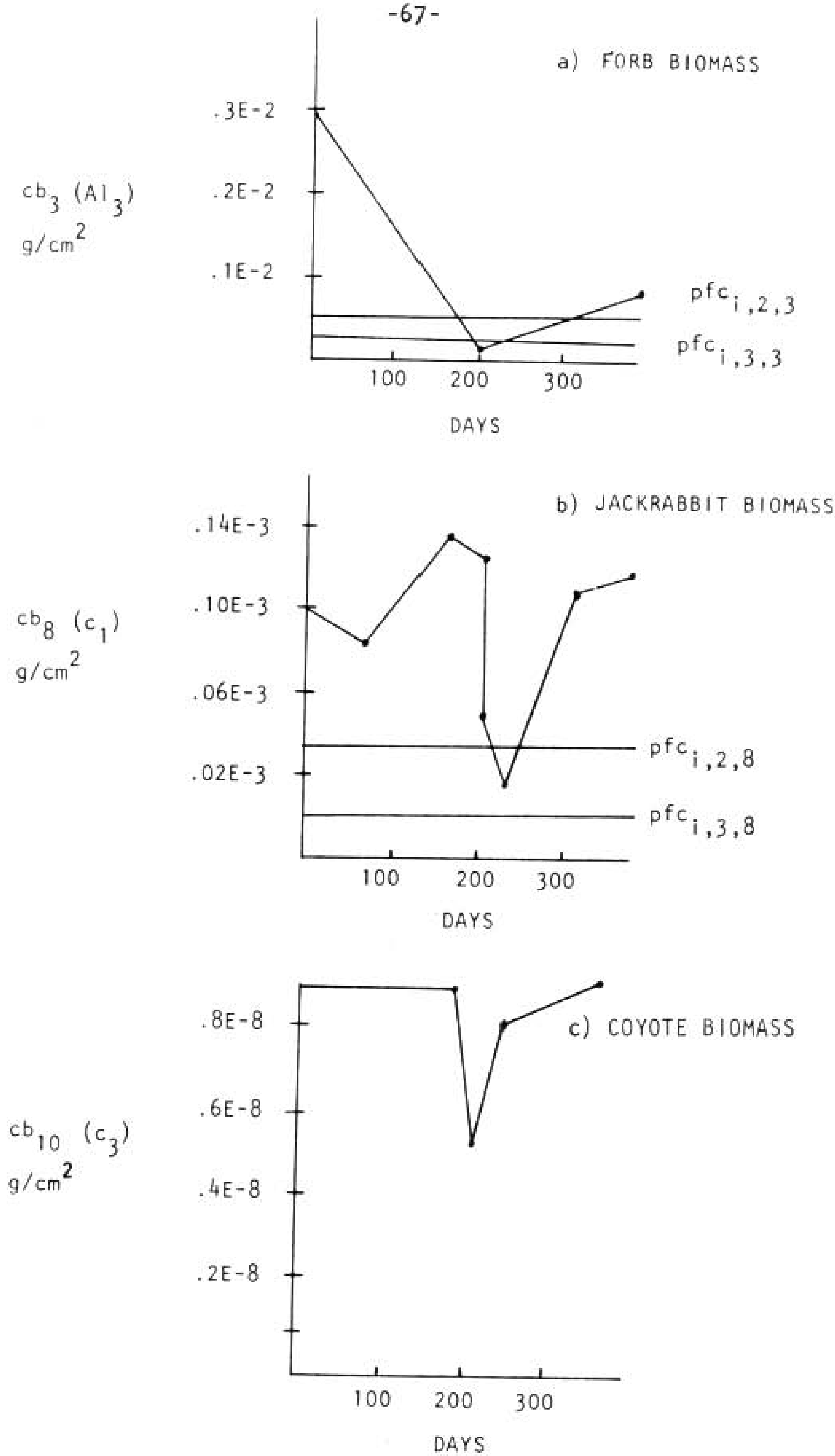
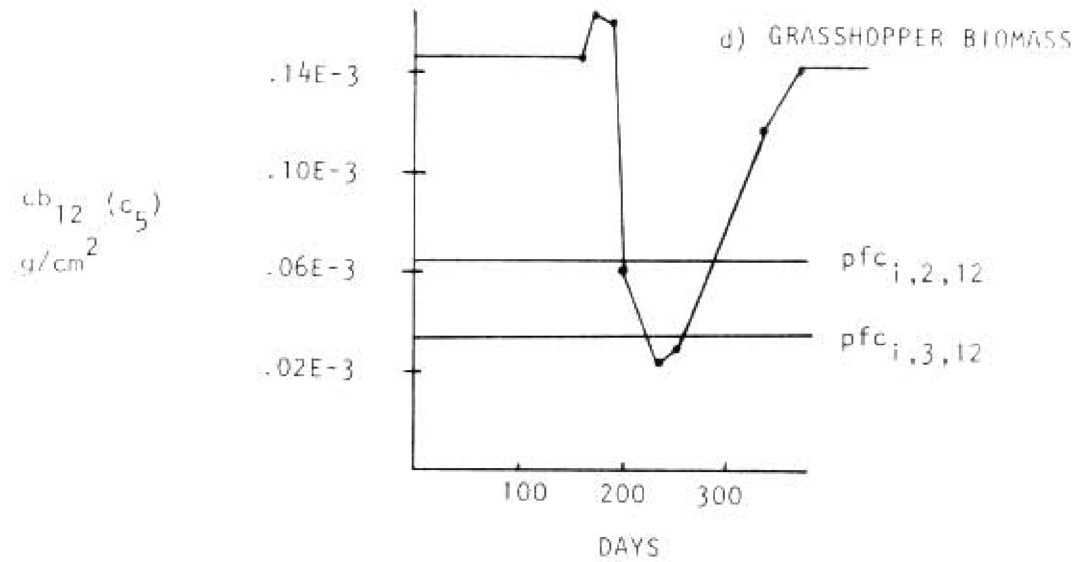


Fig. 6.1. Results of 365 day simulation of consumer submodel. a) Forb artificial driving function, b) through e) biomass densities of consumers.


d) GRASSHOPPER BIOMASS
cb12 (c5)
g/cm2
.14E-3
.10E-3
.06E-3
.02E-3
pfci,2,12
pfci,3,12
100
200
300
DAYS

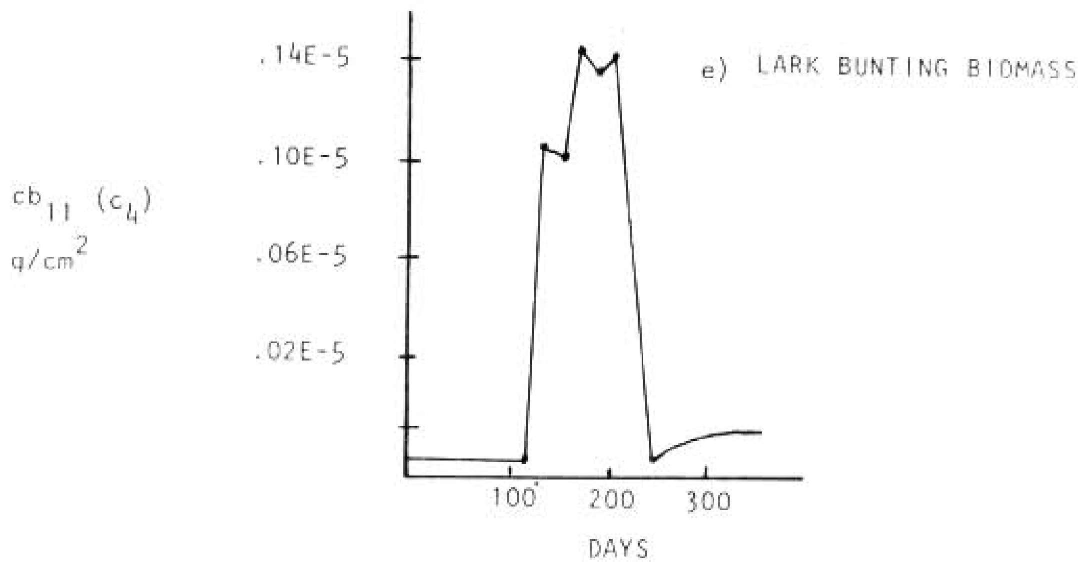

e) LARK BUNTING BIOMASS
cb11 (c4)
g/cm2
.14E-5
.10E-5
.06E-5
.02E-5
100
200
300
DAYS

Fig. 6.1. (cont.)

Table 6.6. Parameter, PSV, and ISV values for a sample solution with dummy producer values (td = 0).

Consumed Species i	Initial Conditions			Parameter Values		
	N_i #/cm^2	W_i g/ind.	C_i g/cm^2	pwu_i g/ind.	pwl_i g/ind.	pfc_i g/g/day
1	.100E-5	.750E+2	.750E-4	.100E+3	.100E+3	.200
2	.200E-9	.181E+6	.362E-4	.181E+6	.181E+6	.200
3	.100E-11	.900E+4	.900E-8	.900E+4	.900E+4	.200
4	.100E-9	.300E+2	.300E-8	.300E+2	.300E+2	.160
5	.500E-9	.300	.150E-3	.300	.300	.270

Consumed Species j	Parameter and ISV Values			
	cb_j g/cm^2	$pfc_{i,3,j}$ g/cm^2	$pfc_{i,2,j}$ g/cm^2	pas_j
1	.300E-2	.100E-2	.200E-2	.300
2	.100E-2	.100E-3	.200E-3	.500
3	.100E-3	.200E-3	.500E-3	.700
4	.300E-2	.100E-2	.200E-2	.350
5	.600E-1	.141E-1	.282E-1	.000
6	.100E-1	.230E-2	.470E-2	.250
7	.500E-2	.120E-2	.240E-2	.200
8	.750E-4	.125E-4	.250E-4	.950
9	.362E-4	0	1.0	.000
10	.900E-8	0	1.0	.000
11	.300E-8	.270E-6	.540E-6	.900
12	.150E-3	.280E-4	.750E-4	.850

Table 6.6. (Continued)

pfp_{ij} - Food Preference Under Ideal Food Availability

Species i	j: 1	2	3	4	5	6	7	8	9	10	11	12
1	.05	.12	.70	.07	-	.059	.001	-	-	-	-	-
2	.25	.25	.45	.015	-	.03	.005	-	-	-	-	-
3	-	-	-	-	-	-	-	.85	-	-	.05	.10
4	-	.175	.175	-	-	-	-	-	-	-	-	.65
5	.05	.12	.70	.07	-	.059	.001	-	-	-	-	-

Computed Values

i	Primary Source of Food mp_i	Food Consumption Rates fc_i (g/g body weight/day)
1	3	.107
2	3	.080
3	8	.200
4	12	.160
5	3	.108

Table 6.6. (Continued)

Food Preference - fp_{ij}; $j = 1, \ldots, 12$; $i = 1, \ldots, 5$

Consumer Species i	j											
	1	2	3	4	5	6	7	8	9	10	11	12
1	.167	.400	-	.233	-	.197	.003	-	-	-	-	-
2	.455	.455	-	.027	-	.055	.008	-	-	-	-	-
3	-	-	-	-	-	-	-	.850	-	-	.050	.100
4	-	.212	-	-	-	-	-	-	-	-	-	.788
5	.167	.400	-	.233	-	.197	.003	-	-	-	-	-

given in Table 6.6, we get the following values (Table 6.7) for the change in the average weight of an individual in consumer group i; $i = 1, \ldots, 5$. Thus, the consumer groups whose primary source of food is not available lose weight while the other two consumer groups remain fairly constant.

A one-year (365 day) run of the consumer submodel was made. The producer variables $\{cb_1, \ldots, cb_7\}$ are the driving forces for the run. All producer variables with the exception of cb_3 (forbs) are kept at constant biomass values, above all critical thresholds, throughout the run. The forb biomass decreases linearly from 0.003 g/cm^2 at time 0 to 0.0002 g/cm^2 at time 200, and then increases linearly to 0.0007 g/cm^2 at time 365. Thus,

(i) $cb_3 < pfc_{i,2,3} \qquad 175 \leq t \leq 310$

(ii) $cb_3 < pfc_{i,3,3} \qquad 193 \leq t \leq 210$

Plots of the biomasses of jackrabbits cb_8, coyotes cb_{10}, grasshoppers cb_{12}, and lark buntings cb_{11} are made against time in Fig. 6.1. It should be noted that, under ideal food availability conditions, the primary source of food for jackrabbits and grasshoppers is forbs, and that the primary sources of food for coyotes and lark buntings are jackrabbits and grasshoppers, respectively. Thus, the forb biomass is the primary controlling factor on the whole system. It is interesting to note that as soon as the forb biomass (cb_3) drops below $pfc_{i,2,3}$, both the jackrabbit and grasshopper biomasses drop. When forbs are completely removed from the diets of jackrabbits and grasshoppers ($cb_3 < pfc_{i,2,3}$), those consumer biomasses decrease at a more rapid rate. As soon as forbs are added back into the diets of jackrabbits and grasshoppers ($cb_3 > pfc_{i,3,3}$), those consumer biomasses start to increase again. In a similar manner the coyote (secondary consumer) starts to decrease as soon

Table 6.7. Rates of change of individual weights for consumer simulation run with producer driving variable.

i	fn_i (g assimilated/ individual/day)	re_i (g respired/ individual/day)	$\dot{W}_i$ (g/individual/ day)
1	.307E+1	.974E+1	-.667E+1
2	.563E+4	.184E+5	-.128E+5
3	.169E+4	.169E+4	.000
4	.372E+1	.366E+1	.006E+1
5	.012	.047	-.035

as the jackrabbit biomass cb_8 drops below $pfc_{i,2,8}$, and the lark bunting biomass starts to decrease as soon as the grasshopper biomass (cb_{12}) drops below $pfc_{i,2,12}$. The lark buntings never completely recover because of the fact that their emigration starts during the period of rapid decline in biomass. Table 6.7 summarizes the harvest rates which occurred during the simulation.

6.4 Future Expansion

Expansion of the PSV's of this section will probably involve association of the variables with particular species rather than broad functional groups. This can certainly be done for small mammals and lagomorphs. Arthropods may remain grouped to some extent, e.g., grasshoppers and sapsuckers as two separate PSV's. Division of PSV's for certain of the consumers into sex and/or age classes will also be desirable. At first these would be only a simple yearling/adult or nymph/adult dichotomy but will be extended in future versions.

The basic ISV's, e.g., respiration, birth rate, etc., will probably remain as such, however, their formulation will be heavily modified. One serious fault of the model which goes across all consumers is the failure to provide mechanisms individually for the functional groups, i.e., the same general form of food preference or respiration function should not be applied across the board to all consumers with variations only in parameter values. Within the specific mechanisms of the model, here are a few areas to be investigated further.

The birth and natural mortality functions need to be tied into the biological and abiotic environment. This is an area where field data is extremely difficult to procure, however, every effort will be made to find sources of reliable information. It is expected that laboratory studies on such subjects as insect egg mortalities may be useful here. Stochasticization of the model (by the method mentioned in section 2.1) to simulate the random character of the population may be a useful alternative when birth and death functions are measurable with variability but no relation to environmental variables is possible. The introduction of density-dependent mechanisms for the birth and death functions is justified only when a specific case can be made based on literature or field data or as a hypothesis to be tested. We expect to have a library of such hypotheses available for substitution in PWNEE, MOD 2. Cattle stocking rates should also be entered as a driving variable for domestic herbivores.

The integro-differential equation approach to animal population dynamics (Bledsoe and Jameson 1969) is the most desired formulation from a standpoint of resolution. This mechanism, however, needs considerably more work as a separate model before it will be ready for inclusion in a whole system model. Both the methods of numerical analysis for the integral equations and the formulation of the required ISV's need further research.

The mechanisms of PWNEE concerned with the quality (i.e., dietary items) and quantity of food ingested are mathematically the most complex in the model. The basic fault of the current formulation is the failure to allow animals to shift their diet readily since ingestion rate is dependent only on the "most preferred species" rather than on total food availability. This single item could be corrected by a fairly simple "patch" to the model, however, we plan to take a broader look at the system. In particular, the array of possible food items should be extended to include additional items such as belowground plant parts, carrion, and feces, as well as supplemental food (as a driving variable) supplied by man. Bledsoe and Jameson (1969) contains a more complex mechanism for food choice based on a time varying dietary composition which can be keyed to plant conditions such as succulence and phenological state. Similarly, assimilation rates will be keyed to plant digestibility functions which will themselves be partially determined by environmental variables. A more complex mechanism involving digestive interactions among dietary items is also a candidate for inclusion.

The respiration function needs to be broadened so that a separate mechanism can be included for different animal types such as poikilotherms vs. homoiotherms. The introduction of an "activity cycle" (Bledsoe and Jameson 1969) for animals will allow a variable respiration rate depending upon animal activity as well as the external environment. This mechanism would also affect ingestion rates. The activity cycle would be basically diurnal with a seasonal variation imposed to allow for extra metabolic levels such as lactation, gestation, egg production, etc.

7. DECOMPOSER VARIABLES

The decomposer section of the PWNEE model consists of three groups of variables interfaced to the rest of the system. These three groups can be described as variables giving characteristics of the soil (So_i, i = 1, . . ., 3), variables giving the density per unit area of soil nitrogen in various forms (Mn_i, i = 1, . . ., 4), and variables giving an index to the activity per unit area of soil surface of various soil microorganisms (Mc_i, i = 1, . . ., 6). The general philosophy is that transformations among the nitrogen variables and some of the soil characteristic variables are regulated by the activity level of microbial populations which are, in turn, dependent on the substrate levels and the abiotic soil characteristics (temperature and soil moisture). Fig. 7.1 gives the names of the variables in the decomposer section and schematically depicts their interrelation.

7.1 Principal System Variables

There are six microbial functional types which comprise the PSV's symbolized by Mc_i as follows: i = 1, protein decomposer; i = 2, deaminifier; i = 3, nitrifier; i = 4, denitrifier; i = 5, nitrogen fixer; and i = 6, cellulose decomposer. The change in activity of each microbial type is determined fundamentally by a logistic equation whose variable coefficients are determined by the substrate levels for the particular microbe and by the abiotic variables, soil temperature (Ts), and soil moisture (Ms). The differential equation is given below:

$$Mc_i = dv_i \cdot \frac{Mc_i\ (mu_i - Mc_i)}{mu_i} \qquad (7.1)$$

where dv_i and mu_i are functions of other PSV's.

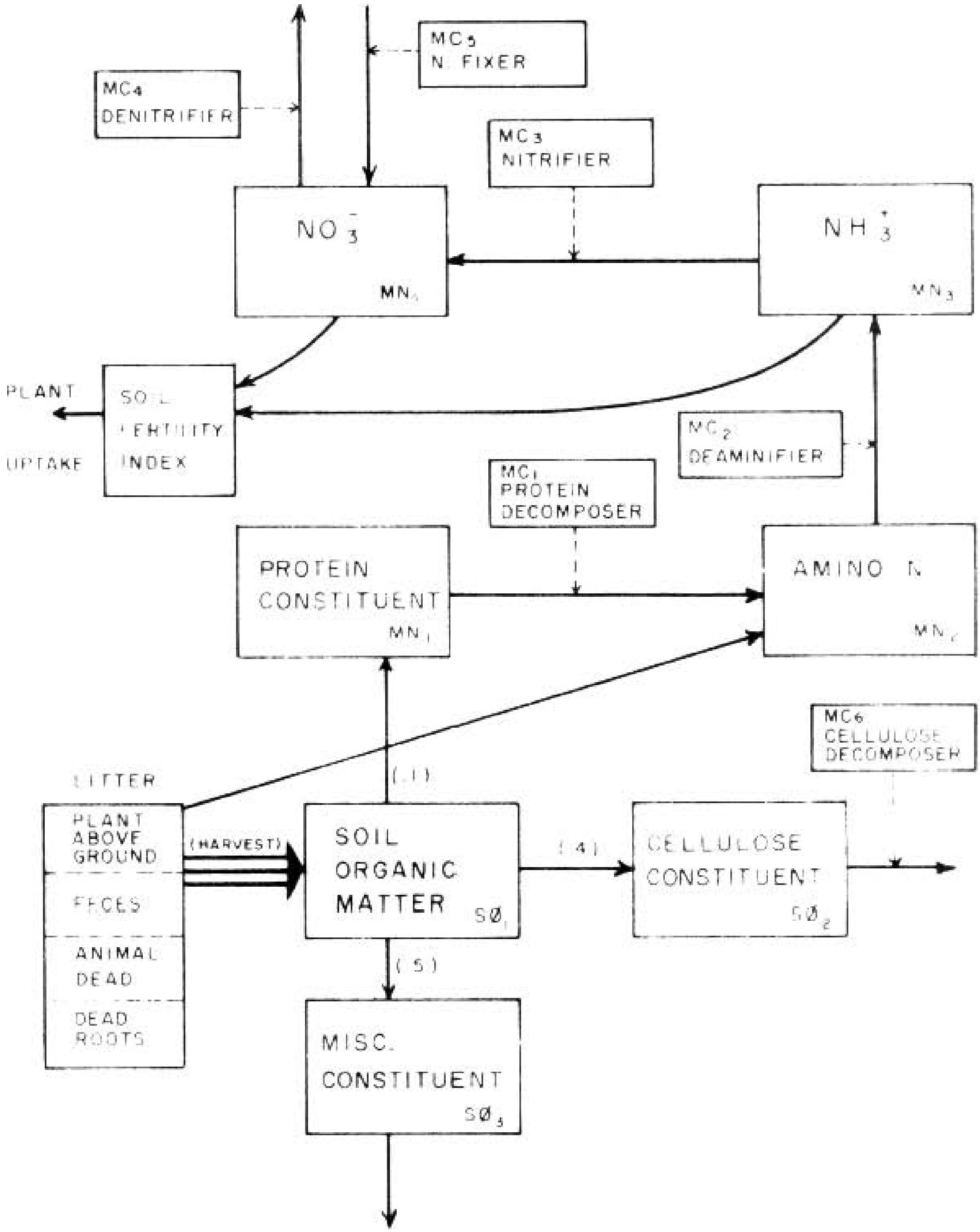


Fig. 7.1. Diagrammatic representation of decomposer section relations.

There are three soil characteristics (So_i) which are currently being utilized as follows: i = 1, soil organic matter total; i = 2, cellulose in soil organic matter; and i = 3, miscellaneous soil organic matter constituents. The first variable, So_1, is an algebraic sum of the other two soil variables and Mn_1 which is the soil insoluble organic nitrogen constituent (protein).

$$So_1 = So_2 + So_3 + Mn_1 \tag{7.2}$$

The philosophy of operation is that input into the soil organic matter compartment is via a microbial or physical harvest process from four other compartments of the ecosystem: dead animals (ha), animal feces (hf), belowground plant parts (db), and plant litter (hm). In each case the harvest rate is calculated as an intermediate system variable in subroutine ISV and each of the harvested variables has a variable exponential decay rate given by products of parameters and ISV's. The sum of all these harvests is characterized by a distribution of constituents given by array pSo_{1i}, i = 1, . . ., 3. For example, if the total detritus is characterized as being 10% protein and 40% cellulose, the remainder going to the SO_3 compartment, then $pSo_{1,1} = .1$, $pSo_{1,2} = .4$, and $pSo_{1,3} = .5$. Notice that

$$\sum_{i=1}^{3} pSo_{1i} = 1.0 \tag{7.3}$$

The cellulose content of soil organic matter is characterized by the following differential equation:

$$\dot{So_2} = pSo_{1,2}(hm + db + hf + ha) - pSo_{2,1} \cdot Mc_6 \cdot So_2 \tag{7.4}$$

which indicates an inflow proportional to the sum of the harvest variables, the proportion being given by the second element in the array pSo_{1i}. The loss is characterized by a decay rate, $pSo_{2,1}$, and is proportional both to the contents of the compartment and to the population level of the sixth microbial type which is responsible for cellulose decomposition.

The third soil characteristic variable, miscellaneous constituents, can be described by the following differential equation:

$$\dot{So_3} = pSo_{1,3}(hm + db + hf + ha) - pSo_{3,1} \cdot So_3 \qquad (7.5)$$

The input rate is again proportional to the sum of harvest variables; the decay rate is proportional to the contents of the compartment, with constant of proportionality $pSo_{3,1}$. Since there is no microbial type associated with this catchall soil compartment, it is characterized by a constant exponential loss rate.

The four species of soil nitrogen are as follows: Mn_1, insoluble organic nitrogen (protein); Mn_2, soluble organic nitrogen (amino nitrogen); Mn_3, soluble ammonia ion; and Mn_4, soluble nitrate ion. There is a differential equation for each of these variables which is characterized by a loss rate for each variable proportional to the contents of the compartment (amount of substrate) and the size of the microbial population responsible for transformation to the next nitrogen form. The *ith* nitrate species has a parameter associated with the loss, pMn_{i1}. The following differential equations describe the nitrate variables:

$$\dot{Mn}_1 = pSo_{1,1}(hm + db + hf + ha) - pMn_{1,1} \cdot Mc_1 \cdot Mn_1, \tag{7.6}$$

$$\dot{Mn}_2 = pMn_{1,1} \cdot Mc_1 \cdot Mn_1 + lc \cdot Lt - pMn_{2,1} \cdot Mc_2 \cdot Mn_2, \tag{7.7}$$

$$\dot{Mn}_3 = pMn_{2,1} \cdot Mc_2 \cdot Mn_2 - pMn_{3,1} \cdot Mc_3 \cdot Mn_3 - au(Mn_3, pt), \tag{7.8}$$

$$\dot{Mn}_4 = pMn_{3,1} \cdot Mc_3 \cdot Mn_3 + nf(Mc_5) - dn(Mn_4, Mc_4) - nu(Mn_4, pt). \tag{7.9}$$

7.2 Intermediate System Variables

For a given substrate level the relation of dv_i and mu_i to soil temperature and moisture is shown in Fig. 7.2. The algebraic formulation of dv_i and mu_i is given below:

$$dv_i = pMc_{i8} \cdot op(pMc_{i1}, pMc_{i2}, Ts) \cdot op(pMc_{i3}, pMc_{i4}, Ms) \cdot sb(pMc_{i5}, pMc_{i6}, d3_i) \tag{7.10}$$

$$mu_i = pMc_{i7} \cdot op(pMc_{i1}, pMc_{i2}, Ts) \cdot op(pMc_{i3}, pMc_{i4}, Ms) \cdot sb(pMc_{i5}, pMc_{i6}, d3_i) \tag{7.11}$$

where $d3_i$ is equated to the PSV which is the substrate for the *ith* microbial type. Thus,

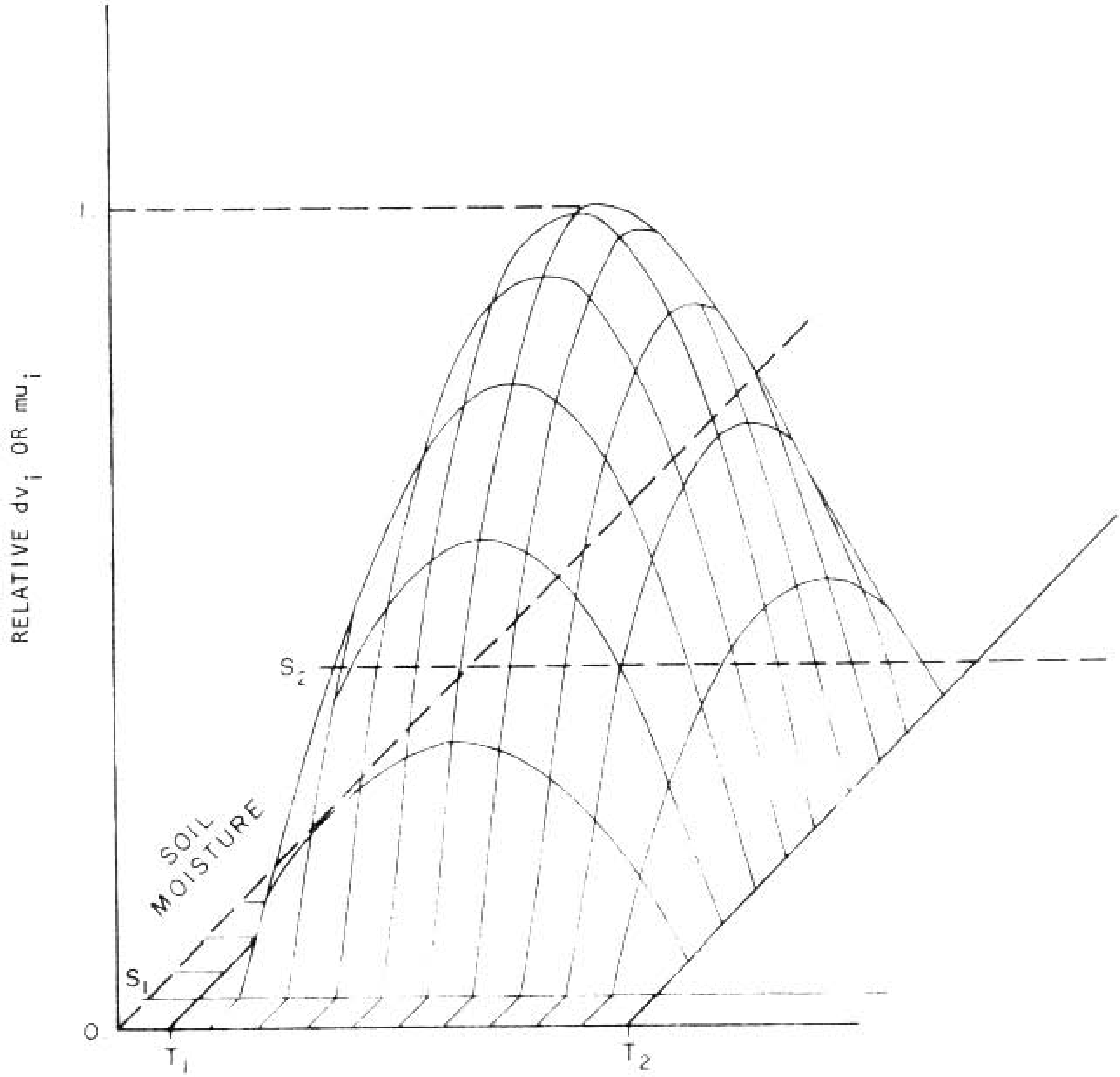


Fig. 7.2. dv_i and mu_i graphed as a function of temperature and soil moisture for a given substrate level.

$$\left.\begin{aligned} d3_1 &= Mn_1 \\ d3_2 &= Mn_2 \\ d3_3 &= Mn_3 \\ d3_4 &= Mn_4 \\ d3_5 &= So_1 \\ d3_6 &= So_1 \end{aligned}\right\} \tag{7.12}$$

The other ISV in equations (7.10) and (7.11) are defined below:

$$op(x_1, x_2, x_3) = \frac{-4.(x_3 - x_1)(x_3 - x_2)}{(x_2 - x_1)^2} \tag{7.13}$$

and

$$sb(x_1, x_2, x_3) = \begin{cases} .001 & \text{if} \quad x_3 \leq x_2 \\ \left.\begin{cases} 1. & \text{if} \quad x_1 = 0 \\ (x_3 - x_2) \cdot x_1 & \text{if} \quad x_1 \neq 0 \end{cases}\right\} & \text{if } x_3 > x_2 \end{cases} \tag{7.14}$$

Equation (7.13) gives a relation of any biological variable (equated to op) to an experimental gradient (x_3) in which the variable has an optimum for some value of the gradient and is essentially zero for much larger or much smaller values. x_1 and x_2 are the upper and lower bounds at which the biological variable is zero; halfway between x_1 and x_2 the variable reaches its optimum. The function is normalized to one, i.e., max(op) = 1. Equation (7.14) gives a relation of a biological variable (equated to sb) to an environmental gradient

(x_3) in which the variable rises linearly with the gradient after it passes a threshold value given by x_2. The slope of the rise is x_1. Fig. 7.3 shows the graph.

The differential equations for the four nitrate variables (equations 7.6 through 7.9) each contain one or more decay rates with positive or negative signs of the form

$$pMn_{i1} \cdot Mc_i \cdot Mn_i$$

There are two major inputs to the set of nitrogen variables, one given by the first term in equation (7.6), due primarily to death of plant roots, and the second given by the second term of equation (7.7), a leaching function, lc. The leach rate is described in the producer section as a loss of soluble nitrogen from the plant litter compartment due to rainfall.

In addition to the inputs and decay rates described above there is a loss from total soil nitrogen due to plant uptake as given by ISV's au(Mn_3, pt) and nu(Mn_4, pt). These functions give the rate of uptake by plants of ammonia and soluble nitrate, respectively. The uptake is proportional to the photosynthesis occurring at the time if photosynthesis is positive. The ionic uptake will be zero if photosynthesis is negative or zero. The ISV giving the total positive photosynthesis is pt. The theory here is that whenever a positive aboveground growth increment occurs a proportional amount of mineral nitrogen (nitrate or ammonia) must be removed from the soil. The proportionality constants are pnu_1 and pau_1, respectively. The uptake rate is proportional to the amount of ions present as well. The parameters pnu_1 and pau_1 are also indicative of the plants "preference" for one ionic species over another. The rate constant for ammonia will be several times larger than the

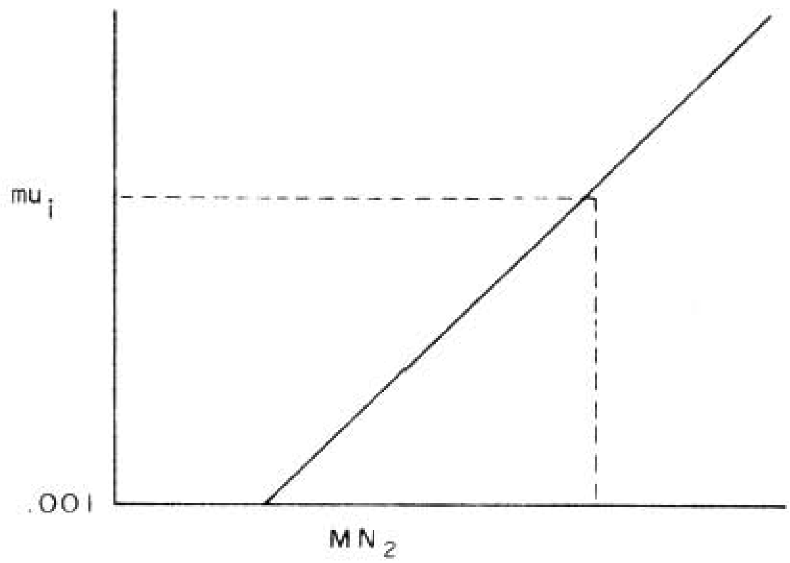


Fig. 7.3. mu_i graphed as a function of the substrate level for optimum temperature and soil moisture.

rate constant for nitrate..

There is a further addition and loss to the nitrogen system given by nitrogen fixation (nf) and denitrification (dn). Nitrogen fixation occurs at a rate proportional to the population of nitrogen fixers (Mc_5); the constant of proportionality is parameter pnf. Denitrification is proportional to the amount of soil nitrate and the population of denitrifying bacteria (Mn_4) and is characterized by a rate constant pdn. The relations for the five ISV's used in the differential equations for nitrogen PSV's are summarized below:

$$lc = 34.5 \cdot Z_2 \tag{7.15}$$

$$au = pau \cdot Mn_3 \cdot pt \tag{7.16}$$

$$nu = pnu \cdot Mn_4 \cdot pt \tag{7.17}$$

$$nf = pnf \cdot Mc_5 \tag{7.18}$$

$$dn = pdn \cdot Mn_4 \cdot Mc_4 \tag{7.19}$$

Interaction between the decomposer and the producer section is given by a "nutrient availability index," ns_i, for the *ith* plant group. This is an index of the relative soil fertility for each of the producer functional groups. The mechanism is designed to involve calculation of a weighted average amount of nitrogen available. The weights are given by pnu and pau; the resulting calculation is passed through a semi-threshold function of the arc tangent type (ISV, at, defined in producer section). The mathematics is described below:

$$ns_i = at(pns_{i1}, pns_{i2}, c_2)$$

$$c_2 = \frac{pau \cdot Mn_3 + pnu \cdot Mn_4}{pau + pnu}$$

Fig. 7.4 graphs ns_1 as a function of c_2, the weighted average nitrogen availability.

7.3 Future Expansion

The decomposer section of PWNEE, MOD 1, conforms in its general formulation to the ideas of soil scientists and microbiologists about microbially mediated processes in the soil (see, for example, Bartholemew and Clark 1965). The difficulty lies in the fact that most of the transformations have been studied in the laboratory under idealized environmental conditions in isolated cultures rather than complete soil ecosystems. The measurement of PSV's and the calculation of rates for the processes described above is largely beyond the technology of soil science at this time. Since PWNEE, MOD 1, was formulated, the microbiologists, soil scientists, and modellers of the Grassland Biome have come up with a restructured decomposer model designed to emphasize variables which can be experimentally attacked. The model is in the conceptual stages, however it is sufficiently far advanced that 1971 experimental studies are being designed around it. The mathematical version of this model will appear in PWNEE, MOD 2; its principal features are described below.

The nitrogen section emphasizes that a certain fraction of nitrogen in the soil must be of a "fast turnover" type which appears yearly as plant protein and returns to the soil via translocation to and death of roots and via rainwater leaching. The relative magnitudes of these processes and their

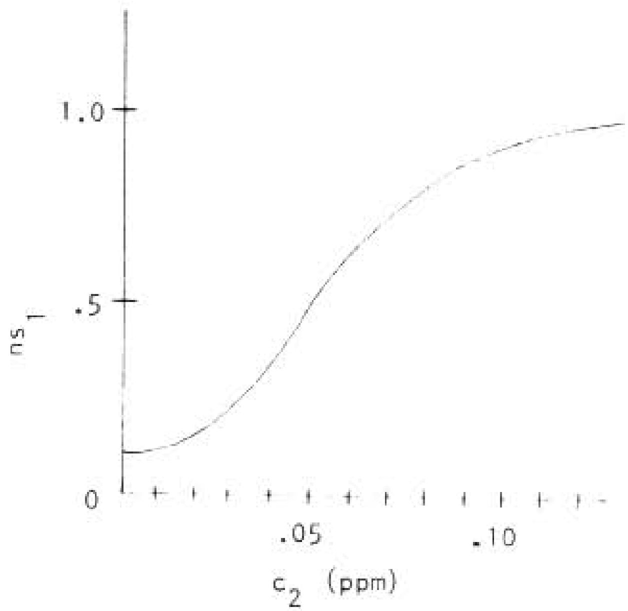


Fig. 7.4. Nutrient stress index ns_1 as a function of weighted average nitrogen availability c_2.

dynamics are under experimental study. Studies of gaseous nitrogen input and losses are also planned.

A practical, conceptual carbon cycle in the soil is one focal point for 1971 decomposer studies. Studies will emphasize the carbon input to the soil from root death and the loss via CO_2 evolution. Dynamics of these processes will be studied under various commonly occurring environmental conditions in the soil. The model will contain a fractionation of soil organic matter, as in MOD 1, with loss rates from the fractions as determined by experiment. The inputs to the various fractions will be via a modification of the mechanism explained in section 7.2. Each input will be characterized by a composition vector giving the proportion, e.g., of dead roots, in each organic matter fraction. This vector may be allowed to vary throughout the season depending upon laboratory analysis of 1971 root samples.

The decomposer model will probably be limited in MOD 2 to carbon and nitrogen, pending experimental plans to measure and analyze dynamics of other nutrients such as phosphorus.

LITERATURE CITED

Amaracho, J. and A. Broastetter. 1967. The representation of storm precipitation fields near ground level. J. Geophys. Res. 72:1145-1164.

Bartholemew, W. V. and F. E. Clark. [ed.] 1965. Soil nitrogen. Amer. Soc. Agron. Monogr. 10.

Bledsoe, L. J. and D. A. Jameson. 1969. Model structure for a grassland ecosystem, p. 410-437. *In* R. L. Dix and R. G. Beidleman [ed.] The grassland ecosystem: A preliminary synthesis. Range Sci. Dep. Sci. Ser. No. 2, Colorado State Univ., Fort Collins. 437 p.

Clymer, A. B. and L. J. Bledsoe. 1970. A guide to the mathematical modelling of an ecosystem, p. I-75-I-99. *In* R. G. Wright and G. M. Van Dyne [ed.] Simulation and analysis of dynamics of a semidesert grassland. Range Sci. Dep. Sci. Ser. No. 6, Colorado State Univ., Fort Collins. p. 50, I-214, II-71, III-10.

Gates, D. M. 1968. Toward understanding ecosystems, p. 1-35. *In* J. P. Cragg [ed.] Advances in ecological research, Vol. 5. Academic Press, New York.

Hansen, R. M. and Barbara Cavender. 1970. Assimilation rates of small mammal herbivores. Tech. Rep. No. 51. 7 p.

Kleiber, M. 1961. The fire of life. John Wiley & Sons. 454 p.

Swartzman, G. L. 1969. A preliminary bird population dynamics and biomass model. Grassland Biome Tech. Rep. No. 3.

Swartzman, G. L. 1970. Some concepts of modelling. Grassland Biome Tech. Rep. No. 32.

Whitman, W. C. 1969. Microclimate and its importance in grassland ecosystems, p. 40-64. *In* R. L. Dix and R. G. Beidleman [ed.] The grassland ecosystem: A preliminary synthesis. Range Sci. Dep. Sci. Ser. No. 2, Colorado State Univ., Fort Collins. 437 p.

Wiegert, R. G. 1965. Energy dynamics of the grasshopper populations in old field and alfalfa field ecosystems. Oikos 16:161-176.

Appendix A. List of all the equations from all the sections.

$$z_1 = \begin{cases} pz_1 \cdot \sin\left[\frac{2\pi}{24.}(\text{thm} - 6.)\right] & \text{if} \quad 6 \le \text{thm} \le 18 \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

$$z_2 = \begin{cases} (pz_{2,1,i-1} - pz_{2,2,i})\, cx[(\text{thm} + 10.) \cdot \frac{\pi}{15.}] + pz_{2,2,i} \\ \qquad \text{if} \quad 0 \le \text{thm} \le 5.0 \\ (pz_{2,1,i} - pz_{2,2,i})(1. - cx[(\text{thm} - 5.)\, \frac{\pi}{9}]) + pz_{2,2,i} \\ \qquad \text{if} \quad 5. < \text{thm} \le 14. \\ (pz_{2,1,i} - pz_{2,2,i+1})\, cx[(\text{thm} - 14.)\, \frac{\pi}{15.}] + pz_{2,2,i+1} \\ \qquad \text{if} \quad 14. < \text{thm} < 24. \end{cases} \tag{3.2}$$

$$cx(x_1) = \tfrac{1}{2}[1 + \cos(x_1)] \tag{3.3}$$

Appendix A. (Continued)

$$z_3 = \begin{cases} 1.39 \cdot 10^{-4}\ pz_{3,i} & \text{if} \quad (6. \le tmm < 9.) \text{ and } (15. \le thm < 17.) \\ 5.56 \cdot 10^{-5}\ pz_{3,i} & \text{if} \quad (tmm < 6. \text{ or } tmm \ge 9) \text{ and } (3. \le thm \le 8.) \\ 0. & \text{otherwise} \end{cases} \tag{3.4}$$

$$z_4 = \begin{cases} 447. & \text{if} \quad 12. \le thm < 18. \\ (thm - 9.) \cdot 149. & \text{if} \quad 9. \le thm < 12. \\ -(thm - 18.) \cdot 149. + 447. & \text{if} \quad 18. \le thm < 21. \\ 0 & \text{if} \quad 21. \le thm < 24. \end{cases} \tag{3.5}$$

$$Tc = \begin{cases} z_2 + 10.\left(\frac{z_1}{pz_1}\right) & \text{if} \quad (z_4 \le 223.5) \text{ and } (z_1 > .0002) \\ z_2 & \text{otherwise} \end{cases} \tag{4.1}$$

Appendix A. (Continued)

$$\dot{Ts} = pTs(Tc - Ts) \qquad (4.2)$$

$$\dot{Ms} = z_3 - pMs \cdot Ms \qquad (4.3)$$

$$\dot{Al}_i = pAl_i \cdot hp_i \cdot Al_i - (bt_i + da_i + ha_i)\ ,\ i = 1, \ldots, 4 \qquad (5.1)$$

$$\dot{Bl} = \sum_{i=1}^{4} bt_i - (db + rb \cdot Bl + hb) \qquad (5.2)$$

$$\dot{Ad} = \sum_{i=1}^{4} da_i - (sa \cdot Ad + hs) \qquad (5.3)$$

$$\dot{Lt} = sa \cdot Ad - (hl + hm + lc \cdot Lt) \qquad (5.4)$$

Appendix A. (Continued)

$$hp = \begin{cases} ms_i \cdot ns_i \cdot cr_i \cdot la_i \cdot php_{i1} \left[\dfrac{php_{i2} \cdot rp}{(1 + php_{i2} \cdot rp)} - \dfrac{php_{i2} \cdot prp_{2i}}{(1 + php_{i2} \cdot prp_{2i})} \right] \\ \quad [(pTc_{i1} - Tc)\exp(-php_{i3}[pTc_{i1} - Tc])] \\ \qquad \text{if } rp > prp_{2i} \text{ and } (0 < Tc < pTc_{i1}) \qquad [\text{case 1}] \\ -ms_i \cdot ns_i \cdot cr_i \cdot la_i \cdot php_{i4} \left[\dfrac{(prp_{2i} - rp)}{prp_{2i}} \right] \left(\dfrac{Tc}{pTc_{i2}}\right)^{php_{i5}} \\ \quad \exp\left\{ \left(\dfrac{php_{i5}}{php_{i6}}\right) \left[1 - \left(\dfrac{Tc}{pTc_{i2}}\right)^{php_{i6}}\right] \right\} \\ \qquad \text{if } rp < prp_{2i} \text{ and } Tc > 0 \qquad [\text{case 2}] \\ -ms_i \cdot ns_i \cdot cr_i \cdot php_{i7} [(Tc - pTc_{i1})\exp\{-php_{i8}(Tc - pTc_{i1})\}] \\ \qquad \text{if } rp > prp_{2i} \text{ and } Tc > pTc_{i1} \qquad [\text{case 3}] \\ 0 \qquad \text{if } Tc \le 0 \qquad [\text{case 4}] \end{cases} \tag{5.5}$$

$$rp = pz_1 \cdot \left(\frac{z_1}{pz_1}\right)^{prp_1} \tag{5.6}$$

Appendix A. (Continued)

$$ms_i = at(pms_{i2},\ pms_{i1},\ fw) \tag{5.7}$$

$$fw = \frac{\left(\frac{Ms}{20.}\right)}{1.4} \tag{5.8}$$

$$at(x_1,\ x_2,\ x_3) = \left(\frac{1}{\pi}\right) \tan^{-1}[c_1(x_3 - x_1)] + .5 \tag{5.9}$$

$$c_1 = \tan(.4 \cdot \frac{\pi}{x_2}) \tag{5.10}$$

$$rb = \begin{cases} prb_1(44 - Ts)\exp\left(\frac{-(44 - Ts)}{8}\right) & \text{if} \quad Ts \leq 44 \\ 0 \quad \text{otherwise} \end{cases} \tag{5.11}$$

$$db = \frac{rb}{2} \tag{5.12}$$

$$da_i = (1 - ms_i)pda_{i1} \cdot at(pda_{i3},\ pda_{i2},\ Tc) \cdot Al_i \tag{5.13}$$

Appendix A. (Continued)

$$bt_i = ms_i \cdot \begin{cases} 0. & \text{if } Ts \le 0 \\ \max[0., (pbt_{i1} - pbt_{i2} \cdot bl_i)Al_i] & \\ & \text{if } (Al_i > pbt_{i3}) \text{ and } (Ts > 0) \\ -\min[0., pbt_{i4}(bl_i - pbt_{i5}) & \\ & \text{if } (Al_i \le pbt_{i3}) \text{ and } (Ts > 0) \end{cases} \tag{5.14}$$

$$bl_i = Bl \cdot \frac{Al_i}{ab} \tag{5.15}$$

$$ab = \sum_{i=1}^{4} Al_i \tag{5.16}$$

$$sa = \begin{cases} .005 & \text{if } z_4 \le 223.5 \text{ cm/sec} \\ (.005 + 2.21 \cdot 10^{-4} \cdot z_4) & \text{if } z_4 > 223.5 \end{cases} \tag{5.17}$$

Appendix A. (Continued)

$$lc = .035 \cdot z_2 \tag{5.18}$$

$$C_i = N_i W_i; \quad i = 1, \ldots, 5 \tag{6.1}$$

$$\dot{N}_i = N_i[nb_i - nd_i] - \frac{ha_{i+7}}{W_i} + im_i - em_i; \quad i = 1, \ldots, 5 \tag{6.2}$$

$$\dot{W}_i = W_i \cdot as_i \cdot fc_i - re_i; \quad i = 1, \ldots, 5 \tag{6.3}$$

$$\dot{Fe} = \left[\sum_{i=1}^{5} C_i \; (1 - as_i) fc_i\right] - ha_{13} \tag{6.4}$$

$$\dot{Cd} = \left[\sum_{i=1}^{5} C_i \cdot nd_i\right] - ha_{14} \tag{6.5}$$

Appendix A. (Continued)

$$nb_1 = \begin{cases} 0 & \text{if } (tdm < 58) \text{ or } (tdm > 252) \\ .007[1 - .0046(tdm - 58)] & \text{if } 58 \leq tdm \leq 252 \end{cases} \tag{6.6}$$

$$nd_1 = .0022 \tag{6.7}$$

$$nb_2 = nd_2 = 0 \tag{6.8}$$

$$nb_3 = nd_3 = 0 \tag{6.9}$$

$$nb_4 = \begin{cases} .0308 & \text{if } 155 \leq tdm \leq 168 \\ .0102 & \text{if } 190 \leq tdm \leq 203 \\ 0 & \text{otherwise} \end{cases} \tag{6.10}$$

$$nd_4 = \begin{cases} .0021 & \text{if } tdm \leq 154 \\ .0035 & \text{if } tdm > 154 \end{cases} \tag{6.11}$$

$$im_4 = \begin{cases} .26 \cdot 10^{-8} & \text{if } 113 \leq tdm \leq 126 \\ 0 & \text{otherwise} \end{cases} \tag{6.12}$$

Appendix A. (Continued)

$$em_4 = \begin{cases} .21 \cdot 10^{-8} & \text{if } 218 \le tdm \le 235 \\ 0 & \text{otherwise} \end{cases} \quad (6.13)$$

$$nb_5 = \begin{cases} .0214[1 - .0145(tdm - 162)] & \text{if } 162 \le tdm \le 224 \\ 0 & \text{otherwise} \end{cases} \quad (6.14)$$

$$nd_5 = \begin{cases} .0071 & \text{if } 162 \le tdm \le 259 \\ 0 & \text{otherwise} \end{cases} \quad (6.15)$$

$$dl_i = \begin{cases} pfc_{i1} \left(\frac{pdl_{i1}}{W_i}\right); & W_i > pdl_{i1} \\ pfc_{i1}; & pdl_{i2} \le W_i \le pdl_{i1} \\ pfc_{i1} \left(\frac{pdl_{i2}}{W_i}\right); & W_i < pdl_{i2} \end{cases} \quad (6.16)$$

= dummy food intake variable

Appendix A. (Continued)

$$fc_i = \begin{cases} d1_i \; ; \; cb_k > pfc_{i2k} \\ d1_i \left(\dfrac{cb_k}{pb_{1k}}\right) \; ; \; pfc_{i3k} \leq cb_k \leq pfc_{i2k} \\ d1_i \left(\dfrac{pfc_{i3k}}{pfc_{i2k}}\right) \; ; \; cb_k < pfc_{i3k} \end{cases} \tag{6.17}$$

$$d2_{ik} = \begin{cases} pfp_{ik}; \; cb_k > pfc_{i2k} \\ pfp_{ik}\left(\dfrac{cb_k}{pfc_{i3k}}\right); \; pfc_{i3k} \leq cb_k \leq pfc_{i2k} \\ 0; \; cb_k < pfc_{i3k} \end{cases} \tag{6.18}$$

= dummy food preference factor

$$d2_{ij} = \begin{cases} pfp_{ij}; \; cb_j \geq pfc_{i3j} \\ 0; \; cb_j < pfc_{i3j} \end{cases} \quad j \neq k \tag{6.19}$$

Appendix A. (Continued)

$$fp_{ij} = \frac{d2_{ij}}{\sum_{j=1}^{12} d2_{ij}} \quad , \tag{6.20}$$

$$\sum_{j=1}^{12} fp_{ij} = 1.0. \tag{6.21}$$

$$ha_j = \sum_{i=1}^{5} hp_{ij} \tag{6.22}$$

$$hp_{ij} = fc_i \cdot fp_{ij} \cdot C_i \tag{6.23}$$

$$as_i = \sum_{j=1}^{12} pas_j \cdot fp_{ij}; \quad i = 1, \ldots, 5 \tag{6.24}$$

$$re_i = pre_{i1} \cdot (W_i)^{pre_{i2}}, \quad i = 1, \ldots, 5 \tag{6.25}$$

Appendix A. (Continued)

$$Mc_i = dv_i \cdot \frac{Mc_i \; (mu_i - Mc_i)}{mu_i} \tag{7.1}$$

$$So_1 = So_2 + So_3 + Mn_1 \tag{7.2}$$

$$\sum_{i=1}^{3} pSo_{1i} = 1.0 \tag{7.3}$$

$$\dot{So}_2 = pSo_{1,2}(hm + db + hf + ha) - pSo_{2,1} \cdot Mc_6 \cdot So_2 \tag{7.4}$$

$$\dot{So}_3 = pSo_{1,3}(hm + db + hf + ha) - pSo_{3,1} \cdot So_3 \tag{7.5}$$

$$\dot{Mn}_1 = pSo_{1,1}(hm + db + hf + ha) - pMn_{i,1} \cdot Mc_1 \cdot Mn_1, \tag{7.6}$$

$$\dot{Mn}_2 = pMn_{1,1} \cdot Mc_1 \cdot Mn_1 + lc \cdot Lt - pMn_{2,1} \cdot Mc_2 \cdot Mn_2, \tag{7.7}$$

Appendix A. (Continued)

$$\dot{Mn}_3 = pMn_{2,1} \cdot Mc_2 \cdot Mn_2 - pMn_{3,1} \cdot Mc_3 \cdot Mn_3 - au(Mn_3, pt), \quad (7.8)$$

$$\dot{Mn}_4 = pMn_{3,1} \cdot Mc_3 \cdot Mn_3 + nf(Mc_5) - dn(Mn_4, Mc_4) - nu(Mn_4, pt). \quad (7.9)$$

$$dv_i = pMc_{i8} \cdot op(pMc_{i1}, pMc_{i2}, Ts) \cdot op(pMc_{i3}, pMc_{i4}, Ms) \cdot sb(pMc_{i5}, pMc_{i6}, d3_i) \quad (7.10)$$

$$mu_i = pMc_{i7} \cdot op(pMc_{i1}, pMc_{i2}, Ts) \cdot op(pMc_{i3}, pMc_{i4}, Ms) \cdot sb(pMc_{i5}, pMc_{i6}, d3_i) \quad (7.11)$$

$$\left.\begin{aligned} d3_1 &= Mn_1 \\ d3_2 &= Mn_2 \\ d3_3 &= Mn_3 \\ d3_4 &= Mn_4 \\ d3_5 &= So_1 \\ d3_6 &= So_1 \end{aligned}\right\} \quad (7.12)$$

Appendix A. (Continued) .

$$op(x_1, x_2, x_3) = \frac{-2.(x_3 - x_1)(x_3 - x_2)}{(x_2 - x_1)} \tag{7.13}$$

$$sb(x_1, x_2, x_3) = \begin{cases} .001 & \text{if} \quad x_3 \leq x_2 \\ \left\{ \begin{array}{lll} 1. & \text{if} & x_1 = 0 \\ (x_3 - x_2) \cdot x_1 & \text{if} & x_1 \neq 0 \end{array} \right\} & \text{if } x_3 > x_2 \end{cases} \tag{7.14}$$

$$lc = 34.5 \cdot Z_2 \tag{7.15}$$

$$au = pau \cdot Mn_3 \cdot pt \tag{7.16}$$

$$nu = pnu \cdot Mn_4 \cdot pt \tag{7.17}$$

$$nf = pnf \cdot Mc_5 \tag{7.18}$$

$$dn = pdn \cdot Mn_4 \cdot Mc_4 \tag{7.19}$$

Appendix B. Glossary of PSV's, ISV's, and parameters. A dash in the units column indicates a variable which is a pure number.

Algebraic	FORTRAN	Description	Units
		DRIVING VARIABLES	
		PSV's	
z_1	IC	Shortwave radiation	$cal/(cm^2 \cdot sec)$
z_2	T1	Air temperature	°C
z_3	PP	Precipitation	cm/sec
z_4	WS	Wind speed	cm/sec
		ISV's	
$cx(x_1)$	COSX	Cosine function translated to vary between 0 and 1	--
		Parameters	
pz_1	SC	Peak solar radiation rate	$cal/(cm^2 \cdot sec)$
$pz_{2,1,i}$	TMX(TD)	Maximum daily air temperature, *ith* day	°C
$pz_{2,2,i}$	TMN(TD)	Minimum daily air temperature, *ith* day	°C
$pz_{3,i}$	PRECIP(I)	Amount of precipitation on *ith* day	cm
pTs	STLAG	Rate constant for soil temperature lag	$(sec)^{-1}$
		ABIOTIC VARIABLES	
		PSV's	
Tc	T2	Soil surface temperature (synonymous with canopy temperature)	°C
Ts	T3	Soil temperature for top 20 cm	°C

Appendix B. (Continued).

Algebraic	FORTRAN	Description	Unit
Ms	SM	Soil moisture, total in top 20 cm	cm
		Parameters	
pTs	STLAG	Rate constant associated with soil temperature lag	$(sec)^{-1}$
pMs	SMLAG	Rate constant associated with soil moisture loss rate	$(sec)^{-1}$
		PRODUCER VARIABLES	
		PSV's	
Al_i, i = 1, 4	VA(I)	Aboveground biomass of live plant material; i = 1-warm season grass, i = 2-cool season grass, i = 3-forbs, i = 4-*Opuntia*	g/cm^2
Bl	VB	Belowground live plant parts, total of all species	g/cm^2
Ad	VS	Biomass of standing dead plants, total	g/cm^2
Lt	VL	Biomass of aboveground plant litter, total	g/cm^2
		ISV's	
hp_i	PHOTO(I)	Net photosynthesis rate in net g of CO_2 fixed per unit plant green tissue, species i	$(sec)^{-1}$
bt_i	TB(VA,I)	Net translocation rate for *ith* species	$g/(sec \cdot cm^2)$

Appendix B. (Continued).

Algebraic	FORTRAN	Description	Unit
fw	PSM	Percent soil moisture by weight in top 20 cm of soil	--
da_i	AGD(I)	Death rate of aboveground plant parts, *ith* plant group	$g/(cm^2 \cdot sec)$
ha_i	HA(I)	Harvest rate of above-ground plant parts, *ith* plant group	$g/(cm^2 \cdot sec)$
db	BGD(RB)	Death rate of belowground plant parts	$g/(cm^2 \cdot sec)$
rb	RB(T3)	Respiration rate of below-ground plant parts per unit of plant tissue (weight)	$(sec)^{-1}$
hb	HB	Harvest rate of belowground plant parts, all harvesters	$g/(cm^2 \cdot sec)$
sa	SHATR(WS)	Shattering rate of standing dead per unit of standing dead tissue	$(sec)^{-1}$
hs	HS	Harvest rate of standing dead vegetation, macrofauna	$g/(cm^2 \cdot sec)$
hl	HL	Harvest rate of plant litter by macrofauna	$g/(cm^2 \cdot sec)$
hm	HM	Harvest rate of plant litter by microfauna	$g/(cm^2 \cdot sec)$
lc	LEACH(PP)	Leaching rate of plant litter into soil by rainwater, per unit of plant litter	$(sec)^{-1}$
ms_i	MOIS(I)	Moisture stress index for *ith* plant species (number between 0 and 1)	--
ns_i	ETE(I)	Nutrient stress index for *ith* plant species (number between 0 and 1)	--
rp	PHAR(IC)	That part of net solar radiation effective for photosynthesis, includes geometry of leaves but not leaf area index	$cal/(cm^2 \cdot sec)$

Appendix B. (Continued).

Albegraic	FORTRAN	Description	Unit
cr_i	CR(I)	Percent cover of *ith* plant species	--
la_i	L(I)	Leaf area index of *ith* plant group	--
at	ATANX	Mathematical function related to arc tangent	--
ab	TVA	Total aboveground biomass	g/cm^2
bl_i	VBI	Prorated amount of total belowground biomass in plant group i	g/cm^2
		Parameters	
pAl_i	EPSI(I)	Conversion factor for photosynthesis function--converts CO_2 fixation rate to photosynthate accumulation, *ith* plant group	--
php_{i1}	K1(I)	Peak photosynthesis rate under optimum temperature, sunlight, moisture and nutrient conditions, per unit of leaf area, *ith* plant group	$(sec)^{-1}$
php_{i2}	K3(I)	Parameter associated with sunlight response of photosynthesis curve, *ith* plant group	--
php_{i3}	K2(I)	Parameter associated with temperature response of photosynthesis curve during normal daylight and moderate temperatures, *ith* plant group	$(°C)^{-1}$
prp_{2i}	SC(I)	Threshold of positive photosynthesis response to sunlight, *ith* plant group	$cal/(cm^2 \cdot sec)$

Appendix B. (Continued).

Albegraic	FORTRAN	Description	Unit
pTc_{i1}	T20(I)	Threshold of positive photosynthesis response to temperature, *ith* plant group	°C
php_{i4}	K6(I)	Peak photosynthesis (respiration) rate under conditions of zero sunlight and optimum temperature, moisture, and nutrients, *ith* plant group	$(sec)^{-1}$
pTc_{i2}	T21(I)	Parameter associated with temperature response (position of peak) of photosynthesis curve during limiting sunlight, *ith* plant group	°C
php_{i5}	K7(I)	Parameter associated with temperature response (spread of curve for moderate temperature) of photosynthesis curve during limiting sunlight, *ith* plant group	--
php_{i6}	K8(I)	Parameter associated with temperature response (spread of curve for high temperature) of photosynthesis curve during limiting sunlight, *ith* plant group	--
php_{i7}	K4(I)	Peak photosynthesis (respiration) rate under conditions of optimum sunlight, moisture, and nutrient and high temperatures, *ith* plant group	$(sec)^{-1}$
php_{i8}	K5(I)	Parameter associated with temperature response (spread) of photosynthesis curve during normal sunlight, moisture and nutrients but high temperatures, *ith* plant group	$(°C)^{-1}$

Appendix B. (Contineed).

Algebraic	FORTRAN	Description	Unit
pms_{i1}	PMS1(I)	Parameter determining sharpness of threshold response of growth to soil moisture	--
pms_{i2}	PMS2(I)	Parameter determining location of threshold response of growth to soil moisture	--
pns_{i1}	PNS1(I)	Parameter related to spread of semi-threshold relation for nutrient stress index (ns_i) for *ith* plant group	cm^2/g
prb_1	PRB1	Parameter relating to peak respiration rate of below-ground plant parts	$(sec)^{-1}$
pda_{i1}	PDA1(I)	Parameter associated with maximum aboveground death rate	sec^{-1}
pda_{i2}	PDA2(I)	Parameter associated with response of aboveground death rate (spread of curve) to temperature	$(°C)^{-1}$
pda_{i3}	PDA3(I)	Parameter associated with response of aboveground death rate (location of inflection) to soil temperature	°C
pbt_{i1}	TBMX(I)	Parameter giving maximum translocation for high aboveground biomass	sec^{-1}
pbt_{i2}	TBSLP1(I)	Parameter associated with rate of change of trans-location rate with increasing belowground biomass for high above-ground biomass	$cm^2/(g \cdot sec)$
pbt_{i3}	VABRK(I)	Threshold for change of sign of translocation function as aboveground biomass changes	g/cm^2

Appendix B. (Continued).

Algebraic	FORTRAN	Description	Unit
pbt_{i4}	TBSLP2(I)	Parameter associated with rate of change of translocation rate with increasing belowground biomass for low aboveground biomass	$(sec)^{-1}$
pbt_{i5}	VBBRK(I)	Parameter giving value of belowground biomass below which zero translocation occurs if aboveground biomass is low	g/cm^2

CONSUMER VARIABLES

PSV's

N_i	PN(I)	Density of animals in group i	$(cm^2)^{-1}$
W_i	W(I)	Average weight of individual in group i	g
C_i	C(I)	Live biomass density of group i, i = 1,5	g/cm^2
Fe	SH	Density of animal feces	g/cm^2
Cd	AD	Density of animal dead	g/cm^2

ISV's

nb_i	B(I)	Instantaneous birth rate of *ith* animal group	$(days)^{-1}$
nd_i	D(I)	Instantaneous death rate of *ith* animal group	$(days)^{-1}$
ha_i	HA(I)	Harvest rate of ith animal group by macro-consumers	g/day
im_i	XIM(I)	Immigration rate of *ith* animal group	$(day)^{-1}$
em_i	EM(I)	Emigration rate of *ith* animal group	$(day)^{-1}$

Appendix B. (Continued).

Algebraic	FORTRAN	Description	Unit
as_i	EFF(I)	Assimilation efficiency of *ith* animal group $\frac{\text{(Ingestion-egestion)}}{\text{Ingestion}}$	--
fc_i	AF(I)	Ingestion rate, *ith* animal group, per unit body weight	$(day)^{-1}$
re_i	RE(I)	Respiration rate of *ith* animal group	g/day
ha_{13}(=hf)	HSH	Harvest rate (by microbes) of animal feces	$g/(day \cdot cm^2)$
ha_{14}(=ha)	HAD	Harvest rate (by microbes) of animal dead	$g/(day \cdot cm^2)$
cb_j	XC(J)	Biomass density of *jth* consumed species, Table 6.2 gives meaning of subscript	g/cm^2
$d1_i$	YF(I)	Dummy variables used in calculation of fc_i	--
$d2_{ij}$	YK(I,J)	Dummy variable used in calculation of fp_{ij}	--
fp_{ij}	AK(I,J)	Proportion of the diet of consumer group i made up of consumed species j under actual food availability conditions	--
fc_i	XINT(I)	Food consumption rate of animal group i, per individual	g/day
hp_{ij}	HC(I,J)	Harvest rate on total biomass basis of consumed group j by consumer group i	$g/cm^2 \cdot day)$
		Parameters	
pfc_{i1}	XF(I)	Food intake rate for *ith* animal group under ideal food or availability conditions	g/day

Appendix B. (Continued).

Algebraic	FORTRAN	Description	Unit
$pd1_{i1}$	TWU(I)	Upper threshold for consumer weights at which food consumption is equal to pfc_{i1}	g
$pd1_{i2}$	TWL(I)	Lower threshold for consumer weights at which food consumption is equal to pfc_{i1}	g
pfc_{i2j}	TT(J)	Upper threshold for density of consumed material (k) at at which food consumption will be equal to pfc_{i1} (program does not allow variation over animal group, i)	g/cm^2
pfc_{i3j}	THO(J)	Lower threshold for density of consumed material (k) at which food consumption will be a constant function of pfc_{i1} (program does not allow variation over animal group, i)	g/cm^2
pmp_{ij}	XK(I,J)	Relative amount of the diet of consumer species i made of consumed species j under ideal food availability conditions	--
pas_j	EF(J)	Efficiency of assimilation of *jth* consumed biomass group	--
pre_{i1}	K1(I)	Coefficient of respiration function	g/day
pre_{i2}	K2(I)	Exponent of respiration function	--

DECOMPOSER VARIABLES

PSV's

Mc_i	MC(I)	Index of activity of *ith* microbial type, i = 1, . . ., 6	$(cm^2)^{-1}$

Appendix B. (Continued).

Algebraic	FORTRAN	Description	Unit
Mn_i	MN(I)	Density of *ith* nitrogen form in the soil, i = 1, . . ., 4	g/cm^2
So_i	SO(I)	Density of *ith* soil constituent, i = 1, . . ., 3	g/cm^2
		ISV's	
dv_i	DIV(I)	Reproduction rate of *ith* microbial type	$(sec)^{-1}$
mu_i	MCMAX(I)	Maximum value of Mc_i	$(cm^2)^{-1}$
lc	LEACH(PP)	Rate of leaching of soluble organic nitrate from plant litter	$(sec)^{-1}$
au	AMUP	Rate of uptake of ammonia nitrogen by plants	$g/(cm^2 \cdot sec)$
nf	NFIX	Rate of fixation of gaseous nitrogen from the atmosphere	$g/(cm^2 \cdot sec)$
dn	DENIT	Rate of loss of nitrate by denitrification	$g/(cm^2 \cdot sec)$
nu	NITUP	Rate of uptake of nitrate by plants	$g/(cm^2 \cdot sec)$
op	PARAB	Mathematical function giving parabolic curve	--
sb	SBSTR	Mathematical function giving a piecewise linear curve	--
tp	TPN	Sum of positive photosynthesis rates	$(sec)^{-1}$
at	ATANX	Mathematical function giving shifted arc tangent curve	--
$d3_i$	--	Dummy variable equated to PSV forming the substrate for the *ith* microbial type	gm/cm^2

Appendix B. (Continued).

Algebraic	FORTRAN	Description	Unit
		Parameters	
$pMc_{i,1}$	MT1(I)	Minimum temperature for microbial functioning	°C
$pMc_{i,2}$	MT2(I)	Maximum temperature for microbial functioning	°C
$pMc_{i,3}$	MM1(I)	Minimum soil moisture for microbial functioning	cm
$pMc_{i,4}$	MM2(I)	Maximum soil moisture for microbial functioning	cm
$pMc_{i,5}$	SLD(I)	Rate of increase of microbial exponential growth factor (dv_i) per unit of substrate density, Mn_j, $j = 1, \ldots, 3$	$cm^2/(sec \cdot g)$
$pMc_{i,6}$	VMND(I)	Level of substrate, Mn_j, $j = 1, \ldots, 3$, below which dv_i if zero	g/cm^2
$pMc_{i,7}$	MCMAXS(I)	Absolute maximum population of *ith* microbial group	$(cm^2)^{-1}$
$pMc_{i,8}$	DIVS(I)	Absolute maximum exponential growth factor for *ith* microbial group	$(sec)^{-1}$
$pSO_{1,i}$	SOMDIS(I)	Fraction of total soil organic matter in the *ith* type, i = 1-protein, i = 2-cellulose, i = 3-miscellaneous, i = 4-not used	--
pMu_{i1}	DECRATE(I)	Parameter associated with loss rate from *ith* nitrogen form	cm^2/sec
pau	NPREF(3)	Parameter associated with ammonia uptake rate by plants	cm^2/g
pnu	NPREF(4)	Parameter associated with nitrate uptake rate by plants	cm^2/g
pnf	NFIXR	Parameter associated with nitrogen fixation rate by soil microorganisms	g/sec

Appendix B. (Continued).

Algebraic	FORTRAN	Description	Unit
pdn	DENITR	Parameter associated with denitrification rate by soil microorganisms	cm^2/g
pnx_{i1}	PNX1(I)	Parameter related to location of threshold in nutrient index (nx)	g/cm^2
pnx_{i2}	PNX2(I)	Parameter related to rate of response of nutrient index to changes in soil nitrogen densities	cm^2/g

Appendix C. Description of Program Organization and Code Listing

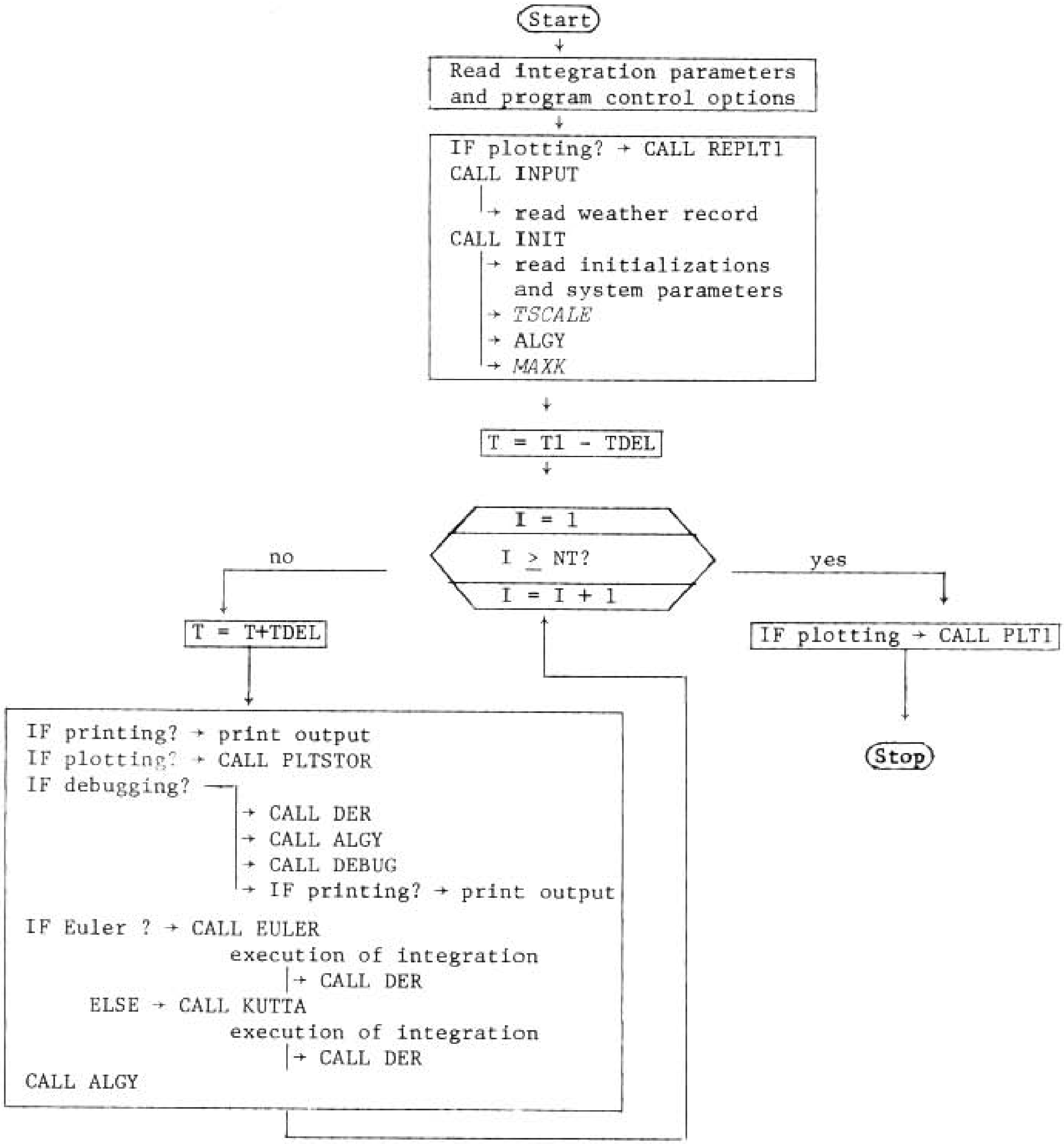
ODE is the name of a program described basically in Grassland Biome Technical Report No. 46[1/] designed to numerically integrate a set of differential equations. The user writes a FORTRAN subroutine to provide the derivatives given the state of the system. The computer implementation of PWNEE, MOD1 was constructed by modifying ODE; however, the basic structure remains as described in Technical Report No. 46. The following diagrams show the structure of ODE and the subprograms which calculate the derivatives of the PSV's. DER is the name of the main subroutine for derivative calculation; it operates by calling other subprograms for specific calculations. The program is implemented under the SCOPE 3.16 Executive system of the Control Data 6400 computer. The FORTRAN RUN compiler of that system is utilized; however, the program is basically ASA FORTRAN IV compatible.

The code contains numerous coding inefficiencies of which we are aware. In our haste to concentrate on the biological mechanisms, we have ignored these but expect to correct them in future program versions, to be recorded from scratch rather than by modification of this program. As a result of these inefficiencies and incomplete analysis of the integration step sizes required, approximately 40 minutes are required for execution over a 140 day growing season with approximately 5% maximum overall error. We expect this to reduce to about 10 minutes when inefficiencies were corrected.

[1/] Bledsoe, L. J. 1970. ODE: Numerical analysis for ordinary differential equations. U.S. IBP Grassland Biome Tech. Rep. No. 46. 42 p.

Calling sequence of subroutines (Roman) and functions (*italics*).

I. Main program and integration loop (Program ODE).



Note: Evaluation of derivatives needed for integration is implemented by the CALLs to DER and proceeding as follows.

II. Derivative evaluation.

```
DER
 ├──→TSCALE
 ├──→ALGY
 │     └──→ABIO1
 │            ├──→TMX
 │            ├──→TMN
 │            └──→PPT
 ├──→ISV
 │     ├──→DIET
 │     └──→HARVC
 ├──→ABIO2
 │     └──→PPT
 ├──→PROD
 │     ├──→PHOTO
 │     │      ├──→PHAR
 │     │      ├──→MOIS
 │     │      │     └──→PSM
 │     │      └──→ETE
 │     ├──→TB
 │     │    └──→MOIS
 │     │          └──→PSM
 │     ├──→AGD
 │     │    └──→MOIS
 │     │          └──→PSM
 │     ├──→BGD
 │     ├──→SHATR
 │     ├──→LEACH
 │     └──→RB
 ├──→CONS
 │     ├──→EFFIC
 │     ├──→RESP
 │     ├──→BIRTH
 │     ├──→DEATH
 │     └──→MIGR
 └──→DECOM
       ├──→MICR
       │     ├──→DIV
       │     │     └──→SBSTR
       │     └──→MCMAX
       │           └──→SBSTR
       ├──→SOIL
       └──→NITR
             ├──→NFIX
             ├──→DENIT
             ├──→NITUP
             └──→AMUP
```

III. Description of subprograms. All subprograms whose function is not described by Technical Report No. 46 are briefly identified below.

Subprogram Name	Purpose
INPUT	Reads parameters associated with driving functions from system file WXDATA.
INIT	Reads parameters of model from file INPUT.
TSCALE	Calculates the independent variable in various time units from the basic time variable in seconds.
ALGY	Calculates all PSV's defined by algebraic, as opposed to differential, equations.
MAXK	Finds index of maximum number in array.
DER	Basic subroutine to convert PSV's from formal parameters to COMMON block L1 and calls derivative finding routines.
ABIO1	Finds values for driving functions by calling other routines.
TMX	Looks up daily maximum temperature in parameter array.
TMN	Looks up daily minimum temperature in parameter array.
PPT	Looks up daily precipitation in parameter array.
ISV	Calculates values of certain ISV's needed in more than one model section (e.g., harvest rates).
DIET	Calculates food intake rates for consumers.
HARVC	Calculates harvest rates.
ABIO2	Main routine for finding derivatives of abiotic PSV's.
PROD	Main routine for finding derivatives of producer PSV's.
PHOTO	Finds photosynthesis rate.
PHAR	Finds photosynthetically active radiation level.
MOIS	Finds moisture stress index.

Subprogram Name	Purpose
PSM	Finds fraction by weight of soil moisture from amount by volume in centimeters.
ETE	Finds nutrient stress index.
TB	Finds plant translocation rate.
AGD	Finds death rate of aboveground live biomass.
BGD	Finds death rate of belowground live biomass.
SHATR	Finds rate of transfer of standing dead plants to litter.
LEACH	Finds rate of leaching of organic material from plant litter.
RB	Finds respiration rate of belowground live plant biomass.
CONS	Main routine for finding derivatives of consumer PSV's.
EFFIC	Finds assimilation rates for consumers.
RESP	Finds respiration rates for consumers.
BIRTH	Finds birth rates for consumers.
DEATH	Finds death rates for consumers.
MIGR	Finds emigration and immigration rates for consumers.
DECOM	Main routine for calculated derivatives of decomposer variables.
MICR	Finds derivatives of microbial density PSV's.
DIV	Finds exponential reproduction coefficients for microbes.
MCMAX	Finds maximum population density for microbes.
SOIL	Finds derivatives of soil characteristic variables.
NITR	Finds derivatives of nitrogen concentration variables.
NFIX	Finds nitrogen fixation rate.
DENIT	Finds denitrification rate.
NITUP	Finds nitrate ion uptake rate by plants.
AMUP	Finds ammonia ion uptake rate by plants.

Subprogram Name	Purpose

The following subprograms are used purely for their mathematical, as opposed to biological, properties.

SUM	Adds elements of an array.
ATANX	Gives a double asymptote function based on the arc tangent curve.
PARAB	Gives an inverted (negative second derivative) symmetric paraboloid function normalized to a peak value of one.
SBSTR	Gives a piecewise linear function.
COSX	Gives cosine function normalized to vary between sero and one.

IV. Program listing.

```
      PROGRAM ODE                                                        ODE
     1(INPUT,OUTPUT,WXDATA,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1=WXDATA,       ODE
     1PWNDRG,TAPE8=PWNDRG)                                               ODE
      DIMENSION V(50),VP(50),ID(8),FC(8)                                 ODE
      DIMENSION NCMP(4),XX1(100),XX2(100),XX3(100),XX4(100),YY1(100),   ODE
     -     YY2(100),YY3(100),YY4(100)                                    ODE
      COMMON/L5/ADRG(4)                                                  ODE
      EXTERNAL DER                                                       ODE
      READ (5,1300) ID                                                   ODE
    1 READ(5,1300) FC                                                    ODE
      DECODE (5,100,FC) N1                                               ODE
      IF (N1   ) 2,11,4                                                  ODE
    4 DECODE (80,100,FC) NCOM,NP,NT,T1,TDEL,ACC,DEL,ITER                 ODE
      READ(5,1300)DB,DESOLV,PLOT,CHK,WRIT                                ODE
    2 WRITE (6,200) ID,D,TT,T1,TDEL,NT                                   ODE
      IF(PLOT.EQ.10RPLOT       ) CALL RDPLT1(NNN,MMM,NCMP,TSTEP,NX1,NX2,  ODE
     -      NX3,NX4,XXL,XXU,YYL,YYU)                                     ODE
      WRITE (6,210) NCOM,NP,ACC,DEL,ITER                                 ODE
      READ (5,300) (V(I),I=1,NCOM)                                       ODE
  300 FORMAT(7F10.6)                                                     ODE
      WRITE (6,3002) DB,DESOLV,PLOT,CHK,WRIT                             ODE
 3002 FORMAT(* SYSTEM OPTIONS -- *5R10//)                                ODE
      IF (N1   .LT.0) GO TO 3                                            ODE
      NT=ISECDAY(NT) $ T1=SECDAY(T1) $ TDEL=SECDAY(TDEL)                 ODE
      CALL INPUT (NP)                                                    ODE
    3 T=T1-TDEL                                                          ODE
      CALL INIT(T,V)                                                     ODE
      IF(CHK.EQ.10RCHKOUT     )CALL CHKOUT                               ODE
      WRITE(6,2000)                                                      ODE
 2000 FORMAT(1H1)                                                        ODE
      WRITE (6,600)                                                      ODE
      DO 20 I=1,NT                                                       ODE
      T=T+TDEL $ T2=T+TDEL                                               ODE
      T3=TSCALE(T)                                                       ODE
      IF (WRIT.NE.10RNOWRIT     ) WRITE (6,800) T3,(V(J),J=1,NCOM),ADRG  ODE
      I4=T3*100.                                                         ODE
      T3=FLOAT(I4)/100.                                                  ODE
      IF(PLOT.EQ.10RPLOT       ) CALL PLTSTOR(NNN,NCMP,T3,NCOM,V,TSTEP,  ODE
     -      XX1,XX2,XX3,XX4,YY1,YY2,YY3,YY4,NX1,NX2,NX3,NX4,XXL,XXU,     ODE
     -      YYL,YYU,ADRG)                                                ODE
      IF (DB.NE.10HDEBUG      ) GO TO 8                                  ODE
      CALL DER(T,V,VP) $ CALL ALGY(T,V) $ CALL DEBUG(T,V,VP)             ODE
      IF (WRIT.NE.10RNOWRIT     ) WRITE (6,900) (VP(J),J=1,NCOM)         ODE
    8 IF (DESOLV.EQ.10HEULER     ) 5,6                                   ODE
    5 CALL EULER(T,T2,V,NCOM,DEL) $ GO TO 7                              ODE
    6 CALL KUTTA(T,T2,V,NCOM,DEL,ACC,ITER,DER)                           ODE
    7 CALL ALGY(T2,V)                                                    ODE
      IF(WRIT.NE.10RNOWRIT     ) WRITE(6,1100)                           ODE
   20 CONTINUE                                                           ODE
      ENDFILE 8                                                          ODE
      IF(PLOT.NE.10RPLOT       ) GO TO 1                                 ODE
      WRITE(6,3000) (NCMP(I),I=1,NNN)                                    ODE
 3000 FORMAT(1H1,*PLOT OF VARIABLES NO.:*,4I5)                           ODE
      CALL PLT1(IERR,TEMP,NNN,MMM,XXL,XXU,YYL,YYU,XX1,NX1,YY1,XX2,NX2,   ODE
     -      YY2,XX3,NX3,YY3,XX4,NX4,YY4)                                 ODE
      IF(IERR.NE.0) WRITE(6,3001) IERR                                   ODE
 3001 FORMAT(1H0,I5,* POINTS WERE NOT PLOTTED *)                         ODE
```

```
      GO TO 1
   11 STOP                                                                 ODE
  100 FORMAT(3I5,4F10.0,I5)                                                ODE
  200 FORMAT(*1DE SOLUTION*/* *8A10/A10*  TIME*A10//                       ODE
     1* TIME STARTS AT *F10.3*, INCREMENTS BY *F15.5* FOR *I5* STEPS*/)    ODE
  210 FORMAT(* *I5* EQUATIONS, *I5* CONSTANTS*/* KUTTA PARAMETERS ARE *    ODE
     12F12.5,I5)                                                           ODE
  600 FORMAT(//* TIME, VARIABLE VALUES*)                                   ODE
  800 FORMAT(//* *G10.3,10G12.5/(11X,10G12.5))                             ODE
  900 FORMAT(*0DERIVATIVE*10G12.5/(11X,10G12.5))                           ODE
 1100 FORMAT(/)                                                            ODE
 1300 FORMAT(8A10)                                                         ODE
      END                                                                  ODE
```

```
      SUBROUTINE EULER (XL,XU,V,N,DEL)                                     ODE
      DIMENSION V(10),VP(50)                                               ODE
      DT=(XU-XL)*DEL                                                       ODE
      T=XL-DT                                                              ODE
      ND=1./DEL+.0001                                                      ODE
      DO 20 K=1,ND                                                         ODE
      T=T+DT                                                               ODE
      CALL DER (T,V,VP)                                                    ODE
      DO 20 I=1,N                                                          ODE
   20 V(I)=V(I)+DT*VP(I)                                                   ODE
      RETURN                                                               ODE
      END                                                                  ODE
```

```
      SUBROUTINE KUTTA(XL,XU,Y,NE,DEL,ACCURC,IMAX,EQUA)                 ODE
C PROGRAM AUTHOR F.D.HAMMERLING, CENTRAL DATA PROCESSING, ORGDP.        ODE
      DIMENSION Y(1),YI(50),YN(50),K1(50),K2(50),K3(50),                ODE
     >     K4(50),K5(50),F(50),E(50),F1(50)                             ODE
      REAL K1,K2,K3,K4,K5                                               ODE
      LOGICAL QUIT,HALVE                                                ODE
      DATA HALVE/.T./                                                   ODE
      ITTER=0                                                           ODE
      N=NE                                                              ODE
      XN=XL                                                             ODE
      H=DEL                                                             ODE
      QUIT=.FALSE.                                                      ODE
      DO 1 I=1,N                                                        ODE
    1 YN(I)=Y(I)                                                        ODE
    2 IF(XN+H.LT.XU)GO TO 3                                             ODE
      DEL=H                                                             ODE
      H=XU-XN                                                           ODE
      QUIT=.TRUE.                                                       ODE
    3 CALL EQUA(XN,YN,F1)                                               ODE
    4 DO 5 I=1,N                                                        ODE
      K1(I)=H*F1(I)/3.                                                  ODE
    5 YI(I)=YN(I)+K1(I)                                                 ODE
      CALL EQUA(XN+H/3.,YI,F)                                           ODE
      DO 6 I=1,N                                                        ODE
      K2(I)=H*F(I)/3.                                                   ODE
    6 YI(I)=YN(I)+K1(I)/2.+K2(I)/2.                                     ODE
      CALL EQUA(XN+H/3.,YI,F)                                           ODE
      DO 7 I=1,N                                                        ODE
      K3(I)=H*F(I)/3.                                                   ODE
    7 YI(I)=YN(I)+3.*K1(I)/8.+9.*K3(I)/8.                               ODE
      CALL EQUA(XN+H/2.,YI,F)                                           ODE
      DO 8 I=1,N                                                        ODE
      K4(I)=H*F(I)/3.                                                   ODE
    8 YI(I)=YN(I)+3.*K1(I)/2.-9.*K3(I)/2.+6.*K4(I)                      ODE
      CALL EQUA(XN+H,YI,F)                                              ODE
      DO 9 I=1,N                                                        ODE
      K5(I)=H*F(I)/3.                                                   ODE
      E(I)=(K1(I)-9.*K3(I)/2.+4.*K4(I)-K5(I)/2.)/5.                     ODE
    9 CONTINUE                                                          ODE
      DO 11 I=1,N                                                       ODE
   11 YN(I)=YN(I)+(K1(I)+4.*K4(I)+K5(I))/2.                             ODE
      TEST=0.                                                           ODE
      DO 20 I=1,N                                                       ODE
      IF(YN(I).EQ.0.) GO TO 21                                          ODE
      ERR=E(I)/YN(I)                                                    ODE
      GO TO 20                                                          ODE
   21 ERR=0.                                                            ODE
   20 TEST=AMAX1(TEST,ABS(ERR))                                         ODE
      IF(TEST.LT.ACCURC)GO TO 10                                        ODE
      ITTER=ITTER+1                                                     ODE
      IF(ITTER.GE.IMAX) HALVE=.F.                                       ODE
      IF(.N.HALVE) GO TO 10                                             ODE
      IF(HALVE) H=H/2                                                   ODE
      QUIT=.FALSE.                                                      ODE
      GO TO 4                                                           ODE
   10 XN=XN+H                                                           ODE
      IF(TEST.LT.ACCURC/32.)H=2.*H                                      ODE
```

```
      IF(.NOT.QUIT)GO TO 2                                              ODE
      DO 12 I=1,N                                                       ODE
   12 Y(I)=YN(I)                                                        ODE
      GO TO 14                                                          ODE
   14 RETURN                                                            ODE
      END                                                               ODE
```

```
      SUBROUTINE DEBUG(T,V,VP)                                          ODE
      DIMENSION V(50),VP(50),D(45)                                      ODE
      COMMON/L3/TIME(8)/L4/H(18)/L5/DRV(4)/CONS/CO(288)                 ODE
      REAL MOIS,TC,LEACH,MCMAX                                          ODE
      PP=DRV(2) $ TC=DRV(3) $ SM=V(5) $ VA=V(6) $ T3=V(4) $ WS=DRV(4)   ODE
      D(1)=PHAR(TC) $ D(2)=PSM(SM)                                      ODE
      DO 10 I=1,4                                                       ODE
      J=I+2 $ K=I+7 $ L=I+13 $ LJ=I+17 $ D(K)=TR(VA,I) $ D(L)=AGD(I)    ODE
      D(LJ)=MOIS(I)                                                     ODE
   10 D(J)=PHOTO(I)                                                     ODE
      D(7)=RR(T3) $ D(12)=ETF(1) $ D(13)=BGD(D(7)) $ D(22)=SHATR(WS)    ODE
      D(23)=LEACH(PP)                                                   ODE
      DO 20 I=1,6                                                       ODE
      J=I+23 $ K1=I+29 $ K2=I+35                                        ODE
      D(K1)=DIV(I) $ D(K2)=MCMAX(I)                                     ODE
   20 D(J)=MAXK(I)                                                      ODE
      D(42)=AMUP(0.) $ D(43)=NETX(0.) $ D(44)=DENIT(V(36))              ODE
      D(45)=NITUP(0.)                                                   ODE
      WRITE(8)T,TIME,V,VP,DRV,D,H,CO                                    ODE
      RETURN                                                            ODE
      END                                                               ODE
```

```
      FUNCTION ISECDAY(ID)                                              ODE
      ISECDAY=ID                                                        ODE
      RETURN                                                            ODE
      END                                                               ODE
```

```
      FUNCTION SECDAY(D)                                                ODE
      SECDAY=D*86400.                                                   ODE
      RETURN                                                            ODE
      END                                                               ODE
```

```
      SUBROUTINE INPUT(NP)                                               PWNEF
C     READS IN VALUES FOR MAXIMUM AND MINIMUM DAILY TEMPERATURES AND    PWNEF
C     DAILY PRECIPITATION                                                PWNEF
      COMMON/ABIO2/TMAX(730),TMIN(730),PRECIP(730)                       PWNEF
      READ(1,100)(TMIN(I),TMAX(I),PRECIP(I),I=1,NP)                      PWNEF
      DO 10 I=1,730                                                      PWNEF
      TMIN(I)=(TMIN(I)-32.)*.5555555556                                  PWNEF
      TMAX(I)=(TMAX(I)-32.)*.5555555556                                  PWNEF
   10 PRECIP(I)=2.54*PRECIP(I)                                           PWNEF
      WRITE (6,300)                                                      PWNEF
      DO 20 I=1,730,20                                                   PWNEF
   20 WRITE(6,200) I,TMIN(I),TMAX(I),PRECIP(I)                           PWNEF
100   FORMAT(15X,3F6.0)                                                  PWNEF
  200 FORMAT(* *I5,10G12.4)                                              PWNEF
  300 FORMAT(* DRIVING FUNCTION*/* TMIN,TMAX,PPT*//)                     PWNEF
      RETURN                                                             PWNEF
      END                                                                PWNEF
```

```
      SUBROUTINE INIT(DUM1,V)                                            PWNFE
C     INITIALIZES VALUES FOR SYSTEM VARIABLES AND SUPPLIES VALUES FOR    PWNFE
C     SYSTEM PARAMETERS                                                  PWNFE
      COMMON/ABIO1/A(4)                                                  PWNFE
      COMMON/PROD1/B(60)                                                 PWNFE
      COMMON/CONS/XF(5),XK(5,12),THO(12),TT(12),EF(12),MJ(5),TWU(5),TWL( PWNFE
     15),AF(5),AK(5,12),HC(5,12),XC(12),Q(5),RE(5),D(5),XIM(5),EM(5),EFF PWNFE
     2(5),XINT(5)                                                        PWNFE
      COMMON/L3/T,TD,TDMOD,TH,THMOD,TMMOD,TWMOD,TY                       PWNFE
      COMMON/L5/T1,PP,TC,WS                                              PWNFE
      COMMON/HARV1/HAR(4)                                                PWNFE
      COMMON/PROD2/TRDATA(20)                                            PWNFE
      COMMON/PROD3/RRDATA(3),DATMOIS(8)                                  PWNFE
      COMMON/CONS2/RE1(5),RE2(5)                                         PWNFE
      COMMON/DECOM1/DIVS(6),MCMAXS(6),MT1(6),MT2(6),MM1(6),MM2(6),       PWNFE
     1SLD(6),SLMX(6),VMND(6),VMNMX(6),SBMN(6),SOMDIS(4),DECRATE(6),      PWNFE
     2SOMDEC,NFIXR,DENITR,NPREF(4)                                       PWNFE
      REAL MCMAXS,MT1,MT2,MM1,MM2,NPREF,NFIXR                            PWNFE
      REAL TC                                                            PWNFE
C     READS IN CONTENTS OF ABIO1                                         PWNFE
      READ(5,100)A                                                       PWNFE
100   FORMAT(7F10.4)                                                     PWNFE
      WRITE(6,200)A                                                      PWNFE
200   FORMAT(*1SYSTEM PARAMETERS*//* ABIOTIC*/* COMMON ABIO1*/* SC,PHEXP PWNFE
     1,STLAG,SMLAG*/(* *10G12.5))                                        PWNFE
C     READS IN CONTENTS OF PROD1                                         PWNFE
      READ(5,100)B                                                       PWNFE
      WRITE(6,300)B                                                      PWNFE
  300 FORMAT(*0PRODUCER*/* COMMON PROD1*/* T20(4),T21(4),CR(4),K1(4),K2( PWNFE
     14),K3(4),K4(4),K5(4),K6(4),K7(4),K8(4),ETF(4),L(4),SC(4),EPSI(4)*  PWNFE
     2/(* *10G12.5))                                                     PWNFE
C     COMMON HARV1, HARVEST FUNCTIONAL COEFFICIENTS                      PWNFE
      READ(5,100) HAR                                                    PWNFE
      WRITE(6,600) HAR                                                   PWNFE
  600 FORMAT(/* COMMON HARV1*/* TOLIT1,TOLIT2,TMNLT,TMXLT*/              PWNFE
     1(1H ,10F13.5))                                                     PWNFE
C     COMMON BLOCK PROD2, DATA FOR TRANSLOCATION FUNCTION                PWNFE
      READ(5,100) TRDATA                                                 PWNFE
      WRITE(6,500) TRDATA                                                PWNFE
  500 FORMAT(*0COMMON PROD2*/* VABPK, VBABK, TRSLP1, TRMX, TRSLP2*/      PWNFE
     1(1H ,4F13.5))                                                      PWNFE
      READ(5,100) RRDATA                                                 PWNFE
      WRITE(6,501) RRDATA                                                PWNFE
  501 FORMAT(*0COMMON PROD3*/* RR1, RR2, RR3*/1H ,3F13.5)                PWNFE
      READ(5,1001) DATMOIS                                               PWNFE
      WRITE(6,502) DATMOIS                                               PWNFE
  502 FORMAT(* PMO1(4), PMO2(4)*/(1H ,4F13.5))                           PWNFE
      READ(5,100) RE1                                                    PWNFE
      READ(5,100) RE2                                                    PWNFE
      WRITE(6,503) RE1,RE2                                               PWNFE
  503 FORMAT(*0COMMON CONS2*/* RE1(5), RE2(5)*/(1H ,5F13.5))             PWNFE
C     READS IN COMMON CONS1                                              PWNFE
      READ(5,2)(XF(I),I=1,5)                                             PWNFE
2     FORMAT(5F5.0)                                                      PWNFE
      READ(5,3)((XK(I,J),J=1,12),I=1,5)                                  PWNFE
3     FORMAT(12F5.0)                                                     PWNFE
      READ(5,700)(EF(I),I=1,12)                                          PWNFE
```

```
700    FORMAT(12F5.0)                                                          PWNEE
       READ(5,4)(THO(I),TT(I),I=1,12)                                          PWNEE
4      FORMAT(2F10.0)                                                          PWNEE
       READ(5,50)(TWL(I),TWU(I),I=1,5)                                         PWNEE
50     FORMAT(2F10.0)                                                          PWNEE
       WRITE (6,102)                                                           PWNEE
  102  FORMAT(//* CONSUMERS*/* COMMON CONS1*)                                  PWNEE
       WRITE(6,101)(XF(I),I=1,5)                                               PWNEE
  101  FORMAT(10H XF         ,5G12.5)                                          PWNEE
       WRITE(6,14)((XK(I,J),J=1,12),I=1,4)                                     PWNEE
   14  FORMAT(10H XK         ,7G12.5/1H+,9X,5G12.5)                            PWNEE
       WRITE(6,150)(TT(J),J=1,12)                                              PWNEE
  150  FORMAT(10H TT         ,7G12.5/1H+,9X,5G12.5)                            PWNEE
       WRITE(6,151)(THO(J),J=1,12)                                             PWNEE
  151  FORMAT(10H THO        ,7G12.5/1H+,9X,5G12.5)                            PWNEE
       WRITE (6,152)(TWL(I),TWU(I),I=1,5)                                      PWNEE
  152  FORMAT(      10H TWL, TWU 10G12.5/(11X10G12.5))                         PWNEE
C      CALCULATE MOST PREFERRED FOOD SPECIES                                   PWNEE
       DO 10 I=1,5                                                             PWNEE
   10  MJ(I)=MAXK(I)                                                           PWNEE
       WRITE (6,400) MJ                                                        PWNEE
  400  FORMAT(        * MJ *10I5)                                              PWNEE
       READ(5,1000) DIVS,MCMAXS,MT1,MT2,MM1,MM2,SLD,SLMX,VMND,VMNMX,SBMN,      PWNEE
     -       DECRATE                                                           PWNEE
 1000  FORMAT(6F10.4)                                                          PWNEE
       READ(5,1001) SOMDIS,NPREF                                               PWNEE
 1001  FORMAT(4F10.4)                                                          PWNEE
       READ(5,1001) SOMDEC,NFIXR,DENITR                                        PWNEE
       WRITE(6,2000) DIVS,MCMAXS,MT1,MT2,MM1,MM2,SLD,SLMX,VMND,VMNMX,          PWNEE
     -       SBMN,DECRATE                                                      PWNEE
 2000  FORMAT(*0COMMON DECOM1*/* DIVS, MCMAXS, MT1, MT2, MM1, MM2, SLD, S      PWNEE
     -LMX, VMND, VMNMX, SBMN, DECRATE*/(1H ,6E13.5))                           PWNEE
       WRITE(6,2001) SOMDIS,NPREF                                              PWNEE
 2001  FORMAT(* SOMDIS, NPREF*/(1H ,4E13.5))                                   PWNEE
       WRITE(6,2002) SOMDEC,NFIXR,DENITR                                       PWNEE
 2002  FORMAT(* SOMDEC, NFIXR, DENITR*/1H ,3(E13.5,5X))                        PWNEE
C      INITIALIZES TIME VALUES AND ALGEBRAIC EQUATIONS                         PWNEE
       DI=TSCALE(0.)                                                           PWNEE
       CALL ALGY(T,V)                                                          PWNEE
       RETURN                                                                  PWNEE
       END                                                                     PWNEE
```

```
      SUBROUTINE DER(T,V,VP)                                            PWNEE
      CALLS ALL SUBROUTINES IN THE SYSTEM WHICH GIVE PRINCIPAL SYSTEM   PWNEE
      VARIABLES IN DERIVATIVE (DIFF. EQUATION) FORM                     PWNEE
      DIMENSION V(1),VP(1)                                              PWNEE
      COMMON/L1/T2,T3,SM,VA(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),    PWNEE
     1MC(6),SO(4)                                                       PWNEE
      COMMON/L2/DT2,DT3,DSM,DVA(4),DVB,DVS,DVL,DC(6),DPN(6),DW(6),DAD,  PWNEE
     1DSH,DMN(4),DMC(6),DSO(4)                                          PWNEE
      REAL MN,MC                                                        PWNEE
      T2=V(3) $ T3=V(4) $ SM=V(5) $ VB=V(10) $ VS=V(11) $ VL=V(12)      PWNEE
      AD=V(31) $ SH=V(32)                                               PWNEE
      DO 10 I=1,4                                                       PWNEE
      K1=I+5 $ K2=I+32 $ K3=I+42                                        PWNEE
      VA(I)=V(K1) $ MN(I)=V(K2)                                         PWNEE
   10 SO(I)=V(K3)                                                       PWNEE
      DO 20 I=1,6                                                       PWNEE
      K1=I+12 $ K2=I+18 $ K3=I+24 $ K4=I+36                             PWNEE
      C(I)=V(K1) $ PN(I)=V(K2) $ W(I)=V(K3)                             PWNEE
   20 MC(I)=V(K4)                                                       PWNEE
      D1=TSCALE(T)                                                      PWNEE
      CALL ALGY(T,V)                                                    PWNEE
      CALL TSV                                                          PWNEE
      CALL ABIO2                                                        PWNEE
      CALL PROD                                                         PWNEE
      CALL CONS                                                         PWNEE
      CALL DECOM                                                        PWNEE
      DPN(6)=DW(6)=0.                                                   PWNEE
      DO 60 I=1,6                                                       PWNEE
   60 DC(I)=0.                                                          PWNEE
      DO 70 I=1,14                                                      PWNEE
   70 DPN(I)=1.157407407E-5*DPN(I)                                      PWNEE
                                                                        PWNEE
      VP(3)=DT2 $ VP(4)=DT3 $ VP(5)=DSM $ VP(10)=DVB $ VP(11)=DVS       PWNEE
      VP(12)=DVL $ VP(31)=DAD $ VP(32)=DSH                              PWNEE
      DO 30 I=1,4                                                       PWNEE
      K1=I+5 $ K2=I+32 $ K3=I+42                                        PWNEE
      VP(K1)=DVA(I) $ VP(K2)=DMN(I)                                     PWNEE
   30 VP(K3)=DSO(I)                                                     PWNEE
      DO 40 I=1,6                                                       PWNEE
      K1=I+12 $ K2=I+18 $ K3=I+24 $ K4=I+36                             PWNEE
      VP(K1)=DC(I) $ VP(K2)=DPN(I) $ VP(K3)=DW(I)                       PWNEE
   40 VP(K4)=DMC(I)                                                     PWNEE
      VP(1)=VP(2)=0.                                                    PWNEE
      RETURN                                                            PWNEE
      END                                                               PWNEE
```

```
      SUBROUTINE ALGY(D1,V)                                                    PWNEE
C     DEFINES THOSE PRINCIPAL SYSTEM VARIABLES GIVEN BY ALGERAIC               PWNEE
C     EQUATIONS... TDMOD (TIME IN DAYS) TMMOD (TIME IN MONTHS) AND T2          PWNEE
C     (CANOPY TEMPERATURE)                                                     PWNEE
      COMMON/L1/T2,T3,SM,VA(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),           PWNEE
     1MC(6),SO(4)                                                              PWNEE
      COMMON/L3/Z,TD,TDMOD,TH,THMOD,TMMOD,TWMOD,TY                             PWNEE
      COMMON/L5/T1,PP,IC,WS                                                    PWNEE
      COMMON/ABIO1/SC,PHEXP,STLAG,SMLAG                                        PWNEE
      REAL IC                                                                  PWNEE
      REAL MN,MC                                                               PWNEE
      DIMENSION V(10)                                                          PWNEE
      V(1)=TDMOD $ V(2)=TMMOD                                                  PWNEE
      CALL ABIO1                                                               PWNEE
      V(3)=T1                                                                  PWNEE
      IF( WS.LE.223.5.AND. IC.GT..0002 ) V(3)=T1+10.*IC/SC                     PWNEE
      T2=V(3)                                                                  PWNEE
C     CALCULATE TOTAL CONSUMER BIOMASS VALUES                                  PWNEE
      DO 10 I=13,18                                                            PWNEE
       J=I+6 $ K=I+12                                                          PWNEE
   10 V(I)=V(J)*V(K)                                                           PWNEE
      RETURN                                                                   PWNEE
      END                                                                      PWNEE
```

```
      SUBROUTINE ISV                                                           PWNEE
      COMMON/L1/T2,T3,SM,VA(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),           PWNEE
     1MC(6),SO(4)                                                              PWNEE
      COMMON/L4/HA(12),HAD,HSH,HB,HS,HL,HM                                     PWNEE
      COMMON/DECOM2/TPN                                                        PWNEE
      COMMON/HARV1/TOLIT1,TOLIT2,TMNLT,TMXLT                                   PWNEE
      COMMON/CONS/XF(5),XK(5,12),THO(12),TT(12),FF(12),MJ(5),TWU(5),TWL(       PWNEE
     15),AF(5),AK(5,12),HC(5,12),XC(12),B(5),RE(5),D(5),XIM(5),EM(5),EFF       PWNEE
     2(5),XINT(5)                                                              PWNEE
      TRANSFER PSV,S TO ARRAY OF CONSUMED SPECIES (XC)                         PWNEE
      DO 2 I=1,4                                                               PWNEE
    2 XC( I)=VA(I)                                                             PWNEE
      XC(5)=VB $ XC(6)=VS $ XC(7)=VL                                           PWNEE
      DO 3 I=1,5                                                               PWNEE
      II=I+7                                                                   PWNEE
    3 XC(II)=C(I)                                                              PWNEE
      CALCULATE ISV FOR HARVESTS                                               PWNEE
      CALL DIET                                                                PWNEE
      CALL HARVC                                                               PWNEE
      CALCULATE TOTAL PHOTOSYNTHESIS RATE                                      PWNEE
      TPN=0.                                                                   PWNEE
      DO 6 I=1,4                                                               PWNEE
    6 TPN=TPN+PHOTO(I)                                                         PWNEE
      CONVERT HARVESTS, G/CM2/DAY TO G/CM2/SEC                                 PWNEE
      DO 4 I=1,12                                                              PWNEE
    4 HA(I)=HA(I)/86400.                                                       PWNEE
      DO 5 I=13,18                                                             PWNEE
    5 HA(I)=0.                                                                 PWNEE
      HM=TOLIT1*VL*PARAB(TMNLT,TMXLT,T2)                                       PWNEE
      HL=TOLIT2*VL                                                             PWNEE
      RETURN                                                                   PWNEE
      END                                                                      PWNEE
```

```
LOGICAL FUNCTION MID(A,B,C)                              PWNEE
MID= .FALSE.                                             PWNEE
IF((A.LE.C).AND.(B.GE.C)) MID=.TRUE.                     PWNEE
RETURN                                                   PWNEE
END                                                      PWNEE
```

```
FUNCTION COSX(Y)                                         PWNEE
COSX=(COS(Y)+1.)*.5                                      PWNEE
RETURN                                                   PWNEE
END                                                      PWNEE
```

```
FUNCTION ATANX(X1,X2,X3)                                 PWNEE
GIVES SHIFTED ARC TANGENT RESPONSE TO ARGUMENT X3        PWNEE
X1 GIVES LOCATION OF INFLECTION POINT                    PWNEE
X2 GIVES SPREAD WIDTH, .5 -.9                            PWNEE
3.077... = TAN(.4*PI), .318... = 1./PI                   PWNEE
C1=3.077683537*(1./X2)                                   PWNEE
ATANX=.3183098861*ATAN(C1*(X3-X1))+.5                    PWNEE
RETURN                                                   PWNEE
END                                                      PWNEE
```

```
      FUNCTION TSCALE(X)                                               PWNEE
      COMMON/L3/T,TD,TDMOD,TH,THMOD,TMMOD,TWMOD,TY                     PWNEE
      DIMENSION XMO(13)                                                PWNEE
      HANDLES SYSTEM TIME SCALING                                      PWNEE
      DATA (XMO(I),I=1,13)/0.,31.,59.,90.,120.,151.,181.,212.,243.,    PWNEE
     1273.,304.,334.,365./                                             PWNEE
      Y=X $ TD=Y*.1157407407E-4 $ T=X                                  PWNEE
      TDMOD=AMOD(TD,365.)$TH=Y*.2777777778E-3                          PWNEE
      THMOD=AMOD(TH,24.)                                               PWNEE
      DO 10 I=1,12                                                     PWNEE
      IF (TDMOD.LT.XMO(I)) GO TO 1                                     PWNEE
   10 CONTINUE                                                         PWNEE
      M=12 $ I=13 $ GO TO 2                                            PWNEE
    1 M=I-1                                                            PWNEE
    2 N=(XMO(I)-XMO(I-1))*1.0001                                       PWNEE
      S1=(T-XMO(M)*86400.)                                             PWNEE
      TMMOD=(M-1.)+S1/(N*86400.)                                       PWNEE
      TMMOD=AMOD(TMMOD,12.)                                            PWNEE
      TWMOD=TDMOD*.1428571429                                          PWNEE
      TY=TD*.2739726027E-02                                            PWNEE
      TSCALE=TD                                                        PWNEE
      RETURN                                                           PWNEE
      END                                                              PWNEE
```

```
      SUBROUTINE ABIO1                                                  ABIO
C     DEFINES THE PRINCIPAL DRIVING FORCES FOR THE SYSTEM...IC (SOLAR   ABIO
C     RADIATION), WS(WIND SPEED), T1(AIR TEMPERATURE), AND PP(DAILY     ABIO
C     PRECIPITATION).                                                   ABIO
      COMMON/L3/T,TD,TDMOD,TH,THMOD,TMMOD,TWMOD,TY                      ABIO
      COMMON/L5/T1,PP,IC,WS                                             ABIO
      COMMON/ABIO1/SC,PHEXP,STLAG,SMLAG                                 ABIO
      REAL IC                                                           ABIO
      LOGICAL MID                                                       ABIO
      DATA C1/.2094395100/                                              ABIO
      DATA C2/.3490658500/                                              ABIO
      DATA TUPI/6.28318530/                                             ABIO
      IC=0.                                                             ABIO
C     SOLAR RADIATION ASSUMED TO VARY SINUSOIDALLY THROUGHOUT THE DAY   ABIO
      IF(MID(6.,18.,THMOD)) IC=SC*SIN((THMOD-6.)*(TUPI/24.))            ABIO
C     WIND SPEED ASSUMED ZERO BETWEEN 9 P.M. AND 9 A.M. IT IS CONSTANT  ABIO
C     AT 44F CM/SEC BETWEEN NOON AND 6 P.M. AND VARIES LINEARLY BETWEEN ABIO
C     ZERO AND 44F THE REST OF THE DAY.                                 ABIO
      WS=0.                                                             ABIO
      IF (THMOD.GE.9.) WS=(THMOD-9.)*149.                               ABIO
      IF(MID(12.,18.,THMOD))WS=447.                                     ABIO
      IF (MID(18.,21.,THMOD))WS=447.-(THMOD-18.)*149.                   ABIO
      IF (THMOD.GE.21.) WS=0.                                           ABIO
C     TEMPERATURE IS A COSINE FUNCTION VARYING  THROUGHTOUT THE DAY     ABIO
C     BETWEEN THE MAXIMUM AND MINIMUM TEMPERATURES. THE MAXIMUM TEMP IS ABIO
C     ASSUMED TO BE AT 2 P.M. AND THE MINIMUM AT 5 A.M.                 ABIO
      IF ((TD.GT.1.).OR.(THMOD.GT.5.)) GO TO 96                         ABIO
      T1=(TMX(TD)-TMN(TD))*COSX((THMOD+10.)*C1)+TMN(TD) $ GO TO 97      ABIO
   96 IF (THMOD.LE.5.) T1=(TMX(TD-1.)-TMN(TD))*                         ABIO
     1COSX((THMOD+10.)*C1)+TMN(TD)                                      ABIO
      IF(MID(5.,14.,THMOD)) T1=(TMX(TD)-TMN(TD))*                       ABIO
     1(1.-COSX((THMOD-5.)*C2))+TMN(TD)                                  ABIO
      IF(THMOD.GE.14.) T1=(TMX(TD)-TMN(TD+1.))*                         ABIO
     1COSX((THMOD-14.)*C1)+TMN(TD+1.)                                   ABIO
   97 PP=PPT(0.)                                                        ABIO
      RETURN                                                            ABIO
      END                                                               ABIO
```

```
      FUNCTION PPT(D1)                                                        ABIO
C     PRECIPITATION, IF IT OCCURS AT ALL, IS ASSUMED TO OCCUR BETWEEN 3      ABIO
C     AND 5 P.M. JULY - SEPETEMBER AND BETWEEN 3 AND 8 A.M. ALL OTHER        ABIO
C     MONTHS                                                                  ABIO
      COMMON/L3/T,TD,TDMOD,TH,THMOD,TMMOD,TWMOD,TY                            ABIO
      COMMON/ABIO2/TMAX(730),TMIN(730),PRECIP(730)                            ABIO
      LOGICAL MID                                                             ABIO
      KD=TDMOD*1.00001+1.                                                     ABIO
      PPT=0.                                                                  ABIO
      IF (MID(6.,9.,TMMOD)) GO TO 1                                           ABIO
      IF (MID(3.,8.,THMOD)) PPT=PRECIP(KD)*5.555555555E-5                     ABIO
      RETURN                                                                  ABIO
    1 IF (MID(15.,17.,THMOD)) PPT=PRECIP(KD)*1.388888889E-4                   ABIO
      RETURN                                                                  ABIO
      END                                                                     ABIO
```

```
      FUNCTION TMX(TDMOD)                                                     ABIO
C     MAXIMUM DAILY TEMPERATURE FUNCTION                                      ABIO
      COMMON/ABIO2/TMAX(730),TMIN(730),PRECIP(730)                            ABIO
      KD=1.00001*(TDMOD+1.)                                                   ABIO
      TMX=TMAX(KD)                                                            ABIO
      RETURN                                                                  ABIO
      END                                                                     ABIO
```

```
      FUNCTION TMN(TDMOD)                                                     ABIO
C     MINIMUM DAILY TEMPERATURE FUNCTION                                      ABIO
      COMMON/ABIO2/TMAX(730),TMIN(730),PRECIP(730)                            ABIO
      KD=1.00001*(TDMOD+1.)                                                   ABIO
      TMN=TMIN(KD)                                                            ABIO
      RETURN                                                                  ABIO
      END                                                                     ABIO
```

```
      FUNCTION PHAR(IC)                                                       ABIO
C     COMPUTES PHOTOSYNTHETICALLY ACTIVE RADIATION FROM INCOMING SOLAR       ABIO
C     RADIATION. PHEXP IS AN INPUT PARAMETER.                                 ABIO
      COMMON/ABIO1/SC,PHEXP,STLAG,SMLAG                                       ABIO
      REAL IC                                                                 ABIO
      PHAR=0.                                                                 ABIO
      IF(IC.EQ.0.) RETURN                                                     ABIO
      PHAR=EXP(PHEXP*ALOG(IC/SC))                                             ABIO
      PHAR=IC*PHAR                                                            ABIO
      RETURN                                                                  ABIO
      END                                                                     ABIO
```

```
      SUBROUTINE ABIO2                                                    ABIO
C     GIVES DIFFERENTIAC EQUATION FOR PRIMARY ABIOTIC SYSTEM VARIABLES    ABIO
      COMMON/L1/T2,T3,SM,VA(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),      ABIO
     1MC(6),SO(4)                                                         ABIO
      COMMON/L2/DT2,DT3,DSM,DVA(4),DVB,DVS,DVL,DC(6),DPN(6),DW(6),DAD,    ABIO
     1DSH,DMN(4),DMC(6),DSO(4)                                            ABIO
      COMMON/ABIO1/D1,PHEXP,STLAG,SMLAG                                   ABIO
      COMMON/L3/T,TD,TDMOD,TH,THMOD,TMMOD,TWMOD,TY                        ABIO
      DT2=0.                                                              ABIO
      DT3=STLAG*(T2-T3)                                                   ABIO
      DSM=PPT(0.)-SM*SMLAG                                                ABIO
      IF (T3.LE.0.) DSM=0.                                                ABIO
      RETURN                                                              ABIO
      END                                                                 ABIO
```

```
      FUNCTION PSM(SM)                                                    ABIO
C     CONVERTS MOISTURE IN UPPER 20 CM OF SOIL TO PER CENTAGE             ABIO
C     ASSUMES BULK DENSITY OF 1.4                                         ABIO
      PSM=(SM/20.)*1.4                                                    ABIO
      RETURN                                                              ABIO
      END                                                                 ABIO
```

```
      SUBROUTINE PROD                                                   PROD
C     GIVES DIFFERENTIAC EQUATIONS FOR PRIMARY SYSTEM VARIABLES IN THE  PROD
C     PRODUCER COMPARTMENT                                              PROD
      COMMON/L1/T2,T3,SM,VA(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),    PROD
     1MC(6),SO(4)                                                       PROD
      COMMON/L2/DT2,DT3,DSM,DVA(4),DVB,DVS,DVL,DC(6),DPN(6),DW(6),DAD,  PROD
     1DSH,DMN(4),DMC(6),DSO(4)                                          PROD
      COMMON/L3/T,TD,TDMOD,TH,THMOD,TMMOD,TWMOD,TY                      PROD
      COMMON/L4/HA(12),HAD,HSH,HB,HS,HL,HM                              PROD
      COMMON/PROD1/T20(4),T21(4),CP(4),K1(4),K2(4),K3(4),K4(4),K5(4),   PROD
     1K6(4),K7(4),K8(4),ETE(4),L(4),SC(4),EPSI(4)                       PROD
      COMMON/L5/T1,PP,IC,WS                                             PROD
      REAL IC                                                           PROD
      REAL K1,K2,K3,K4,K5,K6,K7,K8,L                                    PROD
C SUPPLIES RATES OF CHANGE IN PRODUCER COMPARTMENT FOR INPUT TO OTHER CO PROD
      REAL LEACH                                                        PROD
      TAGD=TTB=0.                                                       PROD
      DO 12 I=1,4                                                       PROD
C COMPUTE RATE OF CHANGE OF ABOVE GROUND BIOMASS BY SPECIES             PROD
      DVA(I)=EPSI(I)*PHOTO(I)-(TB(VA,I)    +AGD(I)+HA(I))               PROD
      TTB=TTB+TB(VA,I)                                                  PROD
12    TAGD=TAGD+AGD(I)                                                  PROD
C COMPUTES TATE OF CHANGE OF BELOW GROUND BIOMASS                       PROD
      D1=RB(T3)                                                         PROD
      DVB=TTB-(BGD(D1)+D1*VB+HB)                                        PROD
C COMPUTES RATE OF CHANGE OF PLANT STANDING DEAD                        PROD
      DVS=TAGD-(SHATR(WS)*VS+HS)                                        PROD
C COMPUTES RATE OF CHANGE OF PLANT LITTER                               PROD
      DVL=SHATR(WS)*VS-(HL+HM+LEACH(PP)*VL)                             PROD
      RETURN                                                            PROD
      END                                                               PROD
```

```
      FUNCTION PHOTO(I)                                                  PROD
C     COMPUTES NET PHOTOSYNTHESIS AS A FUNCTION OF CANOPY TEMPERATURE,   PROD
C     PHOTOSYNTHETICALLY ACTIVE RADIATION, SOIL MOISTURE, AND SOIL       PROD
C     NUTRIENT STATUS.                                                   PROD
      COMMON/L1/T2,T3,SM,VA(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),     PROD
     1MC(6),SO(4)                                                        PROD
      COMMON/PROD1/T20(4),T21(4),CR(4),K1(4),K2(4),K3(4),K4(4),K5(4),    PROD
     1K6(4),K7(4),K8(4),ETE(4),L(4),SC(4),EPSI(4)                        PROD
      COMMON/L5/T1,PP,IC,WS                                              PROD
      REAL IC                                                            PROD
      REAL K1,K2,K3,K4,K5,K6,K7,K8,L                                     PROD
      REAL MOIS                                                          PROD
      S=PHAR(IC)                                                         PROD
      PHOTO=0.                                                           PROD
      IF(T2.LT.0.) RETURN                                                PROD
      IF(T2.LT.T20(I).AND.S.GT.SC(I)) GO TO 10                           PROD
      IF(T2.GE.T20(I).AND.S.GT.SC(I)) GO TO 11                           PROD
C NET PHOTOSYN, ABOVE COM TEMP. AND BELOW COMP. NET RADIATION            PROD
      PHOTO=-CR(I)*VA(I)*K6(I)*((SC(I)-S)/SC(I))*((T2/T21(I))            PROD
     1**K7(I))*EXP((K7(I)/K8(I))*(1.-(T2/T21(I))**K8(I)))*MOIS(I)*ETE(I) PROD
      PHOTO=PHOTO*L(I)                                                   PROD
      RETURN                                                             PROD
C  FORMULA FOR NET PHOTOSYNTHESIS BELOW COMPENSATION TEMPERATURE         PROD
   10 D2=S/(1.+K3(I)*S)                                                  PROD
      D3=SC(I)/(1.+K3(I)*SC(I))                                          PROD
      D4=T20(I)-T2                                                       PROD
      D5=EXP(-K2(I)*(T20(I)-T2))                                         PROD
      PHOTO    =CR(I)*VA(I)*L(I)*MOIS(I)*ETE(I)*(D2-D3)*D4*D5            PROD
      PHOTO=PHOTO*K1(I)                                                  PROD
      RETURN                                                             PROD
C NET PHOTOSYNTHESIS ABOVE COMPENSATION TEMP. AND COMP. NET RADIATION    PROD
11    PHOTO    =-CR(I)*VA(I)*K4(I)* (T2-T20(I))*EXP(-K5(I)*(T2-T20(I)))* PROD
     1MOIS(I)*ETE(I)                                                     PROD
      RETURN                                                             PROD
      END                                                                PROD
```

```
      FUNCTION RB(T3)                                                   PROD
C COMPUTES RESPIRATION OF BELOW GROUND PARTS OF PLANTS                   PROD
C     RESP IS A PIECEWISE LINEAR FUNCTION OF SOIL TEMP, MAX AT 36 DEG C PROD
C     RESP IS ZERO IF T3 GT 44 OR LT 0                                  PROD
      COMMON/PROD3/RB1,RB2,RB3,PMO1(4),PMO2(4)                          PROD
      RB=0.                                                             PROD
      IF(T3.GE.RB2) RETURN                                              PROD
      TRB=RB2-T3                                                        PROD
      RB=RB1*TRB*EXP(-TRB/RB3)                                          PROD
      RETURN                                                            PROD
      END                                                               PROD
```

```
      FUNCTION TB(XX,I)                                                 PROD
C FUNCTION TO DETERMINE TRANSLOCATION BELLOW GROUND                     PROD
C TRANSLOCATION DEPENDS ON BELOW GROUND BIOMASS SUBJECT TO THRESHOLED LI PROD
C  IONS BASED ON ABO VE GROUND BIOMASS                                  PROD
      COMMON/L1/T2,T3,SM,V(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),     PROD
     1MC(6),SO(4)                                                       PROD
      COMMON/PROD2/VABRK(4),VBBRK(4),TBSLP1(4),TBMX(4),TBSLP2(4)        PROD
      REAL MOIS                                                         PROD
      TVA=V(1)+V(2)+V(3)+V(4)                                           PROD
      VBI=(V(I)/TVA)*VB                                                 PROD
      TB=0.                                                             PROD
      IF (T3.LE.0.) RETURN                                              PROD
      IF (V(I).LT.VABRK(I)) GO TO 1                                     PROD
      TB=(TBMX(I)-TBSLP1(I)*VBI)*V(I)*MOIS(I)                           PROD
      IF (TB.LT.0.)TB=0.                                                PROD
      RETURN                                                            PROD
    1 TB=(-TBSLP2(I)*(VBI-VBBRK(I)))*MOIS(I)                            PROD
      IF (TB.GT.0.) TB=0.                                               PROD
      RETURN                                                            PROD
      END                                                               PROD
```

```
      FUNCTION FTE(I)                                                   PROD
C       NUTRIENT AVAILABILITY INDEX FOR PLANT SPECIES I                 PROD
      COMMON/L1/T2,T3,SM,V(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),     PROD
     1MC(6),SO(4)                                                       PROD
      COMMON/DECOM1/DIVS(6),MCMAXS(6),MT1(6),MT2(6),MM1(6),MM2(6),      PROD
     1SLD(6),SLMX(6),VMND(6),VMNMX(6),SBMN(6),SOMDIS(4),DECRATE(6),     PROD
     2SOMDEC,NFIXR,DENITR,NPREF(4)                                      PROD
      C1=(NPREF(3)*MN(3)+NPREF(4)*MN(4))/(NPREF(3)+NPREF(4))            PROD
      FTE=ATANX(.286E-5,.143E-5,C1)                                     PROD
      RETURN                                                            PROD
      END                                                               PROD
```

```
      FUNCTION BGD(R)                                                    PROD
C FOR A FIRST PASS DEATH RATE OF BELOW GROUND PARTS WILL BE HALF THE RES PROD
      COMMON/L1/T2,T3,SM,VA(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),     PROD
     1MC(6),SO(4)                                                        PROD
      BGD=(R*VB)/2.                                                      PROD
      RETURN                                                             PROD
      END                                                                PROD
```

```
      FUNCTION AGD(I)                                                    PROD
C  DEATH RATE OF ABOVE GROUND PARTS BY SPECIES                           PROD
C WILTING BEGINS AT 35 DEG. CENT. FOR ALL SPECIES IN FIRST PASS MODLE    PROD
      COMMON/L1/T2,T3,SM,VA(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),     PROD
     1MC(6),SO(4)                                                        PROD
      REAL MOIS                                                          PROD
      AGD=VA(I)*(1.-MOIS(I))*ATANX(35.,42.1600484521,T2)*.43E-8          PROD
      RETURN                                                             PROD
      END                                                                PROD
```

```
      REAL FUNCTION MOIS(I)                                              PROD
C     COMPUTES PERCENTAGE OF SOIL MOISTURE CAPACITY IN TOP 20 CM OF SOIL PROD
C     ASSUME MOISTURE STRESS IS AN ARCTANGENT FUNCTION OF THE PERCENTAGE PROD
C     OF SOIL MOISTURE CAPACITY IN THE TOP 20 CM OF THE SOIL             PROD
      COMMON/L1/T2,T3,SM,VA(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),     PROD
     1MC(6),SO(4)                                                        PROD
      COMMON/PROD3/RB1,RB2,RB3,PMO1(4),PMO2(4)                           PROD
      PMO1(I)=3.077683537/PMO1(I)                                        PROD
      MOIS=ATANX(PMO1(I),PMO2(I),PSM(SM))                                PROD
      RETURN                                                             PROD
      END                                                                PROD
```

```
      FUNCTION SHATR(WS)                                                 PROD
C     COMPUTES SHATTER RATE OF STANDING DEAD AS A FUNCTON OF WIND SPEED  PROD
      IF(WS.LE.223.5)SHATR=.005                                          PROD
      IF(WS.GT.223.5)SHATR=.005+(2.2148E-04)*WS                          PROD
      SHATR=SHATR*1.1574074074E-5                                        PROD
      RETURN                                                             PROD
      END                                                                PROD
```

```
      REAL FUNCTION LEACH(PP)                                            PROD
C COMPUTES DIRECT LEACHING OF LITTER AS A FUNCTION OF RAINFALL           PROD
C ASSUME LEACHING IS A LINEAR FUNCTIN OF RAINFALL                        PROD
      LEACH=.03448*PP                                                    PROD
      RETURN                                                             PROD
      END                                                                PROD
```

```
      SUBROUTINE CONS                                                   CONS
      COMMON/L1/T2,T3,SM,VA(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),    CONS
     1MC(6),SO(4)                                                       CONS
      COMMON/L2/DT2,DT3,DSM,DVA(4),DVB,DVS,DVL,DC(6),DPN(6),DW(6),DAD,  CONS
     1DSH,DMN(4),DMC(6),DSO(4)                                          CONS
      COMMON/L3/T,TD,TDMOD,TH,THMOD,TMMOD,TWMOD,TY                      CONS
      COMMON/L4/HA(12),HAD,HSH,HB,HS,HL,HM                              CONS
      COMMON/CONS/XF(5),XK(5,12),THO(12),TT(12),EF(12),MJ(5),TWU(5),TWL( CONS
     15),AF(5),AK(5,12),HC(5,12),XC(12),B(5),RE(5),D(5),XIM(5),EM(5),EFF CONS
     2(5),XINT(5)                                                       CONS
C                                                                       CONS
C     CONSUMED SPECIES 1 = BUGR                                         CONS
C     CONSUMED SPECIES 2 = AGSM                                         CONS
C     CONSUMED SPECIES 3 = CA                                           CONS
C     CONSUMED SPECIES 4 = OPPO                                         CONS
C     CONSUMED SPECIES 5 = BELOW GROUND PARTS                           CONS
C     CONSUMED SPECIES 6 = STANDING DEAD                                CONS
C     CONSUMED SPECIES 7 = LITTER                                       CONS
C     CONSUMED SPECIES 8 = CONSUMER SPECIES 1 = JACK RABBITT            CONS
C     CONSUMED SPECIES 9 = CONSUMER SPECIES 2 = COW                     CONS
C     CONSUMED SPECIES 10= CONSUMER SPECIES 2 = COYOTE                  CONS
C     CONSUMED SPECIES 11= CONSUMER SPECIES 3 = LARK BUNTING            CONS
C     CONSUMED SPECIES 12= CONSUMER SPECIES 5 = GRASSHOPPER             CONS
C     XF(I)=FOOD CONSUMPTION RATE FOR CONSUMER SPECIES I UNDER IDEAL    CONS
C        FOOD AVAILABILITY CONDITIONS (G/G/DAY)                         CONS
C     AF(I)=ACTUAL FOOD CONSUMPTION RATE FOR CONSUMER SPECIES I         CONS
C     XK(I,J)=AMOUNT OF DIET OF CONSUMER SPECIES I MADE UP OF CONSUMED  CONS
C     SPECIES J UNDER IDEAL FOOD AVAILABILITY CONDITIONS                CONS
C     AK(I,J)=ACTUAL AMOUNT OF DIET OF CONSUMER SPECIES I MADE UP OF    CONS
C        CONSUMED SPECIES J                                             CONS
C     THO(J)=THRESHOLD DENSITY BELOW WHICH CONSUMED SPECIES J CEASES    CONS
C        TO BE CONSUMED                                                 CONS
C     MJ(I)=MOST PREFERED FOOD SOURCE FOR CONSUMER SPECIES I            CONS
C                                                                       CONS
      CALL EFFIC                                                        CONS
      CALL RESP                                                         CONS
      CALL BIRTH                                                        CONS
      CALL DEATH                                                        CONS
      CALL MIGR                                                         CONS
      DO300K=1,5                                                        CONS
300   XINT(K)=W(K)*EFF(K)*AF(K)                                         CONS
      DO 100I=1,5                                                       CONS
      II=I+7                                                            CONS
C                                                                       CONS
C     HA(J),J=1,12 , IS COMPUTED IN SUBROUTINE HARVC                    CONS
C     AF(I),I=1,5 , IS COMPUTED IN SUBROUTINE DIET                      CONS
C     EFF(I) IS COMPUTED IN FUNCTION SUBROUTINE EFF(I)                  CONS
C     RE(I),I=1,5, IS COMPUTED IN SUBROUTINE RESP                       CONS
C                                                                       CONS
      DW(I)=W(I)*EFF(I)*AF(I)-RE(I)                                     CONS
101   DPN(I)=PN(I)*(B(I)-D(I))-HA(II)/W(I )+XIM(I)-EM(I)                CONS
100   CONTINUE                                                          CONS
C                                                                       CONS
C     MUST GET DEAD ANIMAL BIMASS (DAD) AND ANIMAL FECES FOR ALL        CONS
C        CONSUMER SPECIES                                               CONS
C                                                                       CONS
      DSH=0.                                                            CONS
```

```
      DAD=0.                                                            CONS
      DO200I=1,5                                                        CONS
      II=I+7                                                            CONS
      DSH=DSH+XC(II)*(1.-FFF(I))*AF(I)                                  CONS
200   DAD=DAD+XC(II)*D(I)                                               CONS
      DSH=DSH-HSH                                                       CONS
      DAD=DAD-HAD                                                       CONS
      RETURN                                                            CONS
      END                                                               CONS
```

```
      SUBROUTINE DIET                                                    CONS
C                                                                        CONS
C     SUBROUTINE DIET COMPUTES ACTUAL FOOD CONSUMPTION RATE AND ACTUAL   CONS
C        DIETARY COMPOSITION FOR CONSUMER SPECIES I,I=1,5                CONS
C                                                                        CONS
      COMMON/L1/T2,T3,SM,VA(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),     CONS
     1MC(6),SO(4)                                                        CONS
      COMMON/CONS/XF(5),XK(5,12),THO(12),TT(12),EF(12),MJ(5),TWU(5),TWL( CONS
     15),AF(5),AK(5,12),HC(5,12),XC(12),B(5),RE(5),D(5),XIM(5),EM(5),EFF CONS
     2(5),XINT(5)                                                        CONS
      DIMENSION YK(12)                                                   CONS
      DO2I=1,5                                                           CONS
C                                                                        CONS
C     DO NOT WANT TO ERASE XK FROM STORAGE SO DEFINE DUMMY VARIABLE YK   CONS
C        TO WORK WITH IN THE SUBROUTINE                                  CONS
C                                                                        CONS
      DO 1  J=1,12                                                       CONS
1     YK(J)  =  XK(I,J)                                                  CONS
C                                                                        CONS
C     ACTUAL FOOD CONSUMPTION RATE INITIALLY SET EQUAL TO IDEAL FOOD     CONS
C        CONSUMPTION RATE                                                CONS
C                                                                        CONS
      AF(I)=XF(I)                                                        CONS
      IF(W(I).LE.0.)GO TO 2                                              CONS
C                                                                        CONS
C     IF AVE. WT. OF CONS. SPECIES I GETS ABOVE A THRESHOLD, TWU(I),     CONS
C        THEN F(I) DECREASES.                                            CONS
C     IF AVE. WT. OF CONS. SPECIES I GETS BELOW A THRESHOLD, TWL(I),     CONS
C        THEN F(I) INCREASED.                                            CONS
C                                                                        CONS
      IF(W(I).LT.TWL(I))AF(I)=TWL(I)*AF(I)/W(I)                          CONS
      IF(W(I).GT.TWU(I))AF(I)=TWU(I)*AF(I)/W(I)                          CONS
C                                                                        CONS
C     ELIMINATE ANY CONSUMED SPECIES FROM THE DIET OF CONSUMER SPECIES   CONS
C        I IF ITS BIOMASS IS BELOW THRESHOLD THO                         CONS
C                                                                        CONS
      DO 20  J=1,12                                                      CONS
      IF(XC(J).GE.THO(J))GO TO 20                                        CONS
      YK(J)  =  0.                                                       CONS
20    CONTINUE                                                           CONS
C                                                                        CONS
C     MOST PREFERED FOOD SOURCE FOR CONSUMER SPECIES I                   CONS
C                                                                        CONS
      NJ=MJ(I)                                                           CONS
C                                                                        CONS
C     IF THE ABUNDANCE OF THE MOST PREFERED SPECIES IS LESS THAN TT,     CONS
C        THEN THE TOTAL AMOUNT OF FOOD CONSUMPTION FOR SPECIES I         CONS
C        DECREASES LINEARLY UNTIL IT REACHES THO/TT OF WHAT IT WAS       CONS
C        UNDER IDEAL CONDITIONS. IF, IN ADDITITION, THE ABUNDANCE OF THE CONS
C        MOST PREFERED SPECIES IS ABOVE THO, THEN THE AMOUNT OF THAT     CONS
C        CONSUMED SPECIES IN THE DIET OF CONSUMER SPECIES I WILL         CONS
C        DECREASE LINEARLY                                               CONS
C                                                                        CONS
      IF(XC(NJ).GE.TT(NJ)) GO TO 40                                      CONS
      AF(I)=AMAX1(XC(NJ),THO(NJ))*AF(I)/TT(NJ)                           CONS
      IF(XC(NJ).LT.THO(NJ)) GO TO 40                                     CONS
      YK(NJ)    =  XC(NJ) * XK(I,NJ)/TT(NJ)                              CONS
```

```
40    CONTINUE                                                          CONS
C                                                                       CONS
C     THE ACTUAL FOOD PREFERENCE FACTORS FOR CONSUMER SPECIES I ARE     CONS
C        NOW COMPUTED                                                   CONS
C                                                                       CONS
      SY=SUM(YK,12)                                                     CONS
      IF(SY.LE.0.)GO TO 3                                               CONS
      DO50 J=1,12                                                       CONS
50    AK(I,J)=YK(J)/SY                                                  CONS
      GO TO 2                                                           CONS
3     DO51 J=1,12                                                       CONS
51    AK(I,J)=0.                                                        CONS
2     CONTINUE                                                          CONS
      RETURN                                                            CONS
      END                                                               CONS
```

```
      SUBROUTINE HARVC                                                  CONS
C                                                                       CONS
C     THIS SUBROUTINE COMPUTES THE RATE OF CONSUMPTION (G/DAY) OF       CONS
C        CONSUMED SPECIES J BY CONSUMER SPECIES I, AND THE TOTAL HARVEST CONS
C        RATE (G/DAY) OF CONSUMED SPECIES J, J=1,12                     CONS
C                                                                       CONS
C     HC(I,J)=RATE OF CONSUMPTION (G/DAY) OF CONSUMED SPECIES J BY      CONS
C        CONSUMER SPECIES I                                             CONS
C     HA(J)=TOTAL HARVEST RATE (G/DAY) OF CONSUMED SPECIES J            CONS
C                                                                       CONS
      COMMON/L1/T2,T3,SM,VA(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),    CONS
     1MC(6),SO(4)                                                       CONS
      COMMON/L4/HA(12),HAD,HSH,HB,HS,HL,HM                              CONS
      COMMON/CONS/XF(5),XK(5,12),THO(12),TT(12),EF(12),MJ(5),TWU(5),TWL( CONS
     15),AF(5),AK(5,12),HC(5,12),XC(12),B(5),RE(5),D(5),XIM(5),EM(5),EFF CONS
     2(5),XINT(5)                                                       CONS
      DIMENSION HD(12)                                                  CONS
      DO1 J=1,12                                                        CONS
      DO2 I=1,5                                                         CONS
      II=I+7                                                            CONS
      HC(I,J)=AK(I,J)*AF(I)*XC(II)                                      CONS
2     HD(I)=HC(I,J)                                                     CONS
1     HA(J)=SUM(HD,5)                                                   CONS
      RETURN                                                            CONS
      END                                                               CONS
```

```
      SUBROUTINE EFFIC                                                   CONS
C                                                                        CONS
C     THIS SUBROUTINE COMPUTES THE ASSIMILATION EFFICIENCY FOR THE FOOD  CONS
C        CONSUMED BY THE CONSUMER SPECIES                                CONS
C                                                                        CONS
C     EF(J)=ASSIMILATION EFFICIENCY OF CONSUMED SPECIES J, J=1,12        CONS
C                                                                        CONS
      COMMON/CONS/XF(5),XK(5,12),THO(12),TT(12),EF(12),MJ(5),TWU(5),TWL( CONS
     15),AF(5),AK(5,12),HC(5,12),XC(12),B(5),RE(5),D(5),XIM(5),EM(5),EFF CONS
     2(5),XINT(5)                                                        CONS
      DIMENSION XF(12)                                                   CONS
      DO10I=1,5                                                          CONS
      DO1J=1,12                                                          CONS
1     XF(J)=EF(J)*AK(I,J)                                                CONS
      EFF(I)=SUM(XF,12)                                                  CONS
10    CONTINUE                                                           CONS
      RETURN                                                             CONS
      END                                                                CONS



      FUNCTION MAXK(I)                                                   CONS
C                                                                        CONS
C     FOR ANY CONSUMER SPECIES I, THIS SUBROUTINE COMPUTES THE MOST      CONS
C        PREFERED CONSUMED SPECIES J                                     CONS
C                                                                        CONS
      COMMON/CONS/XF(5),XK(5,12),THO(12),TT(12),EF(12),MJ(5),TWU(5),TWL( CONS
     15),AF(5),AK(5,12),HC(5,12),XC(12),B(5),RE(5),D(5),XIM(5),EM(5),EFF CONS
     2(5),XINT(5)                                                        CONS
      XMX=XK(I,1)                                                        CONS
      MAXK=1                                                             CONS
      DO10 J=2,12                                                        CONS
      YMX=AMAX1(XMX,XK(I,J))                                             CONS
      IF(YMX.LE.XMX)GO TO 10                                             CONS
      MAXK=J                                                             CONS
      XMX=YMX                                                            CONS
10    CONTINUE                                                           CONS
      RETURN                                                             CONS
      END                                                                CONS



      FUNCTION SUM(Y,N)                                                  CONS
      DIMENSION Y(1)                                                     CONS
      SUM=0.                                                             CONS
      DO1J=1,N                                                           CONS
1     SUM=SUM +Y(J)                                                      CONS
      RETURN                                                             CONS
      END                                                                CONS
```

```
      SUBROUTINE RESP                                                   CONS
C                                                                       CONS
C     THIS SUBROUTINE COMPUTES THE RESPIRATION RATE (G/IND/DAY) FOR THE CONS
C        CONSUMER SPECIES                                               CONS
C                                                                       CONS
      COMMON/L1/T2,T3,SM,VA(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),    CONS
     1MC(6),SO(4)                                                       CONS
      COMMON/CONS/XF(5),XK(5,12),THO(12),TT(12),EF(12),MJ(5),TWU(5),TWL(CONS
     15),AF(5),AK(5,12),HC(5,12),XC(12),B(5),RE(5),D(5),XIM(5),EM(5),EFF CONS
     2(5),XINT(5)                                                       CONS
      COMMON/CONS2/RE1(5),RE2(5)                                        CONS
      DO 10 I=1,5                                                       CONS
      RE(I)=0.                                                          CONS
   10 IF(W(I).GT.0.) RE(I)=RE1(I)*W(I)**RE2(I)                          CONS
      RETURN                                                            CONS
      END                                                               CONS
```

```
      SUBROUTINE BIRTH                                                  CONS
      COMMON/CONS/XF(5),XK(5,12),THO(12),TT(12),EF(12),MJ(5),TWU(5),TWL(CONS
     15),AF(5),AK(5,12),HC(5,12),XC(12),B(5),RE(5),D(5),XIM(5),EM(5),EFF CONS
     2(5),XINT(5)                                                       CONS
      COMMON/L3/T,TD,TDMOD,TH,THMOD,TMMOD,TWMOD,TY                      CONS
C     THIS SUBROUTINE COMPUTES DAILY INSTANTANEOUS BIRTH RATES FOR      CONS
C     CONSUMER SPECIES                                                  CONS
      DO 1 I=1,5                                                        CONS
    1 B(I)=0                                                            CONS
      IF(TDMOD.LE.252..AND.TDMOD.GE.58.)B(1)=.007*(1.-.0046*(TDMOD-58.)) CONS
      IF(TDMOD.LE.168..AND.TDMOD.GE.155.) B(4)=.0308                    CONS
      IF(TDMOD.LE.203..AND.TDMOD.GE.190.) B(4)=.0102                    CONS
      IF(TDMOD.LE.224..AND.TDMOD.GE.162.)B(5)=.0214*(1.-.0145*(TDMOD-162 CONS
     1.))                                                               CONS
      RETURN                                                            CONS
      END                                                               CONS
```

```
      SUBROUTINE DEATH                                                  CONS
      COMMON/CONS/XF(5),XK(5,12),THO(12),TT(12),EF(12),MJ(5),TWU(5),TWL(CONS
     15),AF(5),AK(5,12),HC(5,12),XC(12),B(5),RE(5),D(5),XIM(5),EM(5),EFF CONS
     2(5),XINT(5)                                                       CONS
      COMMON/L3/T,TD,TDMOD,TH,THMOD,TMMOD,TWMOD,TY                      CONS
C     THIS SUBROUTINE COMPUTES THE INSTANTNAEOUS DAILY NATURAL MORTALITY CONS
C     RATES FOR THE CONSUMER SPECIES.                                   CONS
      D(1)=.0022                                                        CONS
      D(2)=0.                                                           CONS
      D(3)=0.                                                           CONS
      D(4)=.0035                                                        CONS
      D(5)=0.                                                           CONS
      IF(TDMOD.LE.154.) D(4)=.0021                                      CONS
      IF(TDMOD.LE.259..AND.TDMOD.GE.162.)D(5)=.0071                     CONS
      RETURN                                                            CONS
      END                                                               CONS
```

```
      SUBROUTINE MIGR                                                   CONS
      COMMON/L3/T,TD,TDMOD,TH,THMOD,TMMOD,TWMOD,TY                      CONS
      COMMON/CONS/XF(5),XK(5,12),THO(12),TT(12),EF(12),MJ(5),TWU(5),TWL(CONS
     15),AF(5),AK(5,12),HC(5,12),XC(12),B(5),RE(5),D(5),XIM(5),EM(5),EFF CONS
     2(5),XINT(5)                                                       CONS
      COMMON/L1/T2,T3,SM,VA(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),    CONS
     1MC(6),SO(4)                                                       CONS
C     THIS SUBROUTINE COMPUTES THE DAILY RATES OF IMMIGRATION AND       CONS
C     EMIGRATION FOR THE CONSUMER SPECIES                               CONS
      DO1I=1,5                                                          CONS
      XIM(I)=0.                                                         CONS
1     EM(I)=0.                                                          CONS
      IF(TDMOD.LE.126..AND.TDMOD.GE.113.)XIM(4)=.26E-8                  CONS
      IF(TDMOD.GE.218..AND.TDMOD.LT.235.)EM(4)=AMIN1(.21E-8,PN(4))      CONS
      RETURN                                                            CONS
      END                                                               CONS
```

```
      SUBROUTINE DECOM                                                  DECOM
      CALL MICR                                                         DECOM
      CALL SOIL                                                         DECOM
      CALL NITR                                                         DECOM
      RETURN                                                            DECOM
      END                                                               DECOM
```

```
      SUBROUTINE MICR                                                   DECOM
      COMMON/L1/T2,T3,SM,VA(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),    DECOM
     1MC(6),SO(4)                                                       DECOM
      COMMON/L2/DT2,DT3,DSM,DVA(4),DVB,DVS,DVL,DC(6),DPN(6),DW(6),DAD,  DECOM
     1DSH,DMN(4),DMC(6),DSO(4)                                          DECOM
      REAL MN,MC                                                        DECOM
      REAL MCMAX                                                        DECOM
      DO 10 I=1,6                                                       DECOM
      C1=MCMAX(I)                                                       DECOM
   10 DMC(I)=DIV(I)*MC(I)*(C1-MC(I))/C1                                 DECOM
      RETURN                                                            DECOM
      END                                                               DECOM
```

```
      SUBROUTINE SOIL                                                   DECOM
      COMMON/L1/T2,T3,SM,VA(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),    DECOM
     1MC(6),SO(4)                                                       DECOM
      REAL MN,MC                                                        DECOM
      COMMON/L2/DT2,DT3,DSM,DVA(4),DVB,DVS,DVL,DC(6),DPN(6),DW(6),DAD,  DECOM
     1DSH,DMN(4),DMC(6),DSO(4)                                          DECOM
      COMMON/DECOM1/DIVS(6),MCMAXS(6),MT1(6),MT2(6),MM1(6),MM2(6),      DECOM
     1SLD(6),SLMX(6),VMND(6),VMNMX(6),SBMN(6),SOMDIS(4),DECRATE(6),     DECOM
     2SOMDEC,NFIXR,DENITR,NPREF(4)                                      DECOM
      REAL MT1,MCMAXS,MT2,MM1,MM2,NFIXR,NPREF                           DECOM
      COMMON/L4/HA(12),HAD,HSH,HB,HS,HL,HM                              DECOM
      DSO(2)=SOMDIS(2)*HL-DECRATE(6)*MC(6)*SO(2)                        DECOM
      DSO(3)=SOMDIS(3)*HL-SOMDEC*SO(3)                                  DECOM
      DSO(1)=DSO(4)=0.                                                  DECOM
      RETURN                                                            DECOM
      END                                                               DECOM
```

```
      SUBROUTINE NITR                                                   DECOM
      COMMON/L1/T2,T3,SM,VA(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),    DECOM
     1MC(6),SO(4)                                                       DECOM
      REAL MN,MC                                                        DECOM
      COMMON/L2/DT2,DT3,DSM,DVA(4),DVB,DVS,DVL,DC(6),DPN(6),DW(6),DAD,  DECOM
     1DSH,DMN(4),DMC(6),DSO(4)                                          DECOM
      COMMON/DECOM1/DIVS(6),MCMAXS(6),MT1(6),MT2(6),MM1(6),MM2(6),      DECOM
     1SLD(6),SLMX(6),VMND(6),VMNMX(6),SBMN(6),SOMDIS(4),DECRATE(6),     DECOM
     2SOMDEC,NFIXR,DENITR,NPREF(4)                                      DECOM
      REAL MT1,MCMAXS,MT2,MM1,MM2,NFIXR,NPREF                           DECOM
      COMMON/L4/HA(12),HAD,HSH,HB,HS,HL,HM                              DECOM
      COMMON/L5/T1,PP,IC,WS                                             DECOM
      DIMENSION C1(3)                                                   DECOM
      REAL NFIX,NITUP,LEACH                                             DECOM
      DO 10 I=1,3                                                       DECOM
   10 C1(I)=DECRATE(I)*MC(I)*MN(I)                                      DECOM
      DMN(1)=SOMDIS(1)*HL-C1(1)                                         DECOM
      DMN(2)=C1(1)+LEACH(PP)*VL-C1(2)                                   DECOM
      DMN(3)=C1(2)-C1(3)-AMUP(0.)                                       DECOM
      DMN(4)=C1(3)+NFIX(0.)-DENIT(MN(4))-NITUP(0.)                      DECOM
      RETURN                                                            DECOM
      END                                                               DECOM
```

```
      REAL FUNCTION NFIX(DUM)                                           DECOM
      COMMON/DECOM1/DIVS(6),MCMAXS(6),MT1(6),MT2(6),MM1(6),MM2(6),      DECOM
     1SLD(6),SLMX(6),VMND(6),VMNMX(6),SBMN(6),SOMDIS(4),DECRATE(6),     DECOM
     2SOMDEC,NFIXR,DENITR,NPREF(4)                                      DECOM
      REAL MT1,MCMAXS,MT2,MM1,MM2,NFIXR,NPREF                           DECOM
      COMMON/L1/T2,T3,SM,VA(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),    DECOM
     1MC(6),SO(4)                                                       DECOM
      REAL MN,MC                                                        DECOM
      NFIX=NFIXR*MC(5)                                                  DECOM
      RETURN                                                            DECOM
      END                                                               DECOM
```

```
      FUNCTION DENIT(NITR)                                              DECOM
      COMMON/L1/T2,T3,SM,VA(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),    DECOM
     1MC(6),SO(4)                                                       DECOM
      REAL MN,MC                                                        DECOM
      COMMON/DECOM1/DIVS(6),MCMAXS(6),MT1(6),MT2(6),MM1(6),MM2(6),      DECOM
     1SLD(6),SLMX(6),VMND(6),VMNMX(6),SBMN(6),SOMDIS(4),DECRATE(6),     DECOM
     2SOMDEC,NFIXR,DENITR,NPREF(4)                                      DECOM
      REAL MT1,MCMAXS,MT2,MM1,MM2,NFIXR,NPREF                           DECOM
      REAL NITR                                                         DECOM
      DENIT=DENITR*NITR*MC(4)                                           DECOM
      RETURN                                                            DECOM
      END                                                               DECOM
```

```
      REAL FUNCTION NITUP(DUM)                                          DECOM
      COMMON/L1/T2,T3,SM,V(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),     DECOM
     1MC(6),SO(4)                                                       DECOM
      COMMON/DECOM1/DIVS(6),MCMAXS(6),MT1(6),MT2(6),MM1(6),MM2(6),      DECOM
     1SLD(6),SLMX(6),VMND(6),VMNMX(6),SBMN(6),SOMDIS(4),DECRATE(6),     DECOM
     2SOMDEC,NFIXR,DENITR,NPREF(4)                                      DECOM
      REAL MT1,MCMAXS,MT2,MM1,MM2,NFIXR,NPREF                           DECOM
      COMMON/DECOM2/TPN                                                 DECOM
      REAL MN                                                           DECOM
      NITUP=NPREF(4)*MN(4)*TPN                                          DECOM
      RETURN                                                            DECOM
      END                                                               DECOM
```

```
      FUNCTION AMUP(DUM)                                                DECOM
      COMMON/L1/T2,T3,SM,V(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),     DECOM
     1MC(6),SO(4)                                                       DECOM
      COMMON/DECOM1/DIVS(6),MCMAXS(6),MT1(6),MT2(6),MM1(6),MM2(6),      DECOM
     1SLD(6),SLMX(6),VMND(6),VMNMX(6),SBMN(6),SOMDIS(4),DECRATE(6),     DECOM
     2SOMDEC,NFIXR,DENITR,NPREF(4)                                      DECOM
      REAL MT1,MCMAXS,MT2,MM1,MM2,NFIXR,NPREF                           DECOM
      COMMON/DECOM2/TPN                                                 DECOM
      REAL MN                                                           DECOM
      AMUP=NPREF(3)*MN(3)*TPN                                           DECOM
      RETURN                                                            DECOM
      END                                                               DECOM
```

```
      FUNCTION DIV(I)                                                  DECOM
      COMMON/L1/T2,T3,SM,VA(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),   DECOM
     1MC(6),SO(4)                                                      DECOM
      REAL MN,MC                                                       DECOM
      COMMON/DECOM1/DIVS(6),MCMAXS(6),MT1(6),MT2(6),MM1(6),MM2(6),     DECOM
     1SLD(6),SLMX(6),VMND(6),VMNMX(6),SBMN(6),SOMDIS(4),DECRATE(6),    DECOM
     2SOMDEC,NFIXR,DENITR,NPREF(4)                                     DECOM
      REAL MT1,MCMAXS,MT2,MM1,MM2,NFIXR,NPREF                          DECOM
      J=SBMN(I)                                                        DECOM
      DIV=DIVS(I)*PARAB(MT1(I),MT2(I),T3)*PARAB(MM1(I),MM2(I),SM)*     DECOM
     1SBSTR(SLD(I),VMND(I),MN(J))                                      DECOM
      RETURN                                                           DECOM
      END                                                              DECOM


      REAL FUNCTION MCMAX(K)                                           DECOM
      COMMON/L1/T2,T3,SM,VA(4),VB,VS,VL,C(6),PN(6),W(6),AD,SH,MN(4),   DECOM
     1MC(6),SO(4)                                                      DECOM
      REAL MN,MC                                                       DECOM
      COMMON/DECOM1/DIVS(6),MCMAXS(6),MT1(6),MT2(6),MM1(6),MM2(6),     DECOM
     1SLD(6),SLMX(6),VMND(6),VMNMX(6),SBMN(6),SOMDIS(4),DECRATE(6),    DECOM
     2SOMDEC,NFIXR,DENITR,NPREF(4)                                     DECOM
      REAL MT1,MCMAXS,MT2,MM1,MM2,NFIXR,NPREF                          DECOM
      I=K $ J=SBMN(I)                                                  DECOM
      MCMAX=MCMAXS(I)*PARAB(MT1(I),MT2(I),T3)*PARAB(MM1(I),MM2(I),SM)* DECOM
     1SBSTR(SLMX(I),VMNMX(I),MN(J))                                    DECOM
      IF (MCMAX.LT.2.E+8) MCMAX=2.E+8                                  DECOM
      RETURN                                                           DECOM
      END                                                              DECOM


      FUNCTION PARAB(X1,X2,X)                                          DECOM
      LOGICAL MID                                                      DECOM
      PARAB=0.                                                         DECOM
      IF (.N.MID(X1,X2,X)) RETURN                                      DECOM
      C1=X2-X1                                                         DECOM
      PARAB=-4.*(X-X1)*(X-X2)/(C1*C1)                                  DECOM
      RETURN                                                           DECOM
      END                                                              DECOM


      FUNCTION SBSTR(SL,VMN,V)                                         DECOM
      SBSTR=.001                                                       DECOM
      IF (V.LE.VMN) RETURN                                             DECOM
      SBSTR=1.                                                         DECOM
      IF (SL.EQ.0.) RETURN                                             DECOM
      SBSTR=(V-VMN)*SL                                                 DECOM
      RETURN                                                           DECOM
      END                                                              DECOM
```

```
      SUBROUTINE RDPLT1(NPL,MMM,NCMP,TSTEP,N1,N2,N3,N4,XL,XU,YL,YU)      PLOT
      DIMENSION NCMP(4)                                                  PLOT
C....READS IN THE PLOT PARAMETERS.                                       PLOT
C        NPL - NO. OF VARIABLES TO BE PLOTTED ≤ 4                        PLOT
C        MMM - THE NO. OF BLOCKS OF TEN LINES DESIRED IN THE GRID FOR    PLOT
C              THE INDEPENDENT VARIABLE                                  PLOT
C        NCMP(I) - THE SUBSCRIPTS POINTING TO THE DEPENDENT VARIABLES    PLOT
C              (FROM VECTOR V(J) IN ODE) TO BE PLOTTED (NPL IN NUMBER)   PLOT
C        TSTEP - TIME STEP FOR PLOTTING > 0.                             PLOT
      READ(5,100) NPL,MMM,TSTEP                                          PLOT
      IF(NPL.GT.4) NPL=4                                                 PLOT
      READ(5,101) (NCMP(I),I=1,NPL)                                      PLOT
C....INITIALIZES VARIABLES USED IN PLTSTOR                               PLOT
      N1=N2=N3=N4=0                                                      PLOT
      XL=YL=1.E250                                                       PLOT
      XU=YU=-XL                                                          PLOT
      RETURN                                                             PLOT
  100 FORMAT(2I5,F10.3)                                                  PLOT
  101 FORMAT(4I5)                                                        PLOT
      END                                                                PLOT
```

```
      SUBROUTINE PLTSTOR(N,NC,T,NCOM,V,TSTEP,X1,X2,X3,X4,Y1,Y2,Y3,Y4,     PLOT
     -      N1,N2,N3,N4,XL,XU,YL,YU,ADRG)                                 PLOT
      DIMENSION NC(4),V(1),X1(1),X2(1),X3(1),X4(1),Y1(1),Y2(1),Y3(1),     PLOT
     -      Y4(1),ADRG(1)                                                 PLOT
      XL=AMIN1(XL,T)                                                      PLOT
      XU=AMAX1(XU,T)                                                      PLOT
      DO 30 I=1,N                                                         PLOT
      J=NC(I)                                                             PLOT
      YL=AMIN1(YL,V(J))                                                   PLOT
      YU=AMAX1(YU,V(J))                                                   PLOT
      GO TO(5,10,15,20),I                                                 PLOT
    5 N1=N1+1                                                             PLOT
      IF(N1.GT.100) GO TO 35                                              PLOT
      X1(N1)=T                                                            PLOT
      Y1(N1)=V(J)                                                         PLOT
      IF (J.GT.50) Y1(N1)=ADRG(J-50)                                      PLOT
      GO TO 30                                                            PLOT
   10 N2=N2+1                                                             PLOT
      IF(N2.GT.100) GO TO 35                                              PLOT
      X2(N2)=T                                                            PLOT
      Y2(N2)=V(J)                                                         PLOT
      IF (J.GT.50) Y2(N1)=ADRG(J-50)                                      PLOT
      GO TO 30                                                            PLOT
   15 N3=N3+1                                                             PLOT
      IF(N3.GT.100) GO TO 35                                              PLOT
      X3(N3)=T                                                            PLOT
      Y3(N3)=V(J)                                                         PLOT
      IF (J.GT.50) Y3(N1)=ADRG(J-50)                                      PLOT
      GO TO 30                                                            PLOT
   20 N4=N4+1                                                             PLOT
      IF(N4.GT.100) GO TO 35                                              PLOT
      X4(N4)=T                                                            PLOT
      Y4(N4)=V(J)                                                         PLOT
      IF (J.GT.50) Y4(N1)=ADRG(J-50)                                      PLOT
   30 CONTINUE                                                            PLOT
      RETURN                                                              PLOT
   35 WRITE(6,200)                                                        PLOT
  200 FORMAT(4H ***,*THE NO. OF VALUES TO PLOT HAS EXCEEDED 100*)         PLOT
      RETURN                                                              PLOT
      END                                                                 PLOT
```

```
      SUBROUTINE PLT1(IERR,TEMP,N,M,XL,XU,YL,YU,X1,NOX1,Y1,X2,NOX2,Y2,   PLOT
     1X3,NOX3,Y3,X4,NOX4,Y4)                                             PLOT
      DIMENSION X1(1),Y1(1),X2(1),Y2(1),MASK(10),PLATE(11),STAMP(5)      PLOT
     1,X3(1),Y3(1),X4(1),Y4(1)                                           PLOT
      DIMENSION KIT(44),KITE(11),TSCAL(11)                               PLOT
      DIMENSION SCAL  (11)                                               PLOT
      INTEGER PLATE ,STAMP                                               PLOT
C     INITIALIZATION OF STAMPER AND MASKER                               PLOT
      MASK(10)=00000000000000000077B                                     PLOT
      MASK(1)=77000000000000000000B                                      PLOT
      MASK(2)=00770000000000000000B                                      PLOT
      DO2 I=1,7                                                          PLOT
    2 MASK(-I+10)=64*MASK(-I+11)                                         PLOT
      STAMP(1)=10H**********                                             PLOT
      STAMP(2)=10H0000000000                                             PLOT
      STAMP(3)=10HXXXXXXXXXX                                             PLOT
      STAMP(4)=10HHHHHHHHHHH                                             PLOT
      STAMP(5)=10HRRRRRRRRRR                                             PLOT
      DO 200 I=1,44                                                      PLOT
      KIT(I)=10H.                                                        PLOT
      IF(I.GT.11) GO TO 200                                              PLOT
      KITE(I)=10HV                                                       PLOT
      TSCAL(I)=10H0123456789                                             PLOT
  200 CONTINUE                                                           PLOT
      MPLATE=10H                                                         PLOT
      MP=10H+                                                            PLOT
      SINCY=(YU-YL)/100.                                                 PLOT
      SINCX=(XU-XL)/(10.*M)                                              PLOT
      HINCY=(YU-YL)/10.                                                  PLOT
      IFF=10*M+1                                                         PLOT
      DO1 I=1,11                                                         PLOT
    1 SCAL(I)=YL+(I-1)*HINCY                                             PLOT
      IERR=0                                                             PLOT
      PRINT2000,SCAL                                                     PLOT
      PRINT 3000,(KIT(K),K=1,44),(KITE(K),K=1,11),(TSCAL(K),K=1,11)      PLOT
      DO 100 LCTR=1,IFF                                                  PLOT
C     INITIALIZE PLATE (ERASE PREVIOUS STAMPINGS)                        PLOT
      NCTR=MOD(LCTR-1,10)                                                PLOT
      IF(NCTR)5,3,5                                                      PLOT
    3 DO 6 K=1,11                                                        PLOT
    6 PLATE(K)=MP                                                        PLOT
      GO TO 11                                                           PLOT
    5 DO 4 K=1,11                                                        PLOT
    4 PLATE(K)=MPLATE                                                    PLOT
C     SEARCH PROCESS BEGINS FOR THE X1 ARRAY                             PLOT
   11 CALL SUBPLT(X1,Y1,1,NOX1,LCTR,PLATE,MASK,STAMP,IERR,SINCX,SINCY,XL PLOT
     1,YL,XU,YU)                                                         PLOT
      IF(N-1)99,101,21                                                   PLOT
   21 CALL SUBPLT(X2,Y2,2,NOX2,LCTR,PLATE,MASK,STAMP,IERR,SINCX,SINCY,XL PLOT
     1,YL,XU,YU)                                                         PLOT
      IF(N-2)99,101,31                                                   PLOT
   31 CALL SUBPLT(X3,Y3,3,NOX3,LCTR,PLATE,MASK,STAMP,IERR,SINCX,SINCY,XL PLOT
     1,YL,XU,YU)                                                         PLOT
      IF(N-3)99,101,41                                                   PLOT
   41 CALL SUBPLT(X4,Y4,4,NOX4,LCTR,PLATE,MASK,STAMP,IERR,SINCX,SINCY,XL PLOT
     1,YL,XU,YU)                                                         PLOT
  101 IF(NCTR)106,105,106                                                PLOT
```

```
 105 SCALX=XL+(LCTR-1)*SINCX                                          PLOT
     PRINT 1001, SCALX, NCTR, PLATE,NCTR, SCALX                       PLOT
     GO TO 100                                                        PLOT
 106 PRINT 1000,NCTR,PLATE,NCTR                                       PLOT
 100 CONTINUE                                                         PLOT
     PRINT 3000,(TSCAL(J),J=1,11)                                     PLOT
     RETURN                                                           PLOT
  99 PRINT 3001,N                                                     PLOT
     CALL EXIT                                                        PLOT
1000 FORMAT(12X,I1,10A10,A1,I1)                                       PLOT
1001 FORMAT(2X,E9.2,1X,I1,10A10,A1,I1,1X,E9.2)                        PLOT
2000 FORMAT(7X,11F10.2)                                               PLOT
3000 FORMAT(13X,10A10,A1)                                             PLOT
3001 FORMAT(27H1HELP NEEDED IN PLOT FOR N=,I10)                       PLOT
     END                                                              PLOT
```

```
      SUBROUTINE SUBPLT(TX,TY,NN,NOX,LCTR,PLATE,MASK,STAMP,IERR,SINCX,SI PLOT
     1NCY,XL,YL,XU,YU)                                                  PLOT
      INTEGER PLATE, STAMP                                              PLOT
      DIMENSION TX(1),TY(1),MASK(10),PLATE(11),STAMP(5),IPW(101),IPC(101 PLOT
     1),IPP(5),ISYM(101)                                                PLOT
      IF(NN-1)99,30,21                                                  PLOT
   30 ISTC=0                                                            PLOT
      IF(LCTR-1)99,31,21                                                PLOT
   31 DO 6 ISTR=1,5                                                     PLOT
    6 IPP(ISTR)=1                                                       PLOT
   21 ID=IPP(NN)                                                        PLOT
      IF(ID-NOX)3,3,2                                                   PLOT
    3 ICPTR=ID                                                          PLOT
      DO 1 IPTR=ICPTR,NOX                                               PLOT
      IF((TX(IPTR).LT.XL).OR.(TX(IPTR).GT.XU)) GO TO 101                PLOT
      IX=(TX(IPTR)-XL)/SINCX+1.5                                        PLOT
    8 IF(IX-LCTR) 1, 10, 1                                              PLOT
   10 IF((TY(IPTR).LT.YL).OR.(TY(IPTR).GT.YU)) GO TO 101                PLOT
      IY=(TY(IPTR)-YL)/SINCY+1.5                                        PLOT
   11 IPWY=1+(IY-1)/10                                                  PLOT
      IPWC=IY + 10 -(10*IPWY)                                           PLOT
      IF(ISTC)57,57,58                                                  PLOT
   58 DO 12 KK=1,ISTC                                                   PLOT
      IF(IPWY - IPW(KK))12,13,12                                        PLOT
   13 IF(IPWC-IPC(KK))12,14,12                                          PLOT
   14 IF(ISYM(KK)-NN)15,9,15                                            PLOT
   12 CONTINUE                                                          PLOT
   57 PLATE(IPWY)=(PLATE(IPWY).AND..NOT.MASK(IPWC)).OR.(STAMP(NN).AND.MA PLOT
     1SK(IPWC))                                                         PLOT
      ISTC=ISTC#1                                                       PLOT
      IPW(ISTC)=IPWY                                                    PLOT
      IPC(ISTC)=IPWC                                                    PLOT
      ISYM(ISTC)=NN                                                     PLOT
      GO TO 9                                                           PLOT
  101 IERR=IERR+1                                                       PLOT
      GO TO 9                                                           PLOT
   15 PLATE(IPWY)=(PLATE(IPWY).AND..NOT.MASK(IPWC)).OR.(STAMP( 5).AND.MA PLOT
     1SK(IPWC))                                                         PLOT
    9 IF(IPTR-ID)17,17,1                                                PLOT
C   9 IF(IPTR-ID)35,17,1                                                PLOT
   17 ID=ID+1                                                           PLOT
    1 CONTINUE                                                          PLOT
      IPP(NN)=ID                                                        PLOT
    2 RETURN                                                            PLOT
   99 PRINT 35                                                          PLOT
   35 FORMAT( 34H1TROUBLE IN SUPLLT WITH NN OR LCTR)                    PLOT
      CALL EXIT                                                         PLOT
      END                                                               PLOT
```

Appendix D. Description of Program Input/Output Format and Parameter Values

Following is a description and listing of the data cards read by the program. The main program (ODE) reads in the first three cards and the group of cards labeled INIT 1 through INIT 7. The remaining data cards are read by subroutine INIT, and the labels in columns 73 through 80 refer to the name of the labeled common block into which the values on the card are stored.

A typical sample of printed output appears on the next page. Specific items are described below.

Item No.	Description
1	These are integration parameters used in both the Euler and Kutta methods. The first is the relative accuracy desired when integrating by the Kutta method. The second is a fraction which, when multiplied by the time increment, is the integration step size. The third is the maximum number of halvings of the integration step size in the Kutta method.
2	These are system options. Here requested are: the debugging options which outputs on an external storage devise (unit no. 8) the PSV's and their derivatives and most ISV's, integration by a fifth order Runge-Kutta method, no plotting, no calls to subroutine CHKOUT, and printer output requested.
3	This is a sampling every 20 days of the weather record used as driving variables.
4	This is an echo check of the parameter values read into each labeled common block.
5	This is a sample of simulation printed output for two days. Printed is the time in days at the left followed by the values of the PSV's from left to right, followed by their corresponding derivatives.

```
DE SOLUTION
 PAWNEE MODEL MOD 1 -- ABIO. PROD. CONS. DECOM -- DECK SETUP 1/25/71

TIME STARTS AT     140.500, INCREMENTS BY          1.00000 FOR    100 STEPS

   46 EQUATIONS.    730 CONSTANTS                    1           2
KUTTA PARAMETERS ARE        .01000       .10000     1
SYSTEM OPTIONS -- DEBUG      KUTTA      NOPLOT      NOCHK       WRIT

DRIVING FUNCTION  3
TMIN.TMAX.PPT
```

1	-16.11	-1.567	0.
21	-6.667	13.89	0.
41	-11.67	11.11	0.
61	-8.333	2.222	0.
81	-6.667	10.56	0.
101	-.5556	23.33	0.
121	6.111	23.89	0.
141	4.444	20.56	.1778
161	8.889	18.89	0.
181	12.22	27.22	0.
201	13.89	31.11	0.
221	11.11	17.78	2.997
241	6.111	18.33	.2794
261	8.889	25.56	0.
281	1.111	7.778	0.
301	-1.667	22.78	0.
321	-16.67	-3.889	0.
341	-12.78	-1.111	0.
361	-17.78	-.5556	.2032
381	-9.444	12.22	0.
401	-5.000	11.67	0.
421	-5.556	15.56	0.
441	-2.222	15.56	0.
461	0.	22.78	0.
481	-5.000	11.11	0.
501	6.111	13.89	.9404
521	6.667	32.22	0.
541	7.778	17.78	0.
561	15.00	32.22	5.0800E-02
581	12.78	31.11	0.
601	12.22	24.44	0.
621	6.667	24.33	0.
641	2.778	26.11	.8890
661	1.111	12.22	0.
681	-2.222	11.67	0.
701	-5.556	15.56	0.
721	-4.444	12.22	0.

```
SYSTEM PARAMETERS  (4)

ABIOTIC
COMMON ABIO1
SC.PHEXP.STLAG.SMLAG
 1.50000E-02  .25000     6.41667E-05 4.01000E-07

PRODUCER
COMMON PROD1
T20(4).T21(4).CP(4).K1(4).K2(4).K3(4).K4(4).K5(4).K6(4).K7(4).K8(4).ETE(4).L(4).SC(4).EPSI(4)
  42.000      34.000      36.000      50.000      .20100      .21100      .21100      .18300      1.0000      1.0000
  1.0000      1.0000      3.36000E-04 2.05500E-04 2.23000E-04 5.70300E-05 .16667      .25000      .25000      .10000
  59.000      58.900      58.900      58.900      1.42860E-05 1.66670E-05 1.66670E-05 1.20000E-05 2.00000E-02 2.32558E-02
 2.32555E-02 1.66667E-02 6.94000E-07 6.94000E-07 8.94000E-07 6.02000E-07 .40000      .40000      .40000      .40000
  .50000      .50000      .50000      .50000      1.0000      1.0000      1.0000      1.0000      1.0000      1.0000
  1.0000      .15000      1.70000E-03 1.70000E-03 1.70000E-03 1.70000E-03 .50000      .50000      .80000      .20000

COMMON HARV1
TOLTT1.TOLTT2.TMNLT.TMXLT
  1.00000E-08  2.00000E-08  0.          6.00000E+01

COMMON PROD2
VAHRK. VRRRK. TASLP1. TRMX. TASLP2
  5.00000E-04  2.00000E-04  2.00000E-05  1.00000E-04
  5.00000E-04  1.00000E-04  1.00000E-04  3.00000E-04
  7.00000E-08  5.80000E-08  1.00000E-07  7.00000E-08
  1.16000E-06  1.16000E-06  4.00000E-06  4.00000E-06
  2.32000E-05  1.16000E-04  4.00000E-04  1.33000E-04

COMMON PROD3
RA1. RA2. RA3
  1.91574E-06  4.40000E+01  4.00000E+00
PMO1(4). PMO2(4)
  5.00000E+01  2.50000E+01  1.00000E+01  1.66667E+01
  8.00000E-02  1.40000E-01  3.00000E-01  1.80000E-01

COMMON CONS2
RF1(5). RF2(5)
  5.40000E-01  2.10000E+00  3.40000E+00  2.80000E-01  1.32000E-01
  6.70000E-01  7.50000E-01  4.70000E-01  7.50000E-01  8.40000E-01


CONSUMERS
COMMON CONS1
KF          .20000       .20000       .20000       .15000       .27000
KK          5.00000E-02  .12000       .70000       7.00000E-02  0.           5.90000E-02  1.00000E-03
            0.           0.           0.           0.           0.
KK          .25000       .25000       .45000       1.50000E-02  0.           3.00000E-02  5.00000E-03
            0.           0.           0.           0.           0.
KK          0.           0.           0.           0.           0.           0.           0.
            .45000       0.           0.           5.00000E-02  .10000
KS          0.           .17500       .17500       0.           0.           0.           0.
            0.           0.           0.           0.           .65000
TT          2.00000E-03  2.00000E-04  5.00000E-04  2.00000E-03  2.82000E-02  4.70000E-03  2.40000E-03
            2.50000E-05  1.0000       1.0000       5.40000E-07  7.50000E-05
THO         1.00000E-03  1.00000E-04  2.00000E-04  1.00000E-03  1.41000E-02  2.30000E-03  1.20000E-03
            1.25000E-05  0.           0.           2.70000E-07  3.80000E-05
TWI. TWO    100.00       100.00       1.81000E+05  1.81000E+05  9000.0       9000.0       30.000       30.000       .30000       .30000

NI     3     3     8     12     3
```

```
COMMON DECOM1
DIVS, MCMAXS, MT1, MT2, MM1, MM2, SLO, SLMX, VMND, VMNMX, SBMN, DECRATE
  5.77000E-04  5.77000E-04  5.77000E-04  5.77000E-04  5.77000E-04  5.77000E-04
  2.00000E+11  2.00000E+11  2.00000E+11  2.00000E+11  2.00000E+11  2.00000E+11
  5.00000E+00  5.00000E+00  5.00000E+00  5.00000E+00  5.00000E+00  5.00000E+00
  4.30000E+01  4.30000E+01  4.30000E+01  4.30000E+01  4.30000E+01  4.30000E+01
  1.00000E+00  1.00000E+00  1.00000E+00  1.00000E+00  1.00000E+00  1.00000E+00
  1.20000E+01  1.20000E+01  1.20000E+01  1.20000E+01  1.20000E+01  1.20000E+01
  0.           0.           0.           0.           0.           0.
  7.77000E+04  7.77000E+04  7.77000E+04  7.77000E+04  7.77000E+04  7.77000E+04
  0.           0.           0.           0.           0.           0.
  1.43000E-06  1.43000E-06  1.43000E-06  1.43000E-06  1.43000E-06  1.43000E-06
  1.00000E+00  2.00000E+00  3.00000E+00  4.00000E+00  1.10000E+01  1.20000E+01
  1.73000E-16  1.34000E-15  1.39000E-15  0.           0.           8.27000E-18
SOMDIS, NPREF
  5.00000E-02  3.00000E-01  7.00000E-01  0.
  0.           0.           2.00000E+01  1.00000E+00
SOMDEC, NFIXP, DENITP
  3.17000E-08       0.                0.
```

TIME, VARIABLE VALUES (5)

```
   140      0.          0.          -12.500      20.000       1.0000      3.00000E-04 1.00000E-04 1.00000E-05 3.00000E-04 8.00000E-02
            3.00000E-03 3.00000E-03 8.34461E-05 5.20360E-05 8.99330E-09 0.          1.18780E-04 0.          9.95610E-07 2.00000E-10
            1.00000E-12 0.          5.00000E-04 0.           83.814     2.60180E+05  8993.3      35.280       .23756     0.
            4.02840E-07 3.34720E-05 1.43000E-05 1.43000E-05 1.43000E-05 1.43000E-05 2.00000E+11 2.00000E+11 2.00000E+11 2.00000E+11
            2.00000E+11 2.00000E+11  1.4000       .28000      1.1200     0.          -12.500     0.          0.          0.

DERIVATIVE  0.          0.          0.          -8.52831E-05-4.01000E-07 4.63579E-07 2.91845E-07 5.08629E-08-1.80341E-10-1.08102E-06
           -3.61049E-09 3.52544E-09 0.          0.          0.          0.          0.          0.          2.47001E-14 0.
            0.          0.          -1.06843E-19 0.         -8.29247E-05 -.20150    -3.39313E-05-3.81509E-06-3.00637E-07 0.
            2.12478E-12 1.81047E-10-4.91780E-10-3.48062E-09-5.60198E-14 3.97540E-09 0.          0.          0.          0.
            0.          0.          0.          -4.63102E-07-3.54620E-08 0.


   141       141.50      4.5935      18.931      16.578       .96595     5.07465E-04 5.54754E-04 5.40198E-05 2.84945E-04 6.42323E-02
            2.88826E-03 3.10461E-03 7.57697E-05 4.86734E-05 8.99055E-09 0.          1.06663E-04 0.          9.97730E-07 2.00000E-10
            1.00000E-12 0.          5.00000E-04 0.           76.944     2.43367E+05  8990.5      34.964       .21333     0.
            5.79912E-07 4.92165E-05 8.04397E-07 1.12733E-07 1.27265E-07 5.64203E-05 2.00000E+11 2.00000E+11 2.00000E+11 2.00000E+11
            2.00000E+11 2.00000E+11  1.4000       .24272      1.1169     0.           18.931     0.          1.50000E-02  447.00

DERIVATIVE  0.          0.          0.          1.50945E-04-3.87345E-07-3.37237E-10-3.56952E-10-2.65241E-11-1.68353E-10-1.63631E-07
           -3.47534E-09 3.39774E-09 0.          0.          0.          0.          0.          0.          2.43809E-14 0.
            0.          0.          -1.18981E-19 0.         -7.61600E-05 -.18781    -2.99081E-05-3.49978E-06-2.61161E-07 0.
            1.95478E-12 1.81161E-10-2.47275E-11-3.50763E-12-4.04087E-12 3.53598E-11 0.          0.          0.          0.
            0.          0.          0.          -4.01433E-07-3.53635E-08 0.


   142       142.50      4.7258      8.0440      7.6923      1.8485     1.04426E-03 5.22515E-04 5.15804E-05 2.67557E-04 5.39368E-02
            2.77325E-03 3.11244E-03 7.04433E-05 4.53861E-05 8.98812E-09 0.          9.58772E-05 0.          9.99823E-07 2.00000E-10
            1.00000E-12 0.          5.00000E-04 0.           70.506     2.26930E+05  8988.1      34.675       .19175     0.
            7.43160E-07 6.56081E-35 3.55647+139-3.59514+139 3.83179+137-6.26458+121-9.03116E+26-5.60943E+35-3.15770E+16 1.27888E+11
            3.26555E+11 3.26236E+11  1.4000       .20874      1.1139     0.           8.0440     0.          1.50000E-02  447.00
```

(Output discontinued for lack of space)

Description of Input Data Format

Card Group	Card Nos.	Columns	Format	Variable Names	Description
Blank	1	1-80	A810	-	Heading for simulation
Blank	2	1- 5	I5	NCOM	No. of PSV's
Blank	2	6-10	I5	NP	No. of constants for weather record
Blank	2	11-15	I5	NT	No. of print steps
Blank	2	16-25	F10.3	T1	Starting time
Blank	2	26-35	F10.3	TDEL	Time step for printing
Blank	2	36-45	F10.3	ACC	Relative accuracy of integration
Blnak	2	46-55	F10.3	DEL	DEL*TDEL is step size for integration
Blank	2	56-60	I5	ITER	Maximum no. of halvings of integration step size
Blank	3	1-10	A10	DB	Debugging option
Blank	3	11-20	A10	DESOLV	Method of integration
Blank	3	21-30	A10	PLOT	Printer plotting option (requires additional data cards)
Blank	3	31-40	A10	CHK	Subroutine CHKOUT call option
Blank	3	41-50	A10	WRIT	Printed output option
INIT	1-7	--	E10.6	V(I)	Initial values of PSV's (NCOM in number)
ABIO1	1	--	--	--	Parameter values for common block
PROD1	1-9	--	--	--	Parameter values for common block
HARV1	1		--	--	Parameter values for common block
PROD2	1-3				Parameter values for common block
PROD3	1-3				Parameter values for common block
CONS2	1-2				Parameter values for common block
CONS1	1-9,A-0				Parameter values for common block
DECOM1					Parameter values for common block

Subroutine INPUT is called where daily minimum temperature, daily maximum temperature, and daily precipitation is read in from an external storage device (unit no. 1). NP days of weather information are read.

Listing of Input Data Cards

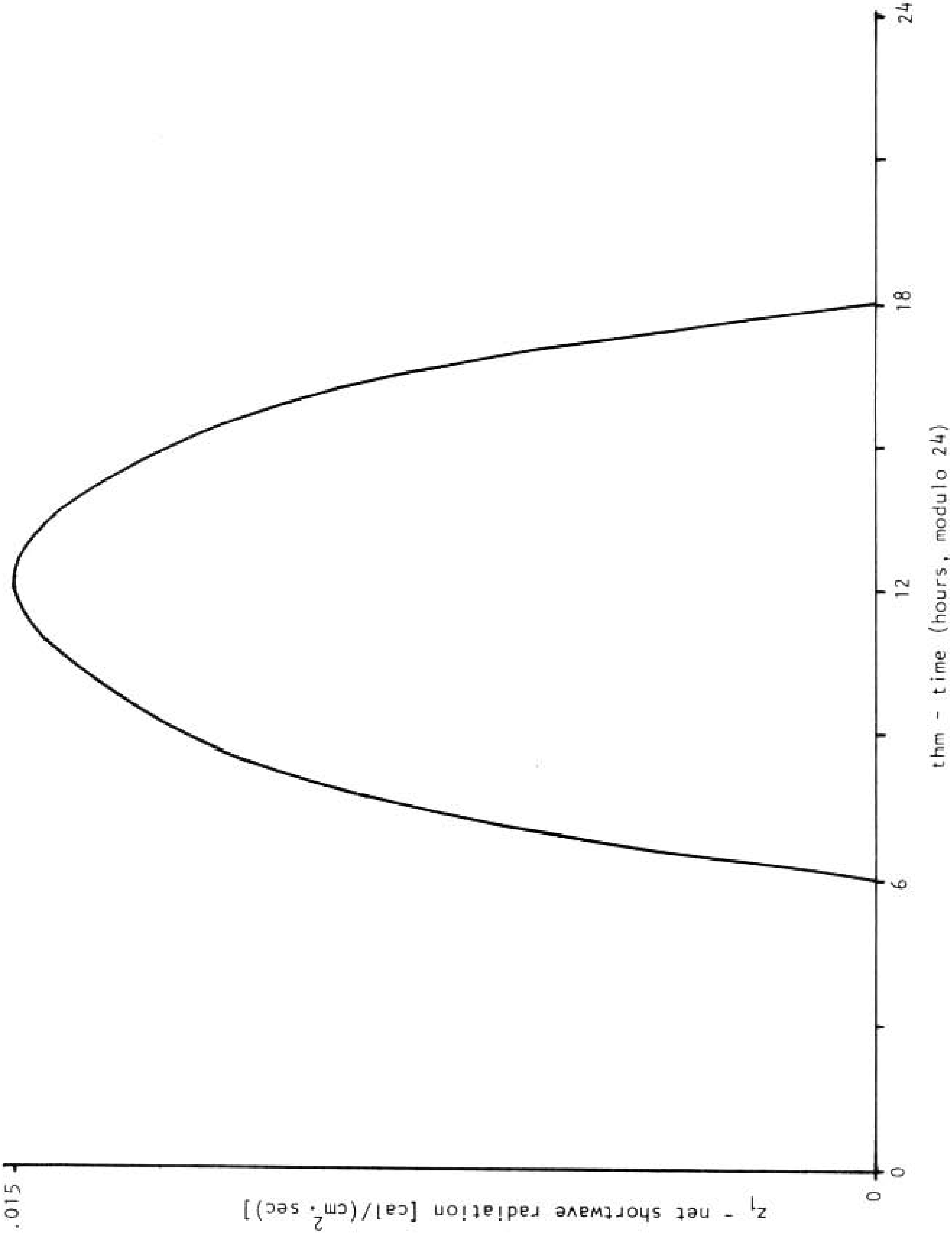
```
COLUMN NUMBER:

================================================================================
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
================================================================================

 PAWNEE MODEL MOD 1 -- ABIO. PROD. CONS. DECOM -- DECK SETUP 1/25/71
   46  730  100     140.50        1.0       .01       0.5   10                 2
DEBUG     KUTTA     NOPLOT    NOCHK     WRIT                                   3
2.4951    .080484   10.       20.       1.         .0003     .0001      INIT 1
.00001    .0003     .08       .003      .003       .000083446.000052037 INIT 2
.89933E-080.        .00011879 0.        .99561E-06.20000E-09.10000E-11 INIT 3
0.        .0005     0.        83.814    .26018E+068993.3    35.28      INIT 4
.23756    0.        .40284E-06.33472E-04 .143E-4   .143E-4   .143E-4   INIT 5
   .143E-4    2.E+11    2.E+11    2.E+11    2.E+11    2.E+11    2.E+11 INIT 6
      1.4       .28      1.12        0.                                INIT 7
.015      .25       .641667E-4 .401E-6                                 ABIO1 1
      42.       36.       36.       50.      .201      .211      .211 PROD1 1
      .183       1.        1.        1.       1..000336   2.055E-4     PROD1 2
   2.23E-4  5.703E-5.166666667.25       .25       .1        59.        PROD1 3
58.9      58.9      58.9      .000014286.000016667.000016667.000012    PROD1 4
.02       .023255814.023255514.016666667 6.94E-7   6.94E-7   8.94E-7   PROD1 5
   6.02E-7        .4        .4        .4        .4        .5        .5 PROD1 6
      .5        .5       1.         1.        1.        1.   1.        PROD1 7
1.        1.          .15      .0017       .0017     .0017     .0017   PROD1 8
      .5        .5        .8       .2                                  PROD1 9
      .1E-7     .2E-7       0.        60.                              HARV1 1
      .5E-3     .2E-3     .2E-4     .1E-3      .5E-3     .1E-3     .1E-3 PROD2 1
      .3E-3     .7E-7     .58E-7    1.E-7      .7E-7   1.16E-6   1.16E-6 PROD2 2
      4.E-6     4.E-6    .232E-4   1.16E-4     4.E-4   1.33E-4           PROD2 3
.191574E-544.          8.                                              PROD3 1
50.       25.       10.       16.66667                                 PROD3 2
0.08      0.14      0.30      0.18                                     PROD3 3
0.54      2.1       3.8       0.28      0.132                          CONS2 1
0.67      0.75      0.67      0.75      0.84                           CONS2 2
  0.2  0.2  0.2 0.16 0.27                                            CONS 1 1
  .05  .12  .70  .07    0 .059 .001    0    0    0    0    0          CONS 1 2
  .25  .25  .45 .015    0  .03 .005    0    0    0    0    0          CONS 1 3
    0    0    0    0    0    0    0  .85    0    0  .05  .10          CONS 1 4
    0 .175 .175    0    0    0    0    0    0    0    0  .65          CONS 1 5
  .05  .12  .70  .07    0 .059 .001    0    0    0    0    0          CONS 1 6
  .30  .50  .70  .35    0  .25  .20  .95    0    0  .90  .85          CONS 1 7
      .001      .002                                                 CONS 1 8
     .0001     .0002                                                 CONS 1 9
```

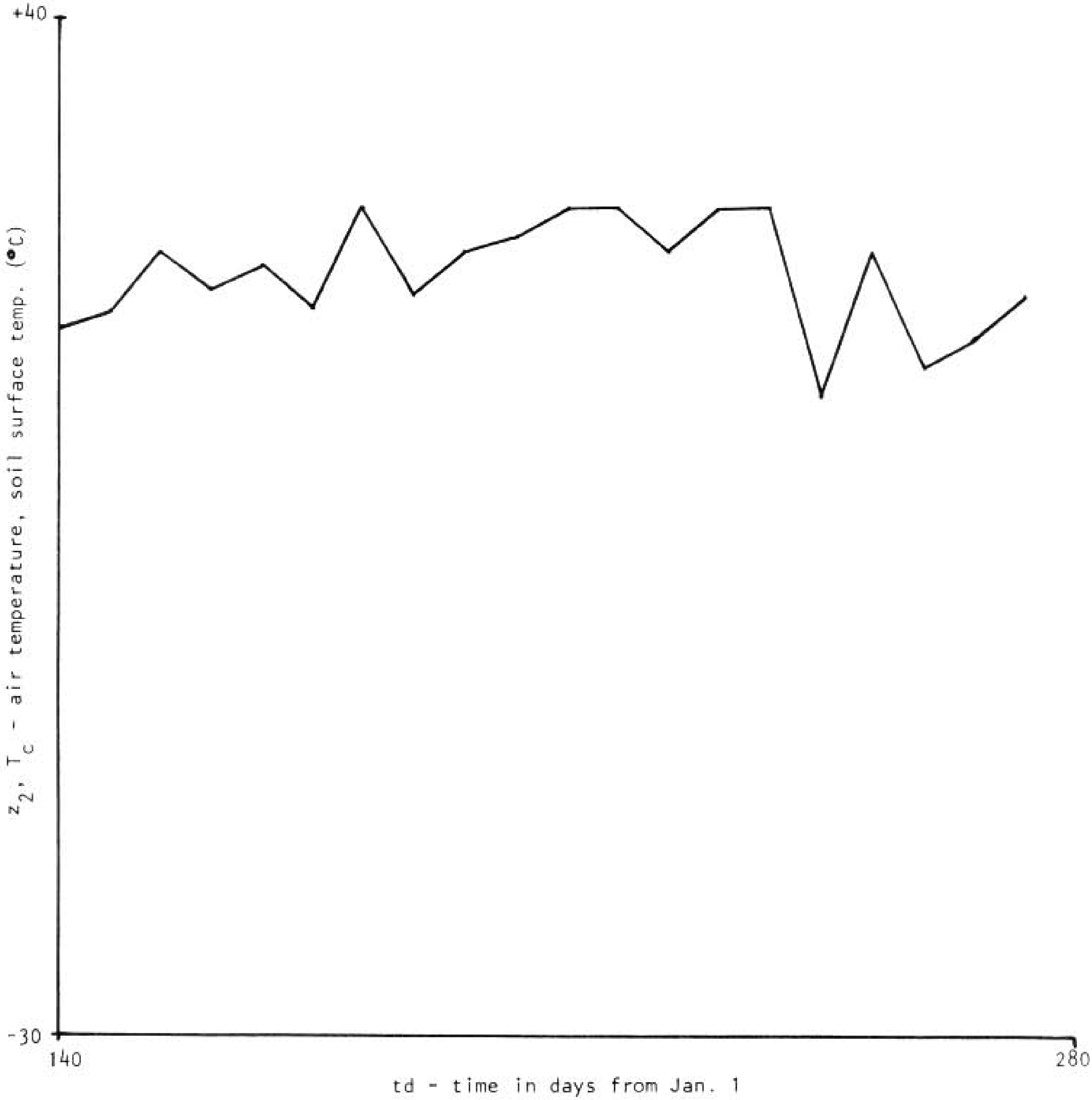
```
      .0002      .0005                                                                                    CONS  1  A
       .001       .002                                                                                    CONS  1  B
      .0141      .0282                                                                                    CONS  1  C
      .0023      .0047                                                                                    CONS  1  D
      .0012      .0024                                                                                    CONS  1  E
    .125E-4     .25E-4                                                                                    CONS  1  F
0.         1.0                                                                                            CONS  1  G
0.         1.0                                                                                            CONS  1  H
     2.7E-7     5.4E-7                                                                                    CONS  1  I
     .38E-4     .75E-4                                                                                    CONS  1  J
      .1E+3      .1E+3                                                                                    CONS  1  K
    .181E+6    .181E+6                                                                                    CONS  1  L
      .9E+4      .9E+4                                                                                    CONS  1  M
      .3E+2      .3E+2                                                                                    CONS  1  N
         .3         .3                                                                                    CONS  1  O
    .577E-3    .577E-3    .577E-3    .577E-3    .577E-3    .577E-3  DIVS                                  DECOM1  1
     2.E+11     2.E+11     2.E+11     2.E+11     2.E+11     2.E+11  MCMAXS                                DECOM1  2
         5.         5.         5.         5.         5.         5.  MT1                                   DECOM1  3
        43.        43.        43.        43.        43.        43.  MT2                                   DECOM1  4
         1.         1.         1.         1.         1.         1.  MM1                                   DECOM1  5
        12.        12.        12.        12.        12.        12.  MM2                                   DECOM1  6
         0.         0.         0.         0.         0.         0.  SLD                                   DECOM1  7
    7.77E+4    7.77E+4    7.77E+4    7.77E+4    7.77E+4    7.77E+4  SLMX                                  DECOM1  8
         0.         0.         0.         0.         0.         0.  VMND                                  DECOM1  9
    .143E-5    .143E-5    .143E-5    .143E-5    .143E-5    .143E-5  VMNMX                                 DECOM110
         1.         2.         3.         4.        11.        12.  SBMN                                  DECOM111
   1.73E-16   1.39E-15   1.39E-15         0.         0.   8.27E-18  DECRATE                               DECOM112
        .05         .3         .7         0.                        SOMDIS                                DECOM113
         0.         0.        20.         1.                        NPREF                                 DECOM114
    3.17E-8         0.         0.                                   MISC                                  DECOM115
     END OF RUN
```

Appendix E. Numerical Solutions of Model Differential Equations

Following are graphs of the principal system variables and driving variables as a function of time for the period 140 through 280 days. The input driving variables were derived from 1967 records of daily precipitation, minimum and maximum temperature of the Akron, Colorado Experiment Station, operated by the Agricultural Research Service of the USDA. Model output was recorded daily at 12:00 noon, thus those features of the model which occur only at times other than 12:00 noon, e.g., rain, can not be observed in these graphs. For some of the driving variables (z_1, z_3, and z_4), a graph of related parameters or daily variation has been provided. Parameter values are given in Appendix D.

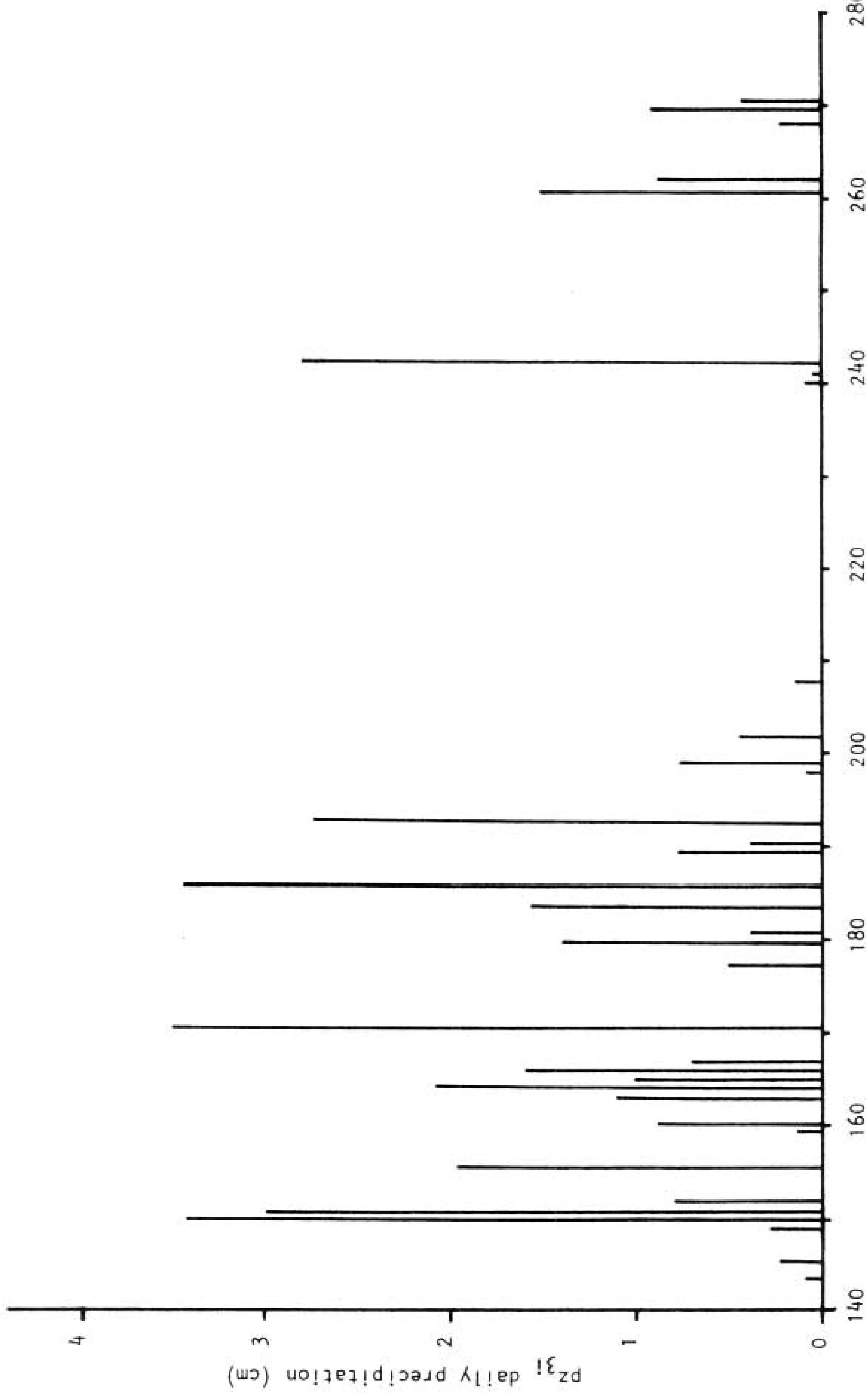
Since neither the functional forms nor the parameter values have been extensively modified from their original formulations for this set of output, the reader may expect to find anomalous results in these graphs. The variables of the decomposer section were not graphed because the values stayed essentially constant during the simulation.

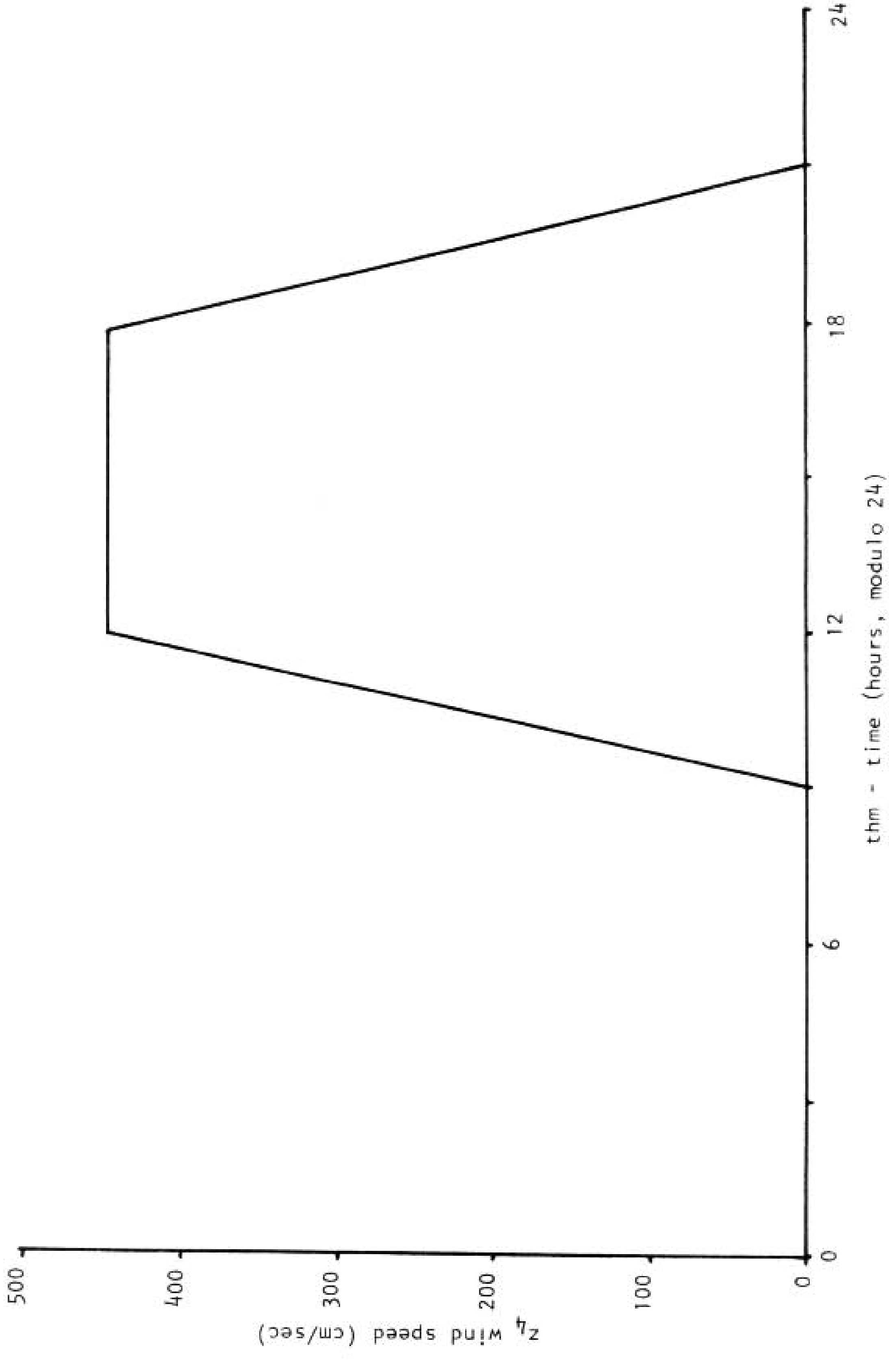

.015
0
0
6
12
18
24
z_1 - net shortwave radiation [cal/(cm^2· sec)]
thm - time (hours, modulo 24)

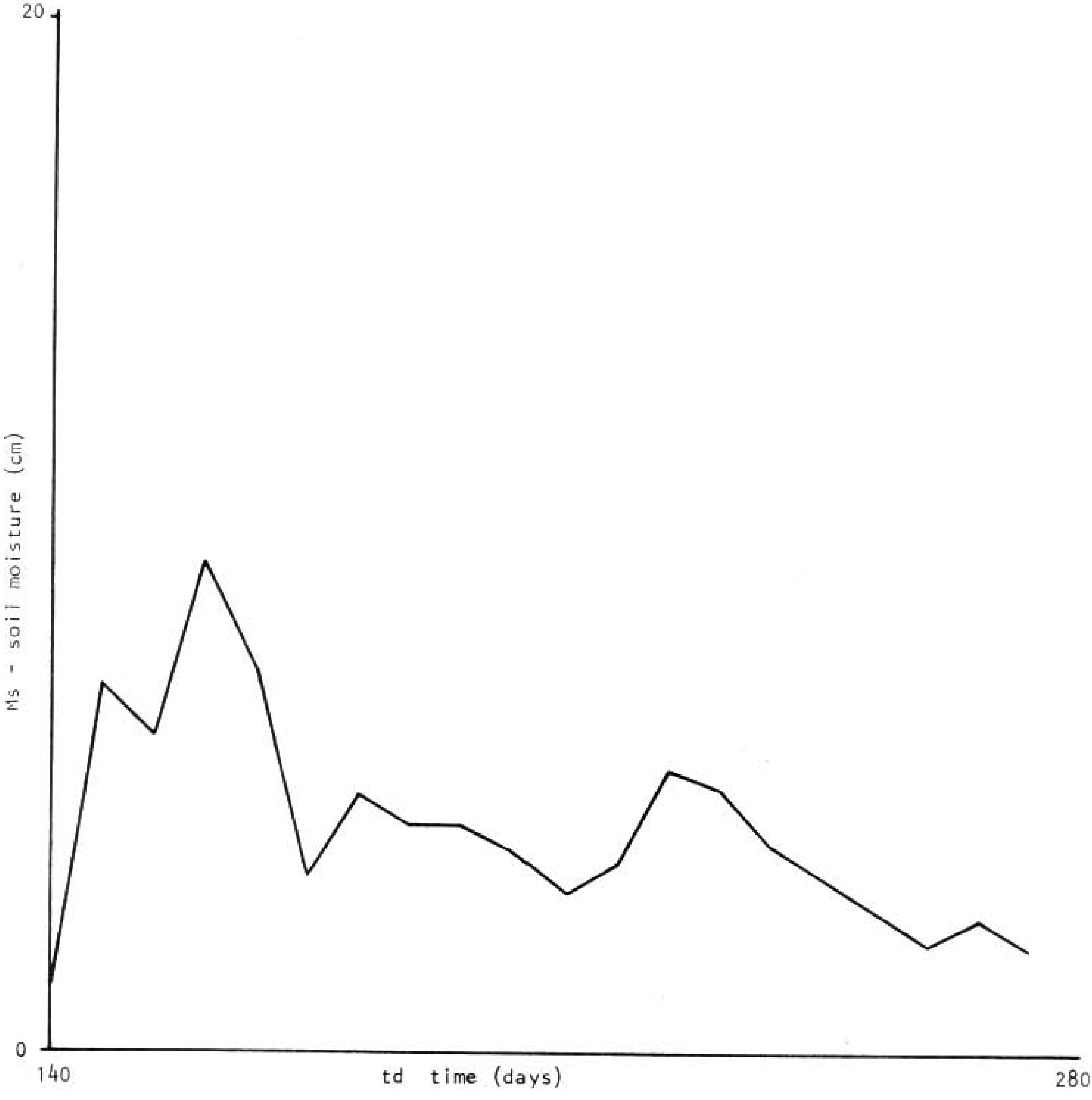


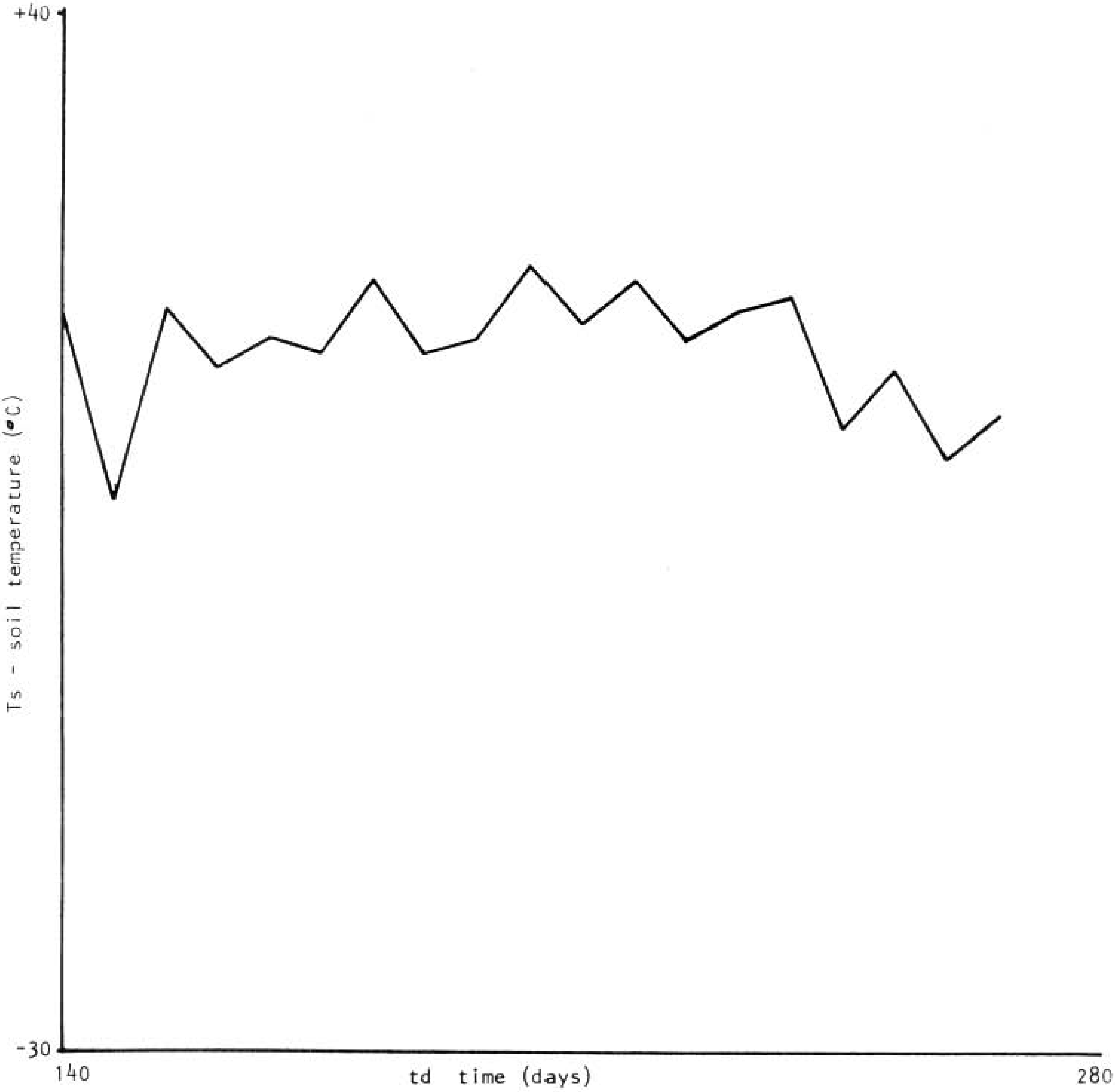
+40
-30
140
280
z_2, T_c - air temperature, soil surface temp. (°C)
td - time in days from Jan. 1

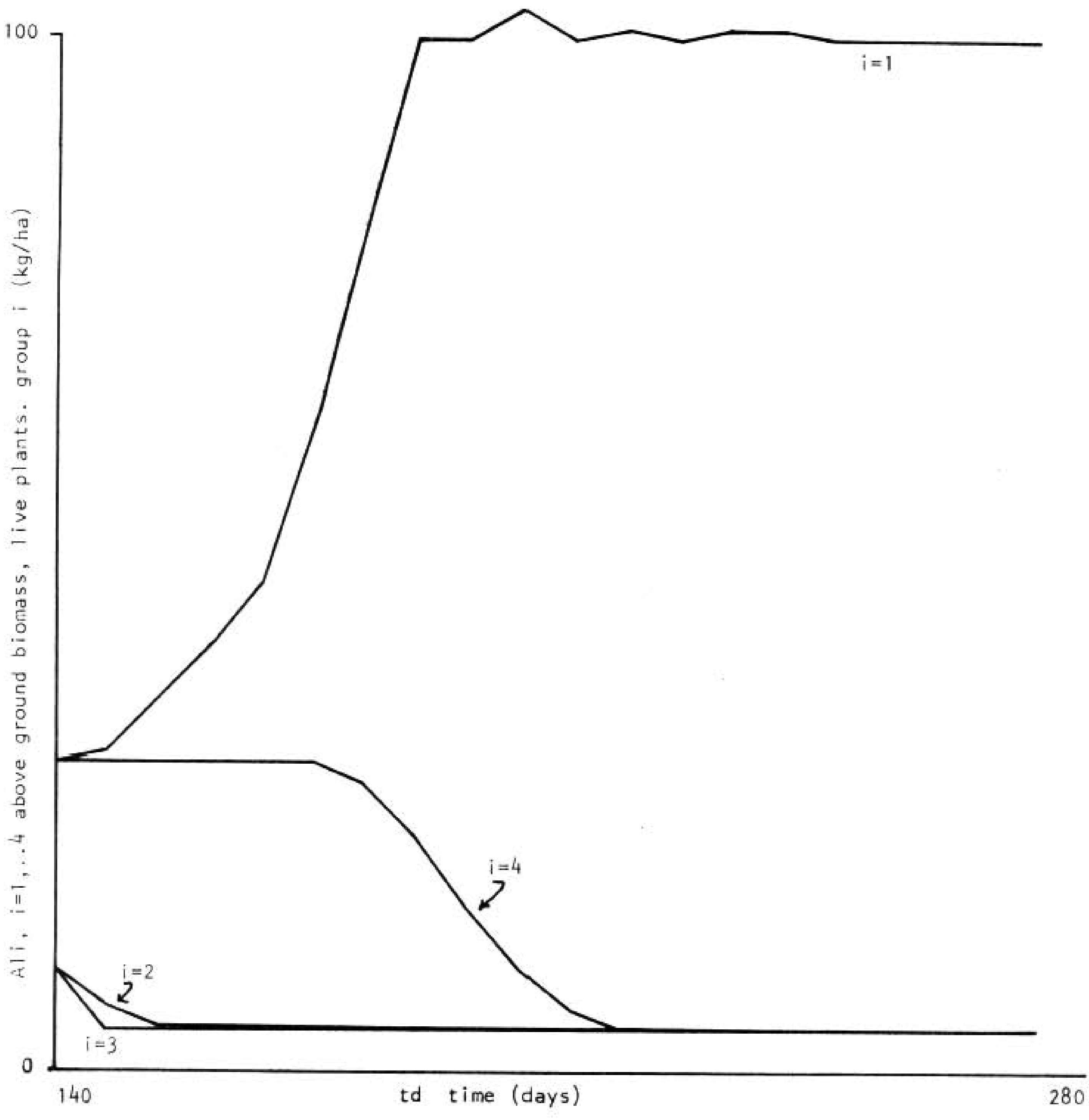
64

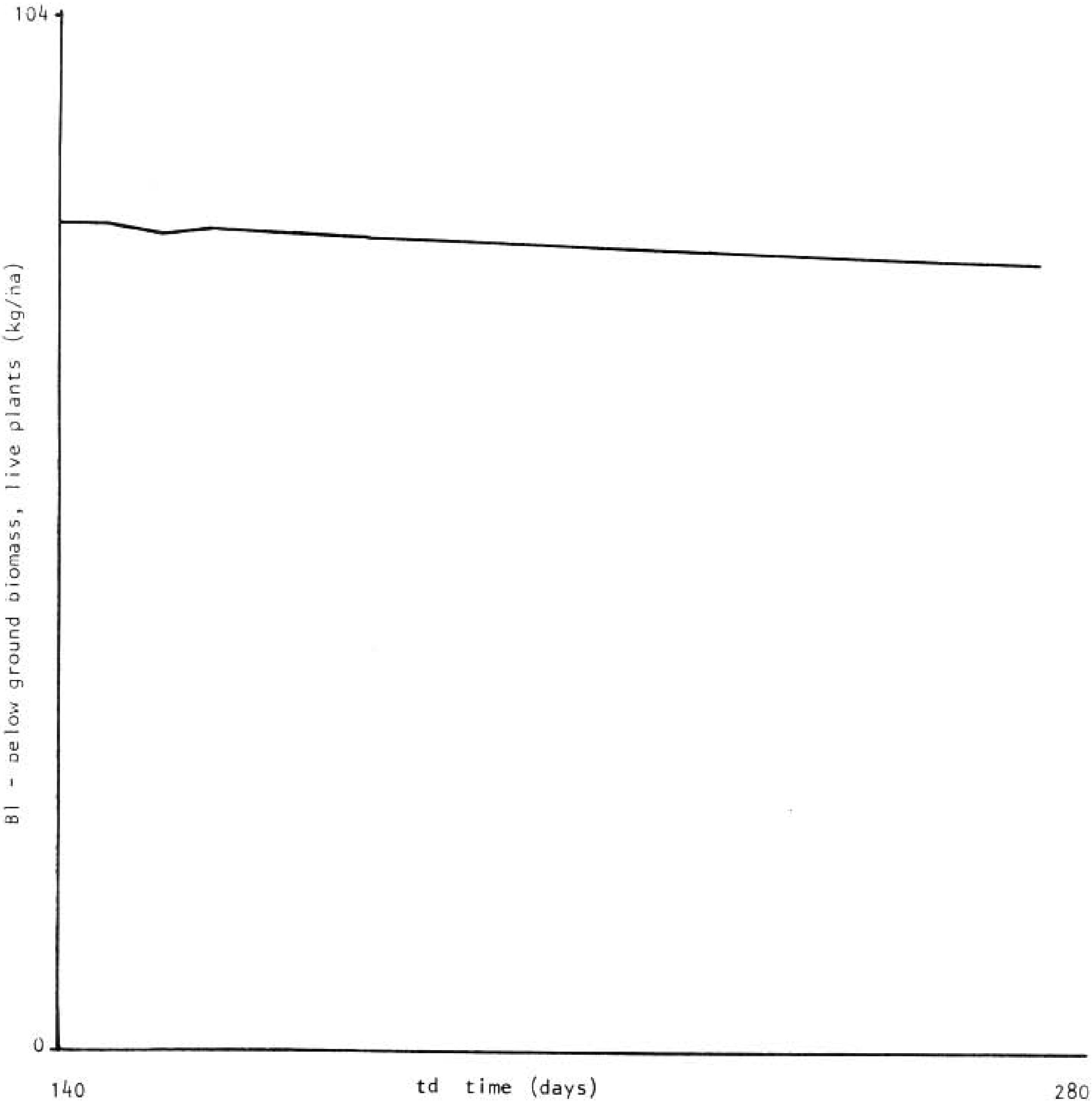

pz_{3i} daily precipitation (cm)
0
1
2
3
4
140
160
180
200
220
240
260
280
td

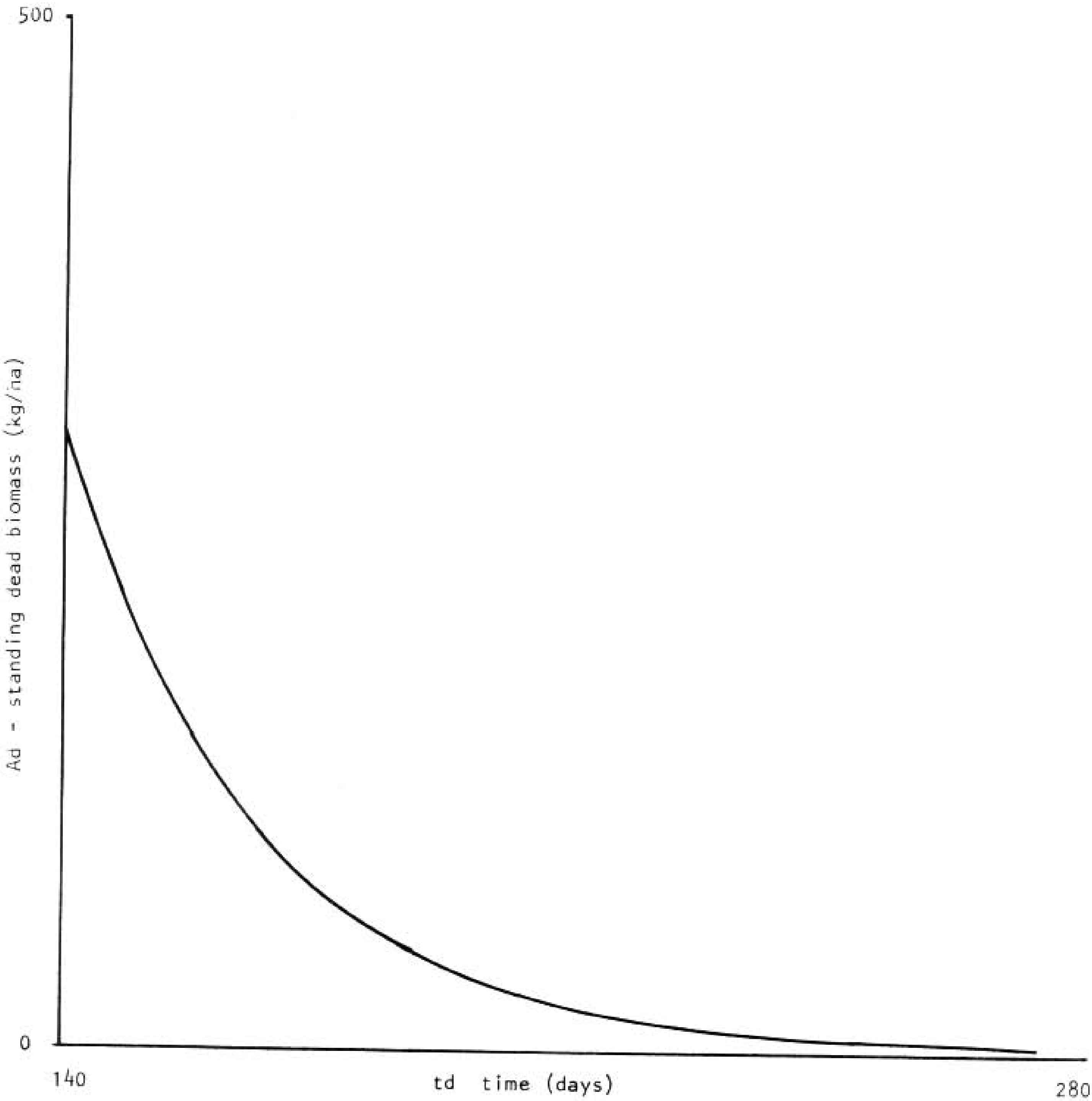

0
6
12
18
24
thm - time (hours, modulo 24)
0
100
200
300
400
500
z_4 wind speed (cm/sec)

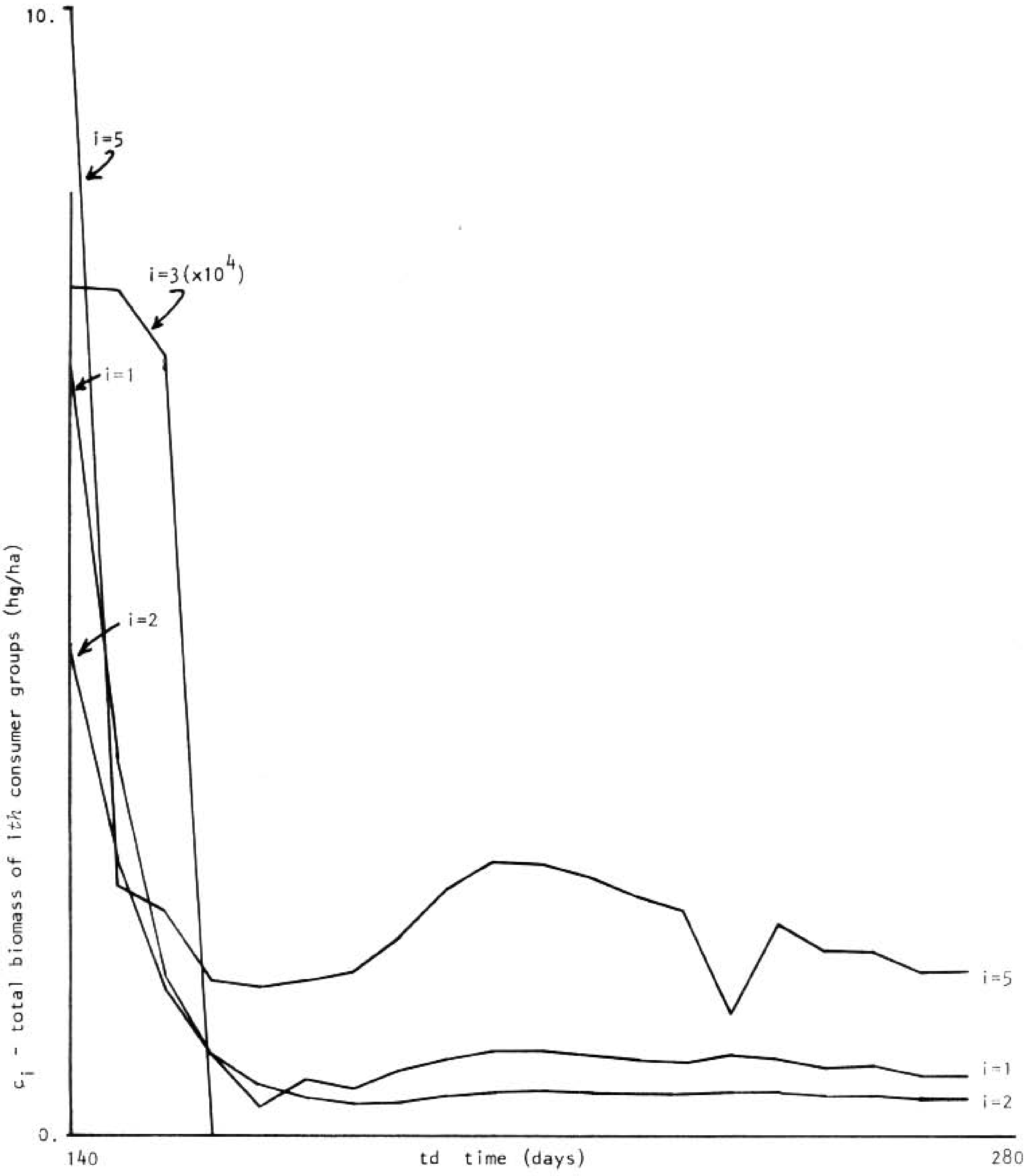

20
Ms - soil moisture (cm)
0
140
td time (days)
280

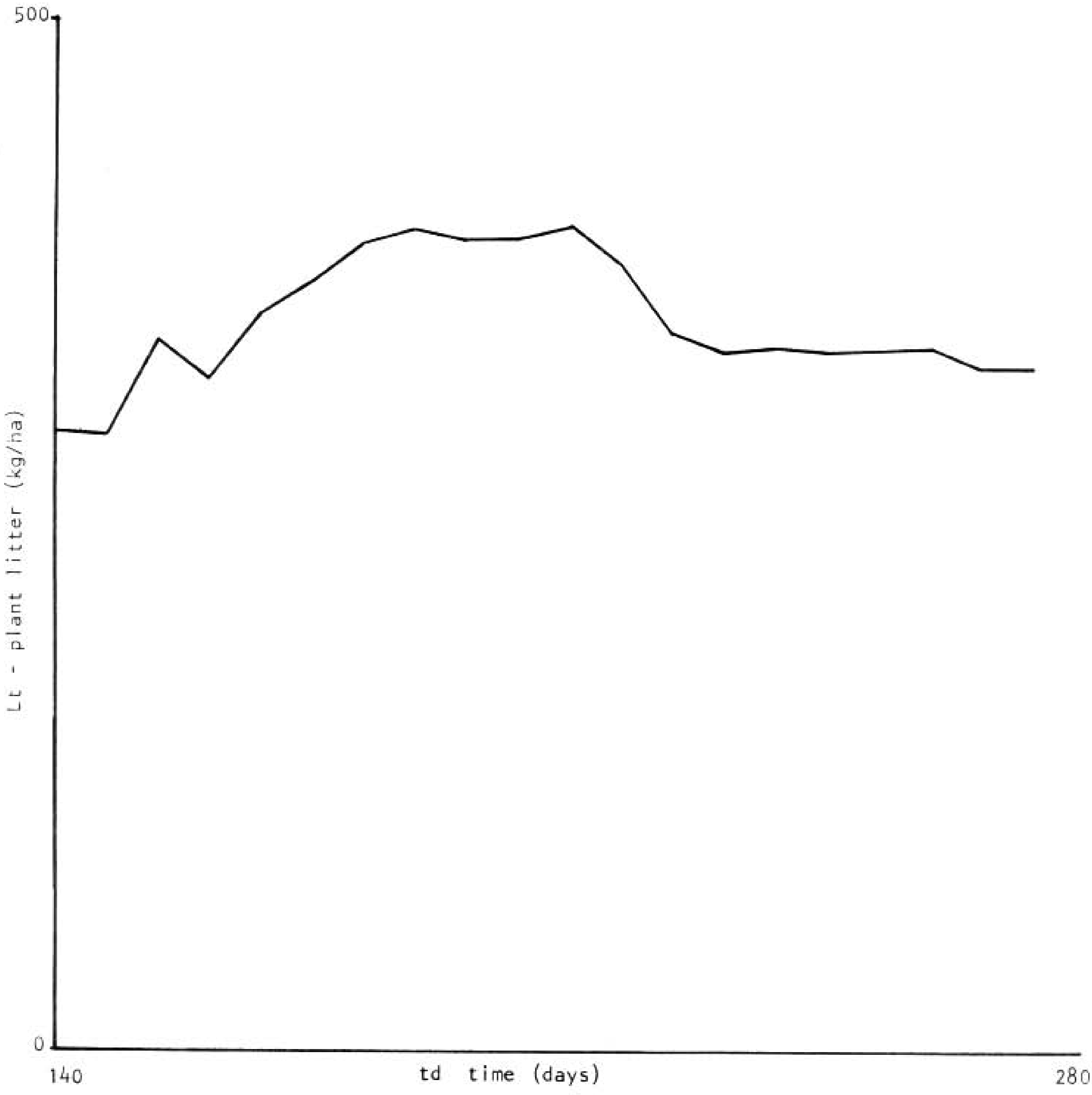

+40
-30
Ts - soil temperature (°C)
140
td time (days)
280

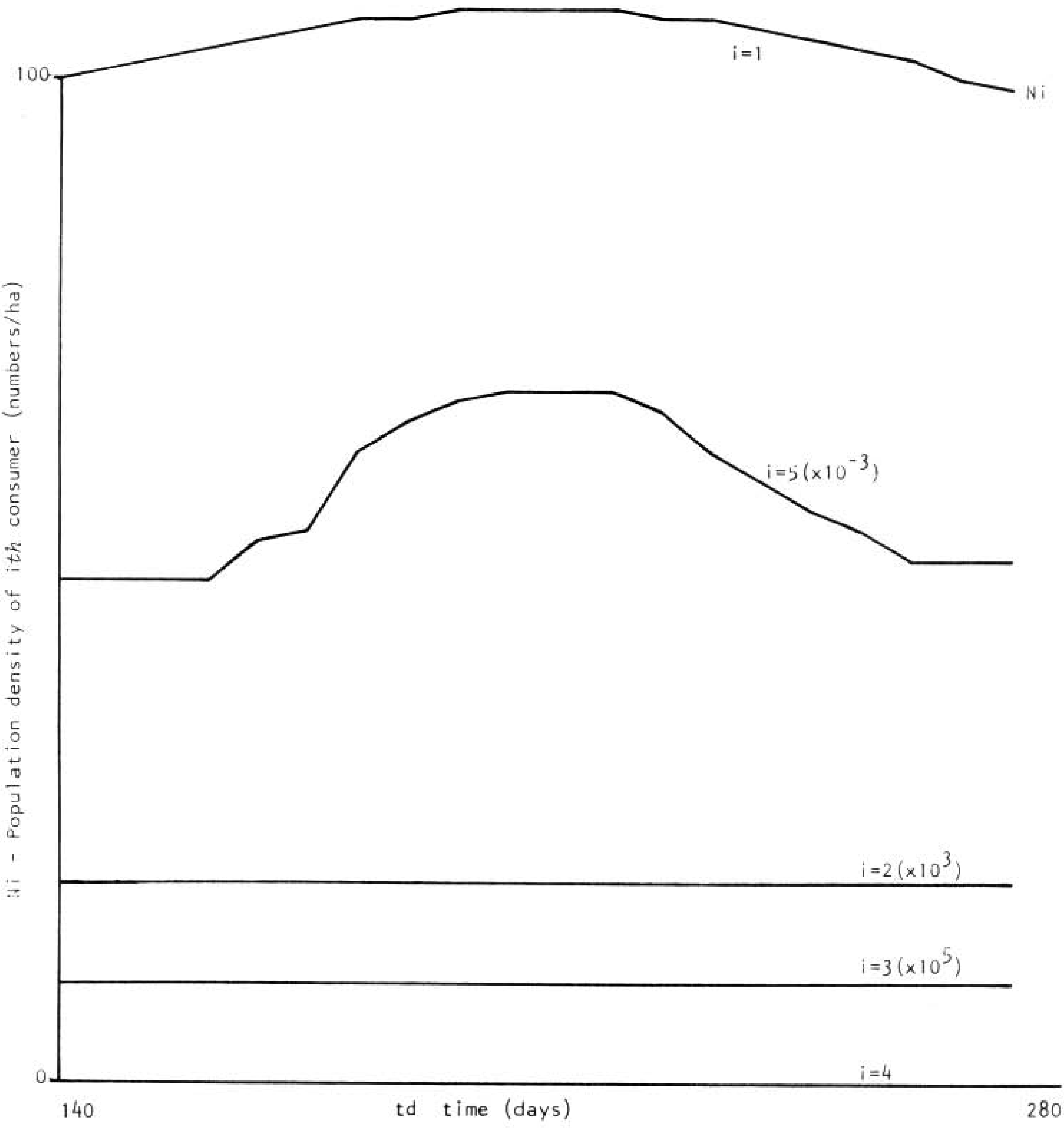

100
0
140
280
td time (days)
Ali, i=1,..4 above ground biomass, live plants. group i (kg/ha)
i=1
i=2
i=3
i=4

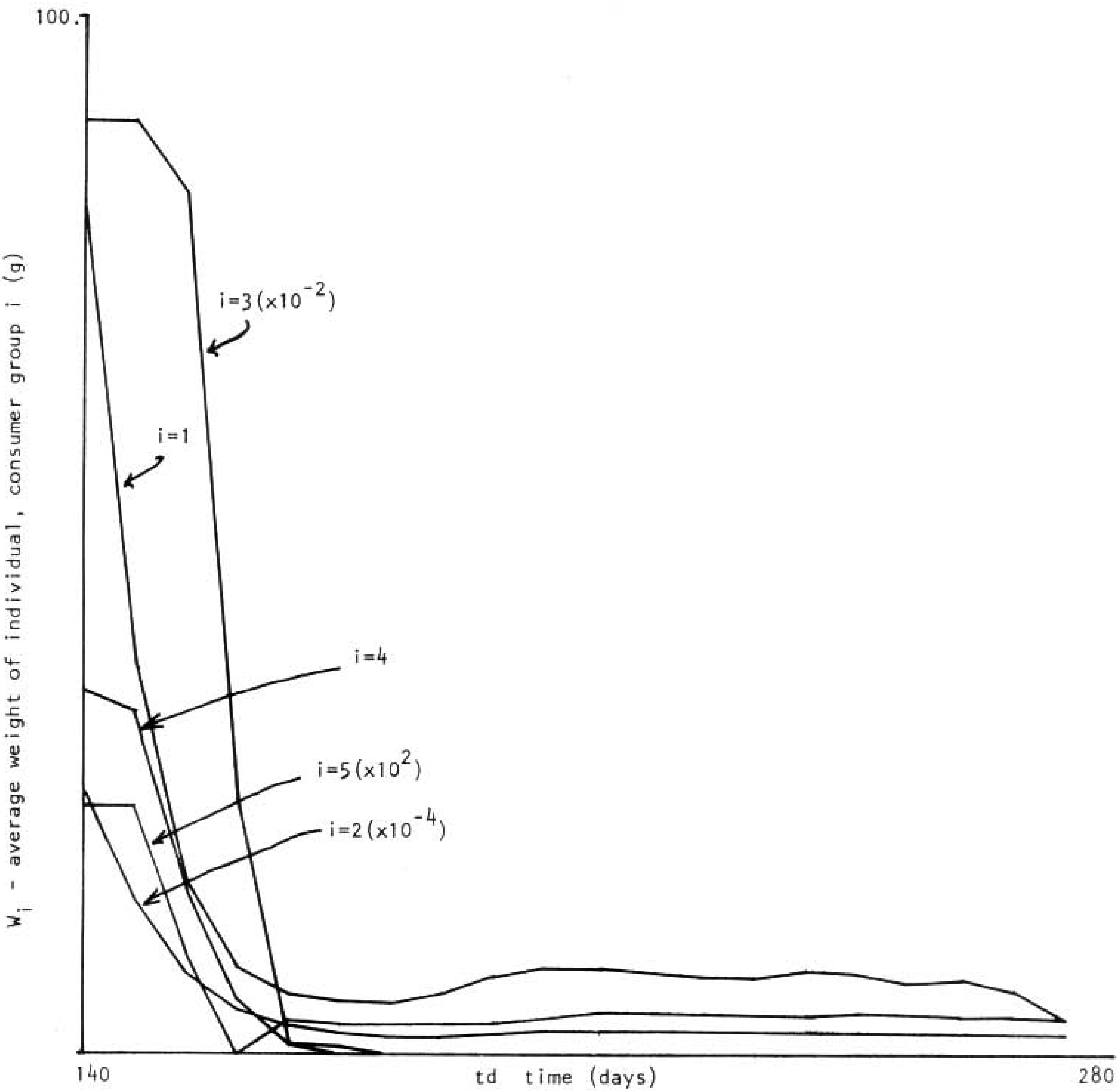

104
0
140
280
td time (days)
Bl - below ground biomass, live plants (kg/ha)

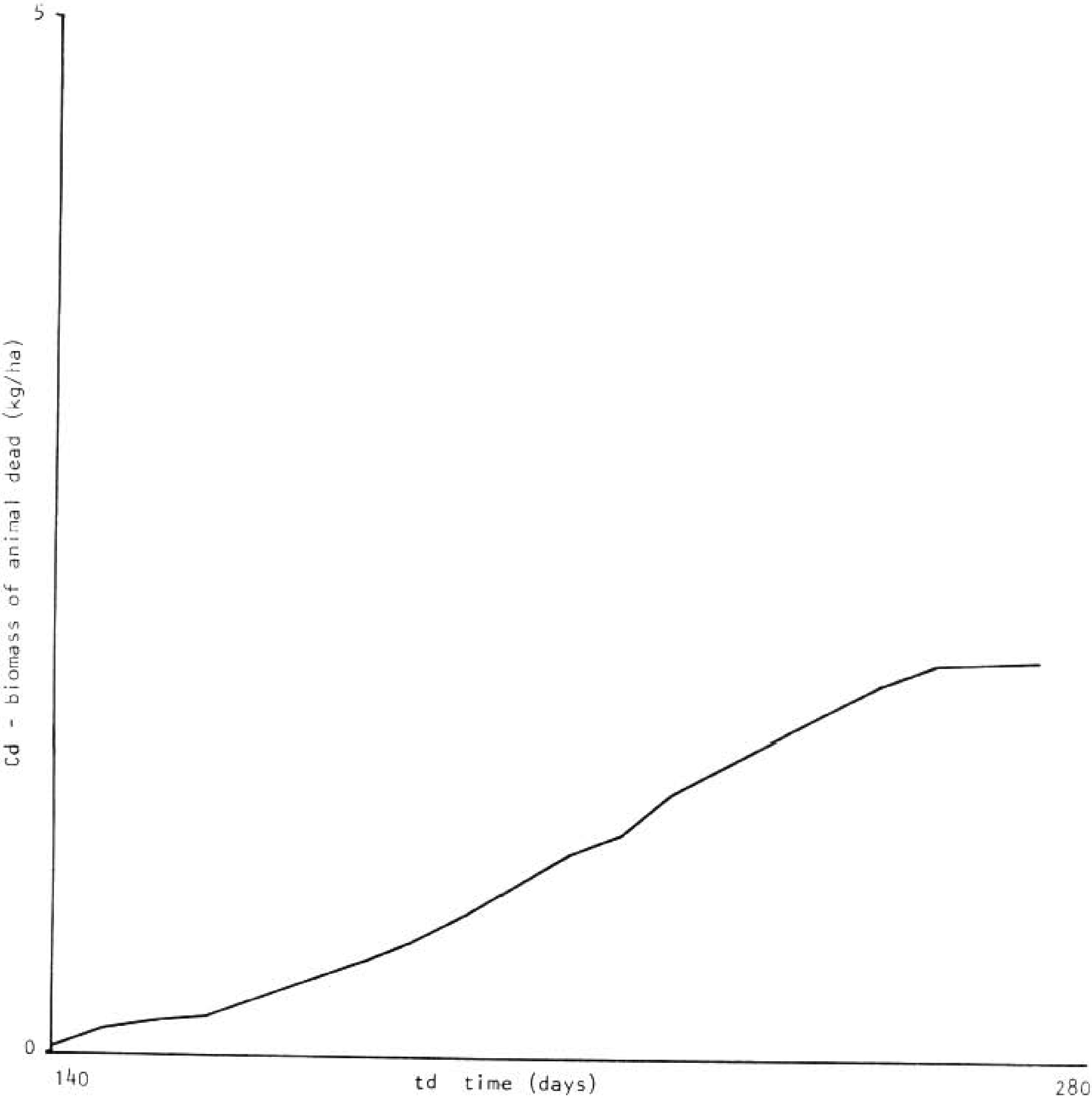

500
0
Ad - standing dead biomass (kg/ha)
140
td time (days)
280

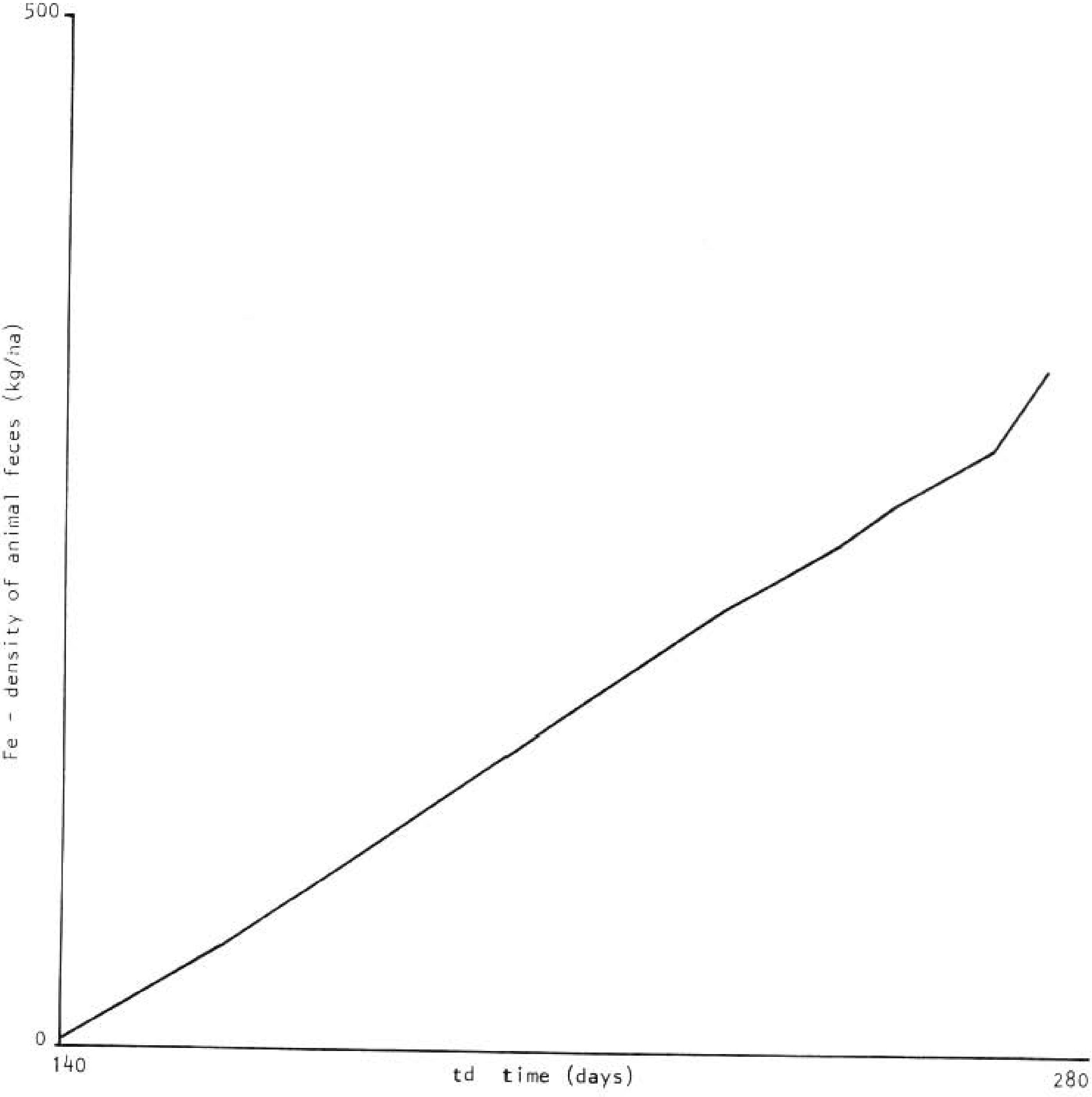

10.
0.
140
280
td time (days)
c_i - total biomass of ith consumer groups (hg/ha)
i=5
i=3(x10^4)
i=1
i=2
i=5
i=1
i=2


500
0
Lt - plant litter (kg/ha)
140
td time (days)
280


100
0
Ni
i=1
i=5 (x10^-3)
i=2 (x10^3)
i=3 (x10^5)
i=4
Ni - Population density of ith consumer (numbers/ha)
140
td time (days)
280


100.
W_i - average weight of individual, consumer group i (g)
i=3(x10^{-2})
i=1
i=4
i=5(x10^{2})
i=2(x10^{-4})
140
td time (days)
280


5
0
Cd - biomass of animal dead (kg/ha)
140
td time (days)
280


500
0
Fe - density of animal feces (kg/ha)
140
td time (days)
280