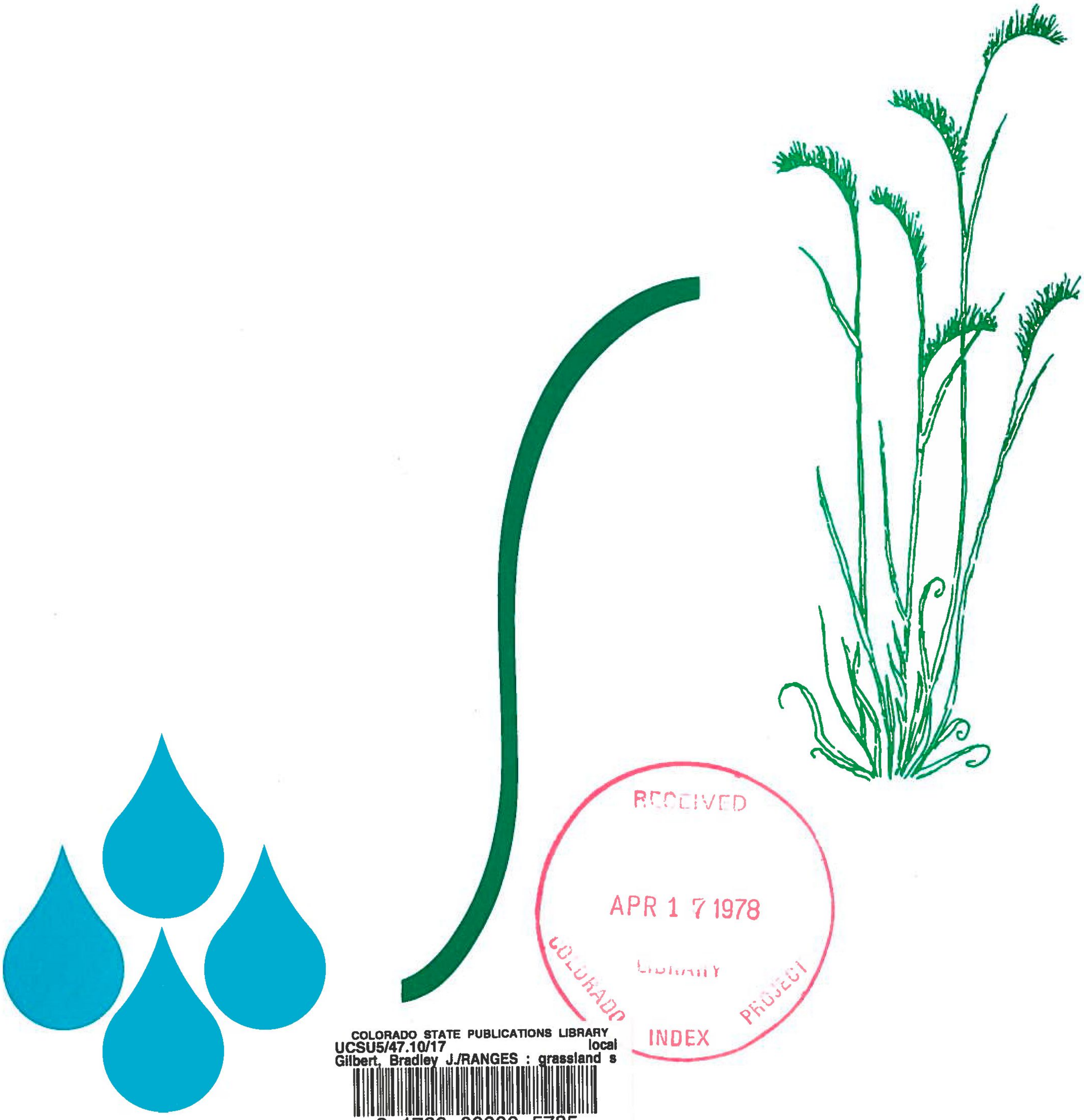
RANGES

Grassland Simulation Model
Bradley J. Gilbert

Science Series No. 17
August 1975



Range Science Department

Colorado State University

RANGES Grassland Simulation Model

Range Science Series No. 17

Bradley J. Gilbert

ABSTRACT

The purpose of the RANGES simulator is to link a rangeland ecosystem with the range management and range economic systems. With this goal in mind, the level of complexity of the model needs only to satisfy the input requirements of the management and economic models.

The model which meets these goals is composed of driving or exogenous variables, a soil water submodel, a producer section within a feedback loop containing a consumer section, and a market or economic section. The driving variables are mean daily temperature and daily precipitation either read from tapes of actual climatic data or generated hypothetically. The soil water submodel is primarily an evapotranspiration function based on soil characteristics such as wilting point and field capacity. The information from the soil water level coupled with the mean daily temperature is used to control plant growth. The forage consumed by livestock consists of green and dead plant material. The protein content of each forage component influences the computed livestock consumption rate, which in turn determines whether or not the animals are gaining weight. The market section of the model calculates an animal price vector of dependent normal random variables from an array of mean net prices and a variance-covariance matrix for net prices.

The plant growth response generated by the model is designed for information to management models such as forage standing crop and the variation of forage available to cattle. For example, the simulation model can be used interactively with a dynamic optimization model, supplying forage response to different levels of grazing intensity and climatic fluctuations. The simulation also supplies cattle weight gains for management purposes, and allows testing of management grazing regimes on simulated forage.

ACKNOWLEDGMENTS

Much of the original coding of this model was done by George S. Innis, Donald A. Jameson, and Richard Miskimins using modeling techniques programmed by Jon Gustafson. Data for various process expressions were provided by John Nunn, Richard Rice, M. J. Trlica, and Warren Whitman.

TABLE OF CONTENTS

INTRODUCTION

The western livestock grazing industry is very dependent upon certain biotic and abiotic processes occurring in the environment. Variations in the temperature and precipitation regime of an area cause unpredictable fluctuations in the resource response. However, given the source of the variation and the processes influencing resources, some questions concerning efficiency of grazing can be answered.

The study of ecological response to abiotic variables (variables that perturb the biotic system and cause variation) can be undertaken in several ways. The first, and most frequently utilized, is the measurement of ecological parameters in a field or laboratory study. This method attempts to reduce variation in an experiment by controlling certain physical parameters which affect the response of the dependent variable. Controlled experiments, however, usually entail removing the measured variable from its natural state.

A contemporary method, which relies on knowledge of the natural behavior of the system, involves adapting empirical information into a mathematical model. The result is a system which can be experimented with, while maintaining the interaction of the components. Field and laboratory experiments become costly and unmanageable quickly when dealing with a complex system. On the other hand, once a mathematical model has been developed, a multitude of experiments can be performed inexpensively. Although mathematical models are always less complete than the natural systems, they are capable of supplying information when only low resolution information is required to solve a problem.

Ecological data of the type generated by the model is of primary importance to grassland management. Given stochastic driving variables, plant production can be generated and the statistical properties of the resulting data can be used in management models.

A series of grassland ecological models were developed under the title RANGES. The objectives were to use available information regarding the growth of herbage in a grassland ecosystem as a function of the major driving variables which are taken to be temperature and rainfall. Forage response, together with the known response of various classes of domestic ruminants can then be used to determine animal weight gains for different stocking strategies and for different supplemental feeding strategies. The model can be used to investigate various marketing strategies and the effects of this entire collection of management practices on income.

These objectives delineate certain biotic, abiotic, and economic systems as being necessary in a model which addresses the objectives. A plant growth system is needed because of its influence on the livestock enterprise. With these data and acknowledging the natural variation inherent in herbage growth, the second system requirement becomes evident. Plant growth is determined by solar radiation, soil temperature, CO_2 availability, soil water and phenological stage; but air temperature and rainfall are assumed to be the most important driving variables.

The soil water system is the link between the abiotic and biotic systems. Although there is a strong correlation between rainfall and plant growth, there are important processes which limit the quantity of water available for plant growth and determine the persistence of available water in the soil layer.

The final model requirement, "investigating market strategies", necessitates information about market behavior, including random fluctuations which might be expected. The economic system also requires knowledge of costs involved in the enterprise to determine when marketing might be most profitable. Generality and simplicity are also necessary attributes of the model if it is to be used on varied grassland sites with minimal data input from a specific site.

CONCEPTS

Simulation techniques. There are two major schools of thought in mathematical modeling of dynamic systems. One revolves around systems of differential equations and their exact solutions, and the other is oreinted toward simulating the solution with systems of differential or difference equations. There are several practical and philosophical reasons for choosing simulation as the method for meeting the objectives of the RANGES model. The first and foremost reason is the inability of analytic solution techniques to solve the system of equations which addresses the objectives of the model (Forrester, 1968). The conceptual difficulty in using instantaneous rates when they are physically unmeasurable (as $\lim_{\Delta t \to 0}$) also lends credibility to the use of simulation techniques (Innis, 1972). Systems are often viewed in difference terms; i.e. the flow from one state to another is measured over a finite time interval, which makes a scheme using difference equations more credible (Innis, 1972).

A system of difference equations is the basis of this model using an initial value solution technique. Thus, each successive state of the system is determined from the previous one, knowing only the flow definitions and the level of the state variable at the previous time step. The general form of the solution scheme is:

$$X(t + \Delta t) = X(t) + \Delta t * (\Sigma F_i - \Sigma F_o) \qquad (1)$$

where $X(t)$ is the amount of material in, or the level of, the state variable X at time t, t is the current simulated time, Δt is the time step or increment of simulated time, ΣF_i is the sum of the flows into state variable X and ΣF_o is the sum of the flows out of state variable X.

Equation 1 states that the level of X after Δt amount of time has elapsed will be the level of X at the previous time plus the sum of the flows into and out of X multiplied by the time step. Thus, interaction in the system can be easily represented by having the flows into or out of one level be dependent on another level. This conceptual framework was retained for all versions of the RANGES models whether they were coded in SIMCOMP or FORTRAN.

The model structure is composed of five principal parts: (i) abiotic driving variables, (ii) soil water submodel, (iii) producer section, (iv) consumer section, and (v) economic section (Fig. 1).

Driving Variables

Driving variables are those that are independent of the simulated system, but are necessary to initiate response by the system within some arbitrary boundary which is being modeled. In RANGES, the driving variables are daily precipitation and mean daily temperature. These variables are not mechanistically described in the model because they are not considered part of the feedback with the biotic subsystem.

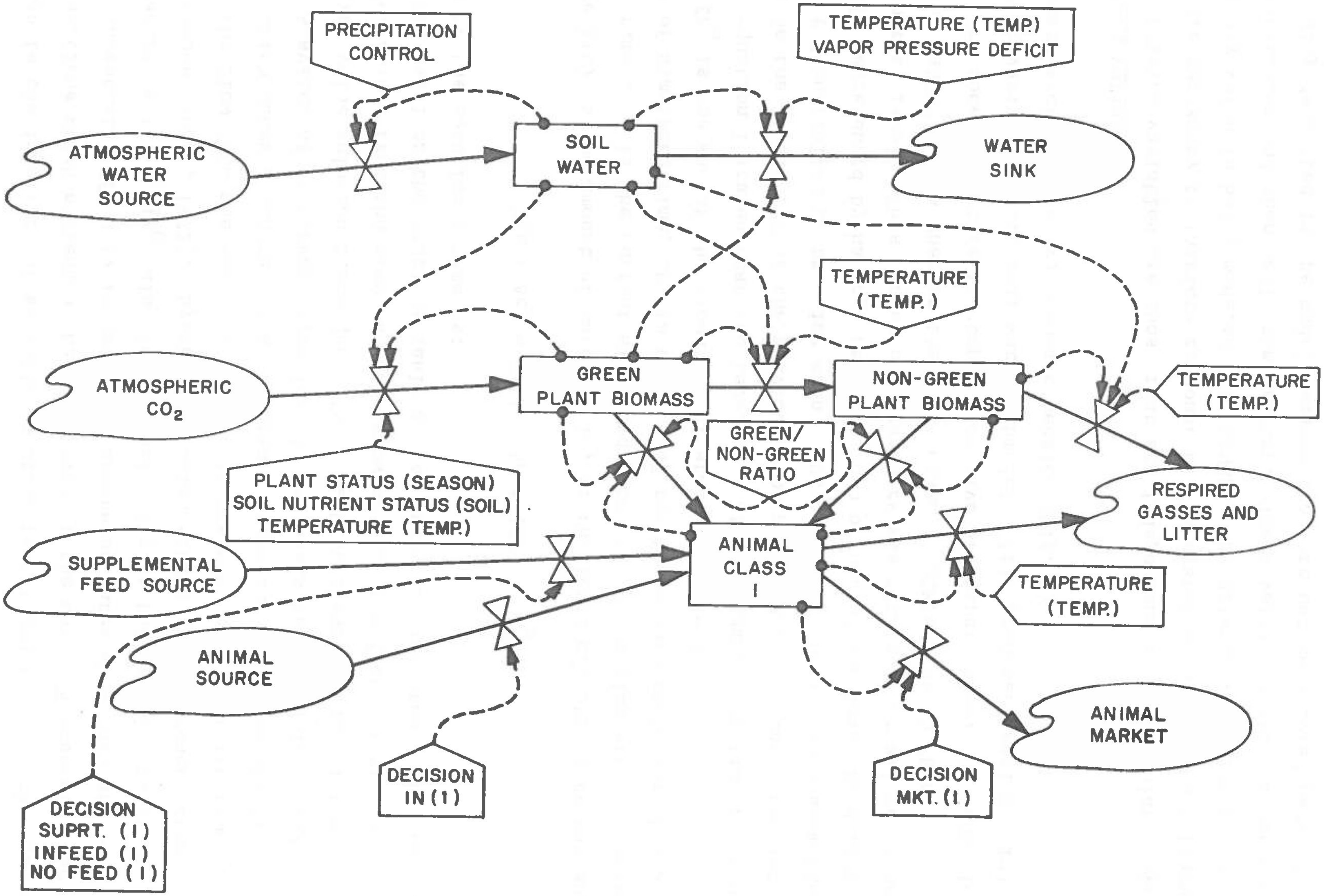


Fig. 1. Diagram for the RANGES grassland simulation model.

Daily precipitation can be obtained in three ways for use in RANGES: (1) from historical weather records read in from magnetic tape, (2) from a sine function representation of average historical data, or (3) from a stochastic precipitation generator. For debugging purposes, average historical precipitation data can be represented by a sine function with appropriate phase and amplitude shifts, such as the following equation for the Pawnee Grasslands (Fig. 2):

$$\text{Daily Rainfall (inches)} = (14.*(\text{SIN}((\text{IDAY}-60)/365.* 2\pi) + 1.))/365. \quad (2)$$

where IDAY is the Julian day.

The average daily temperature has two possible forms for entry into the model: (1) historical data from magnetic tape, or (2) as a sine function representation of historical averages, such as the following for the Pawnee Grasslands (Fig. 3):

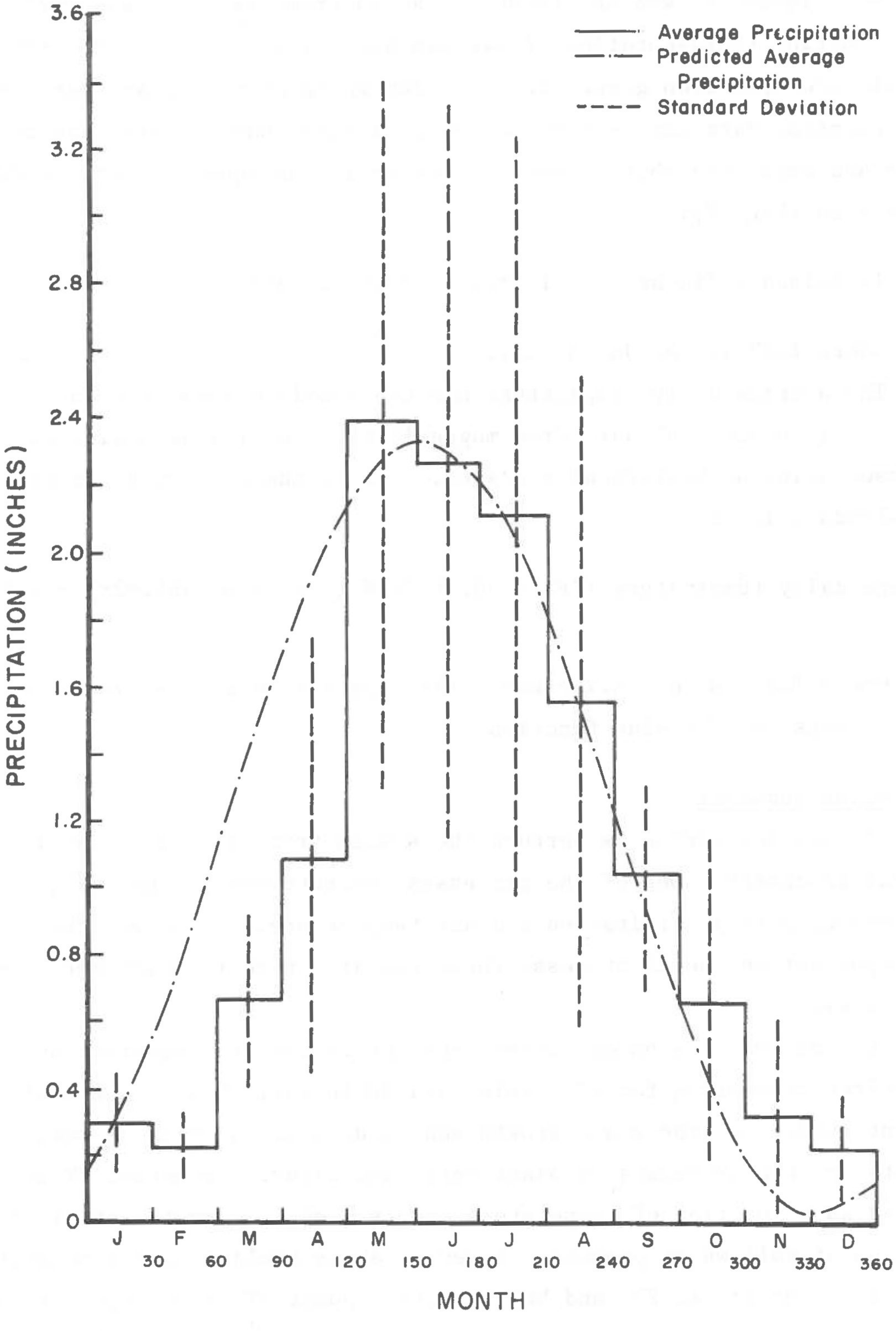
$$\text{Average daily temperature (°F)} = 30. * (\text{SIN}((\text{IDAY}-120)/365.*2\pi) + 1.0) + 22.0 \quad (3)$$

where IDAY is the Julian Day. (See Appendix 6 for derivation of coefficients for the sine function.)

Soil Water Submodel

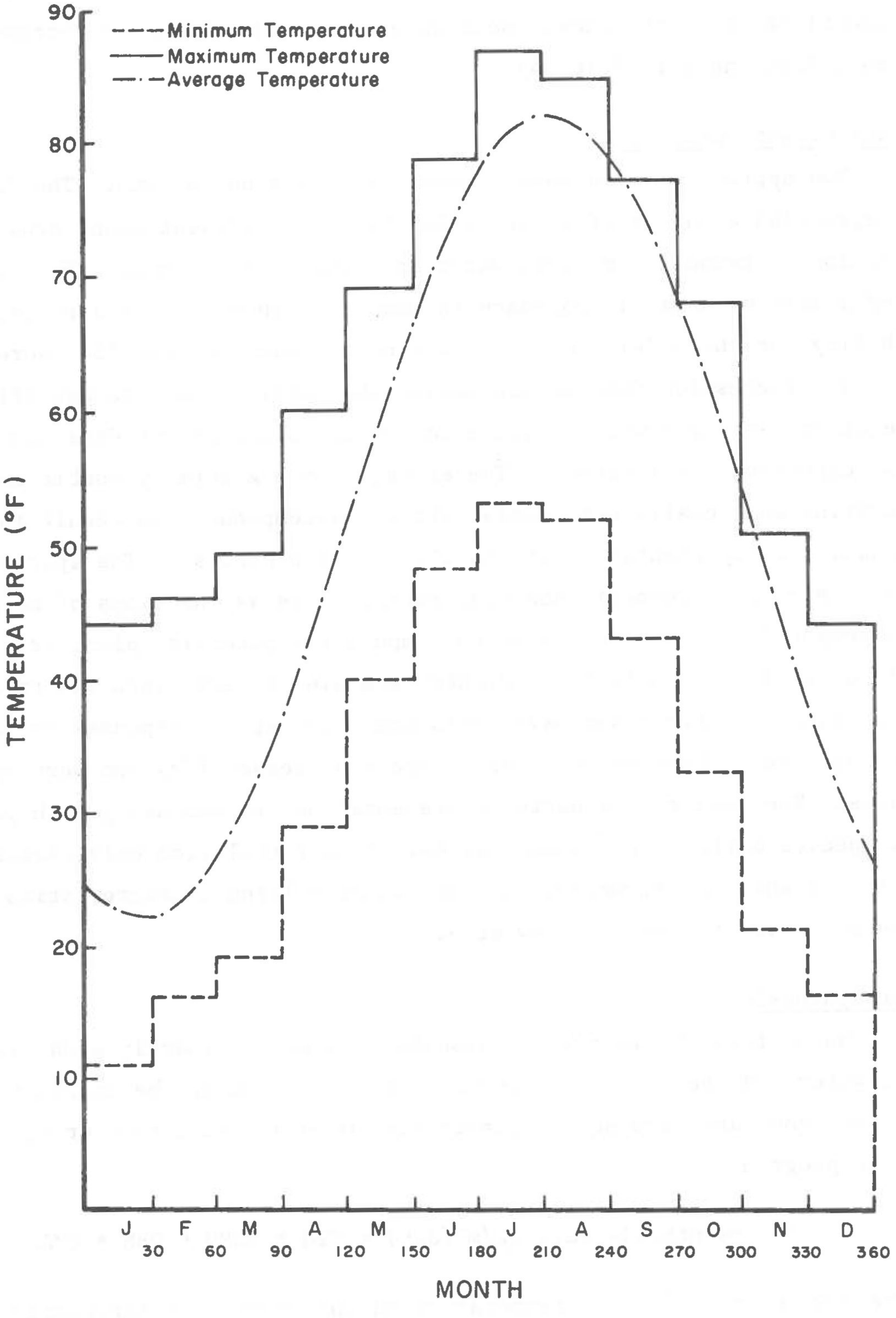
The driving variables perturb the system from its initial state via several processes. One of the processes, evapotranspiration (ET), is influenced by both precipitation and air temperature. Although other factors are important in the ET process, these two give results which meet the model objectives.

The soil water submodel takes temperature and precipitation as inputs and, after accounting for ET, yields available soil water. The soil water present influences the plant growth and is used as an index to moisture effects on such processes as plant death and decay. Potential ET is calculated as a function of temperature and is used to compute actual ET as a function of soil water present. However, above field capacity potential ET is used as the actual ET, and below wilting point ET is set equal to one-hundredth of ET at wilting point, i.e. there is no more transpiration. When



PPT = 14.* (SIN((IDAY - 60)/365.* 2π) + 1)/12.

Fig. 2. Predicted monthly precipitation amounts using a sine function for the Pawnee National Grasslands. Averages and standard deviations are from Rasmussen et al., 1971.



Temperature = 30. * (SIN ((I DAY −120)/365. * 2π) + 1.0) + 22.0

Fig. 3. Predicted daily temperature values using a sine function for the Pawnee National Grasslands. Maximum and minimum temperatures are from Rasmussen et al., 1971.

soil water exceeds total soil water holding capacity, then ET is increased proportionally to the excess soil water, which is a ploy to represent water runoff from the soil (Fig. 4).

Plant Growth Submodel

Two approaches were used in dealing with plant growth. The first used a regression equation of a particular form to represent plant growth as a function of temperature, soil water and live plant biomass. The second method used a more mechanistic approach in hopes of achieving a wider range of applicability for the model. Only the second approach is described here.

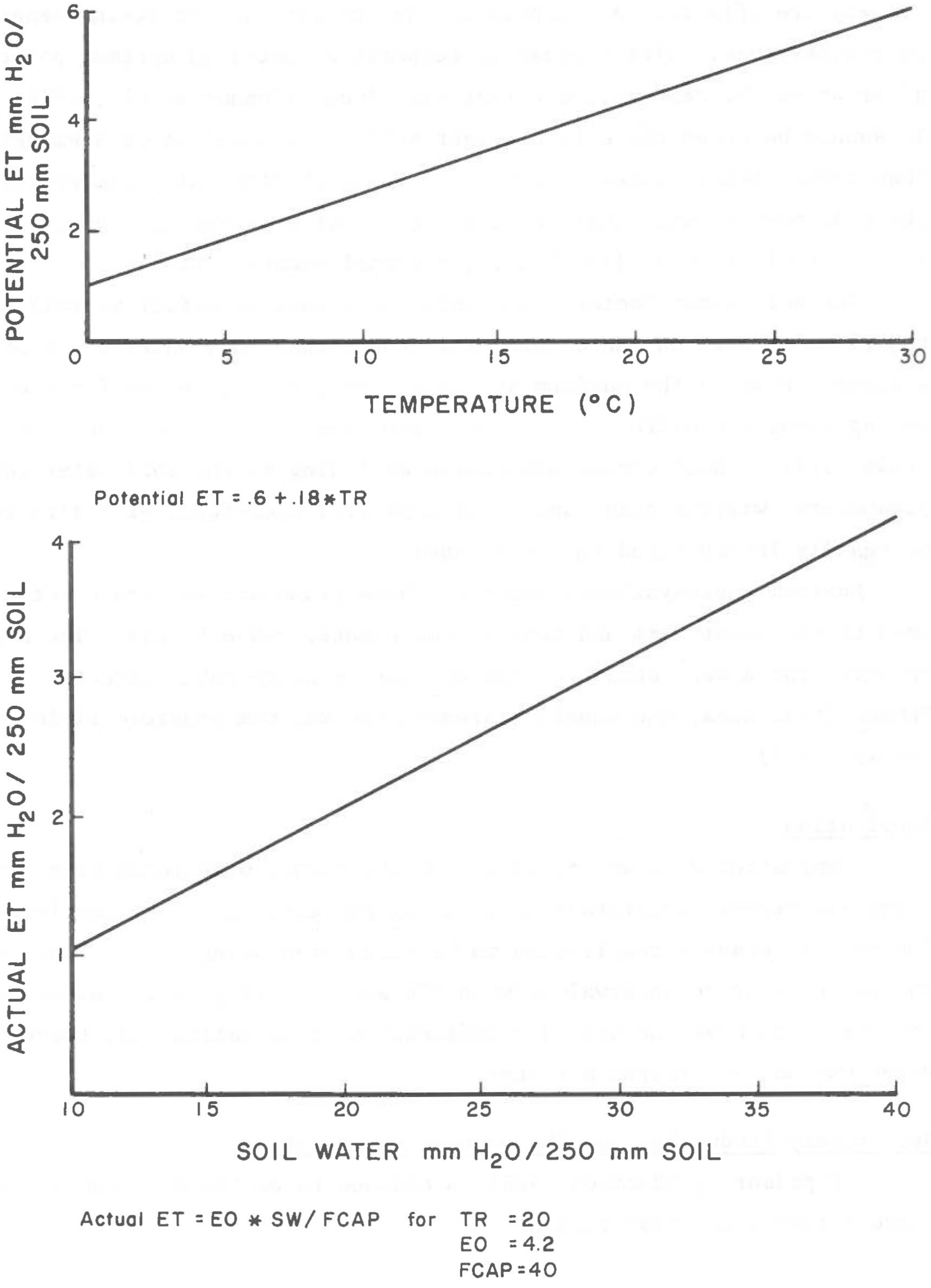
The regression formulation caused the model to be site specific. Its use at another site would require collecting plant growth data and determining new regression coefficients. Therefore, a more generally usable formulation requiring only easily obtainable data was attempted. The result is a more mechanistic representation of the plant growth process. The approach is to take basic photosynthesis and respiration rates as functions of soil water and temperature, and from these to compute the potential plant production per unit of live plant material. Plants were also divided into two categories that exhibit distinct photosynthesis and respiration responses to temperature and moisture. These two categories are cool season (C3) and warm season (C4) plants. The result is a better representation of seasonal growth patterns and species differences between sites. This formulation only requires information on abiotic parameters and soil water holding characteristics as input when the model is run on a new site.

Photosynthesis

The photosynthesis rate is assumed to depend primarily upon available soil water and average daily temperature. The form of the equation is the same for both cool and warm season plants with Q being equal to C or W, respectively in the program.

$$\text{Photosynthesis rate } (g/m^2/day) = ETQ * ESMQ * QBM * QMAX \qquad (4)$$

where ETQ is the effect of temperature on photosynthesis (proportion); ESMQ is the effect of soil water on photosynthesis (proportion); QBM is aboveground green plant biomass (g/m^2); and QMAX is the maximum photosynthetic rate (grams photosynthate/gram live plant biomass/day).



Potential ET = .6 + .18*TR

Actual ET = EO * SW/FCAP for TR = 20
EO = 4.2
FCAP = 40

Fig. 4. (a) Potential ET calculated as a function of temperature.

(b) Actual ET calculated from potential ET and the amount of water present in the soil.

The temperature term is intended to embody radiation effects, as well as temperature effects. A parabola is used to give an increasing beneficial effect on photosynthesis with increasing temperature until an optimum point is reached, after which the temperature effect diminishes (Connor et al., 1974) (Fig. 5). It should be noted there is a slight shift in the curves to seemingly lower than normal temperatures, a result of using average daily temperature. Also, there is new evidence that western wheatgrass requires only very low temperatures for growth initiation (Joe Trlica, personal communication).

The soil water factor represents the impact of effective soil water in the top 250mm of soil on photosynthesis. Additional soil water increases the photosynthesis rate to the maximum at field capacity. Different functions were used to represent the differences between cool and warm season plants reported by Brown (1974). Both curves are scaled according to the soil-water relation parameters, wilting point and field capacity; therefore, site differences can be readily incorporated into the model.

Maximum photosynthesis rates for blue grama and western wheatgrass were used to represent warm and cool season plants, respectively. The highest reported rates were obtained from CO_2 gas exchange data collected at the Pawnee Grasslands, and should represent the maximum possible field rate (Brown, 1974).

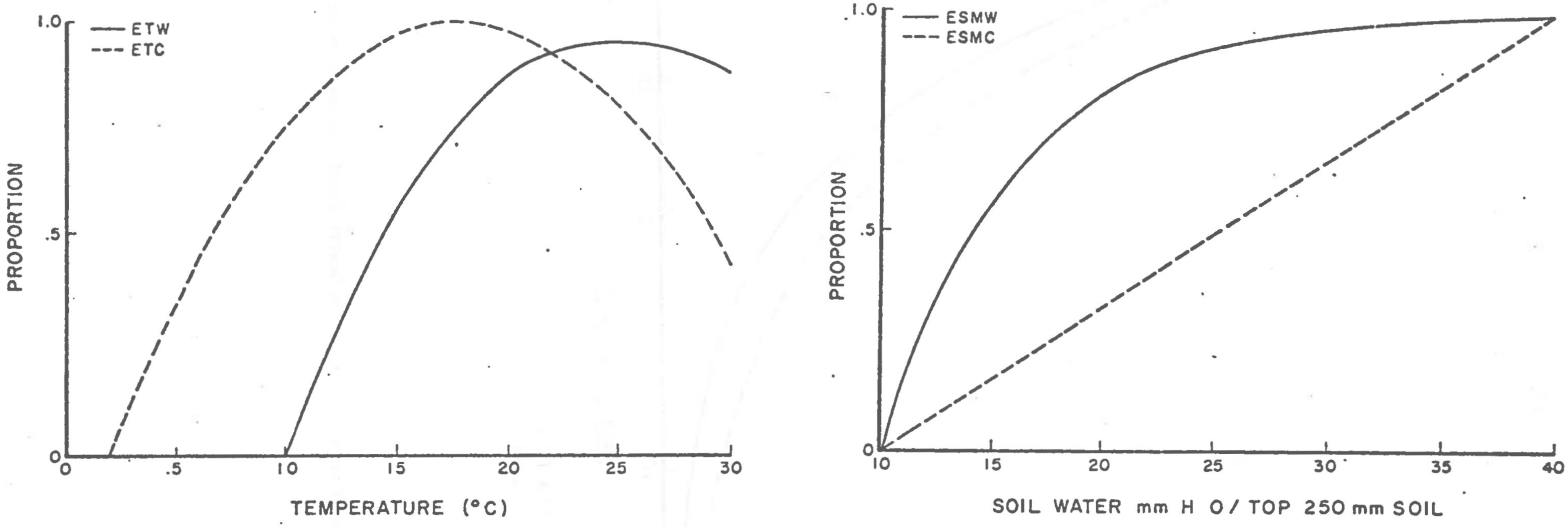
Respiration

Respiration data was collected on the Pawnee Grasslands site for blue grama and western wheatgrass using a CO_2 gas exchange mechanism (Brown, 1974). The results enabled respiration to be calculated using an exponential function on the temperature interval between 0°C and 30°C (Fig. 6). Temperature was considered to have the dominant influence on respiration and, therefore, soil water was not considered a factor.

Net Primary Production and Aboveground Accumulation

Net primary production (NPP) is assumed to be the difference between photosynthesis and respiration:

$$NPP = P_s - R_s \tag{5}$$

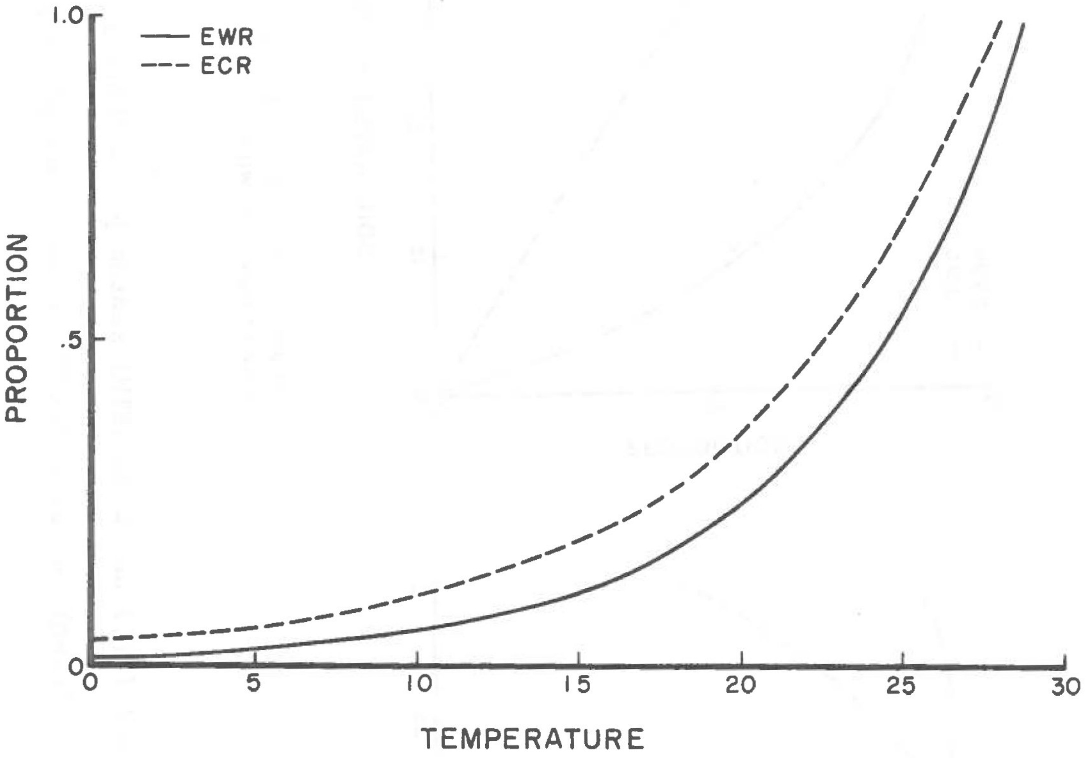


ETW = 1. -(TR-25.) ** 2 / (15.*15.)
ETC = 1. -(TR-18.) ** 2 / (16.*16.)

ESMW = 1.- EXP (-QW * (SW - WILT)) where, QW = 5./(FCAP-WILT)
ESMC = QC * (SW-WILT)
QC = 1./(FCAP-WILT)
FCAP = 40
WILT = 10

Fig. 5. (a) Effect of temperature on cool (ETC) and warm (ETW) season plant photosynthesis.

(b) Effect of soil water on cool (ESMC) and warm (ESMW) season plant photosynthesis.



EWR = .01 * EXP (.161 * TR)

ECR = .03 * EXP (.126 * TR)

Fig. 6. Effect of temperature on cool (ECR) and warm (EWR) season plant respiration.

All of the carbon fixed by the plant shoots is not retained aboveground, however. Large quantities of the photoassimilate are translocated belowground to crowns and roots. There is evidence to suggest that as much as 60 to 88% of the photoassimilate is translocated (Singh and Coleman, 1974). Although seasonal variation undoubtedly exists as a function of plant phenology, the lumping of plant species in this model removes the importance of the variation. Therefore, a value of 70% was used to represent translocation.

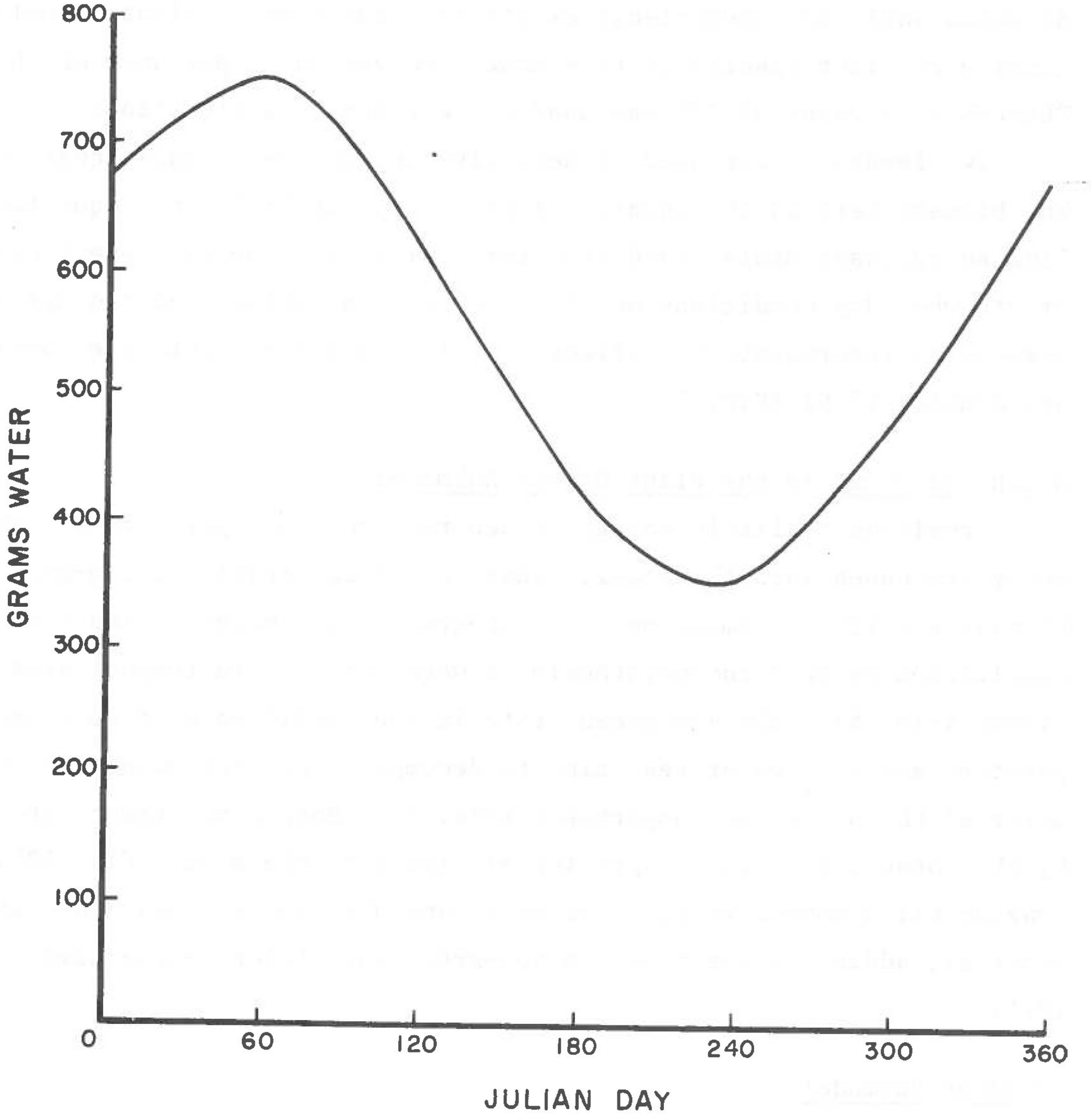
Two feedbacks are used to help give the photosynthesis equation stability. The biomass term in the equation itself helps stabilize the equation, and limited photosynthesis based on water availability helps prevent unreasonable growth when dry conditions occur. The grams of water required are varied over seasons to incorporate the effects of plant phenology into the model (McGinnies and Arnold, 1939) (Fig. 7).

Death and Decay in the Plant Growth Submodel

Providing realistic forage values requires incorporation of death and decay processes into the model. These processes result in non-green plant biomass and litter compartments. Non-green plant material can accumulate when respiration exceeds photosynthesis or when hot or cold temperatures kill plant tissue (Fig. 8). The non-green state is then modified as a function of temperature and soil water resulting in decomposition, represented by the flow of material to the litter compartment (Fig. 9). Both compartments are influenced by the intensity of cattle grazing simulated in the model (Fig. 10). Heavy grazing kills some live plant material and also knocks down non-green plant material, adding to the flows to non-green and litter, respectively (Whitman, 1974).

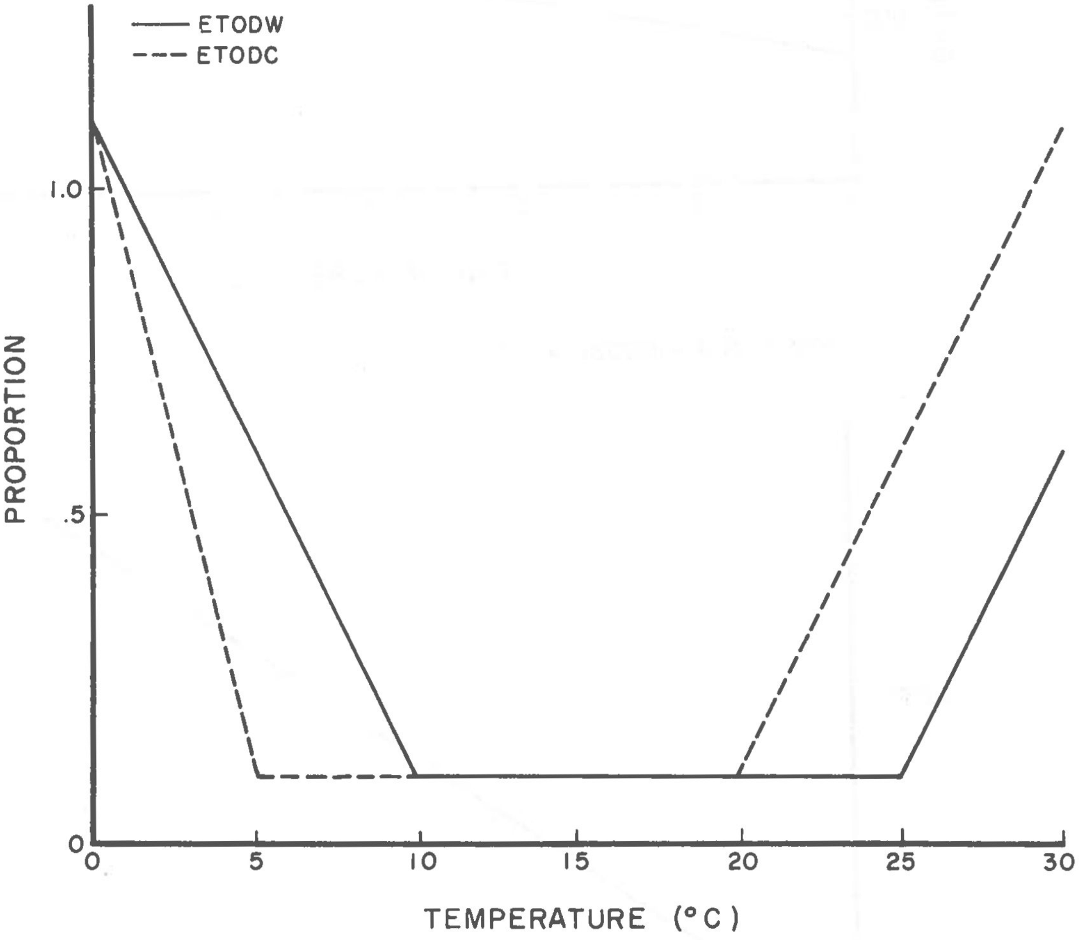
Consumer Submodel

The consumer submodel is part of the feedback loop with the producer submodel. Consumers exhibit control on plant growth as a function of the number of animals present. Control is attributed not only to grazing by consumers, but also to trampling loss. The model is designed to handle up to five livestock classes, e.g. steers, calves, cows, sheep, etc. All that is required for a consumer to be entered into the model is that the consumer



B = 550. + 200. * SIN((IDAY + 40)/365. * 6.28)

Fig. 7. Grams of water required to produce one gram dry weight of green plant biomass.



$$\text{ETODW} = \begin{cases} .10 * (\text{TR} - 25.) + .10 & \text{for } X > 25 \\ 1.1 - .1 * \text{TR} & \text{for } X < 10 \\ .10 & \text{otherwise} \end{cases}$$

$$\text{ETODC} = \begin{cases} .10 * (\text{TR} - 20.) + .10 & \text{for } X > 20 \\ 1.1 - .2 * \text{TR} & \text{for } X < 5 \\ .10 & \text{otherwise} \end{cases}$$

Fig. 8. Effect of temperature on death of cool (ETODC) and warm (ETODW) season plants.

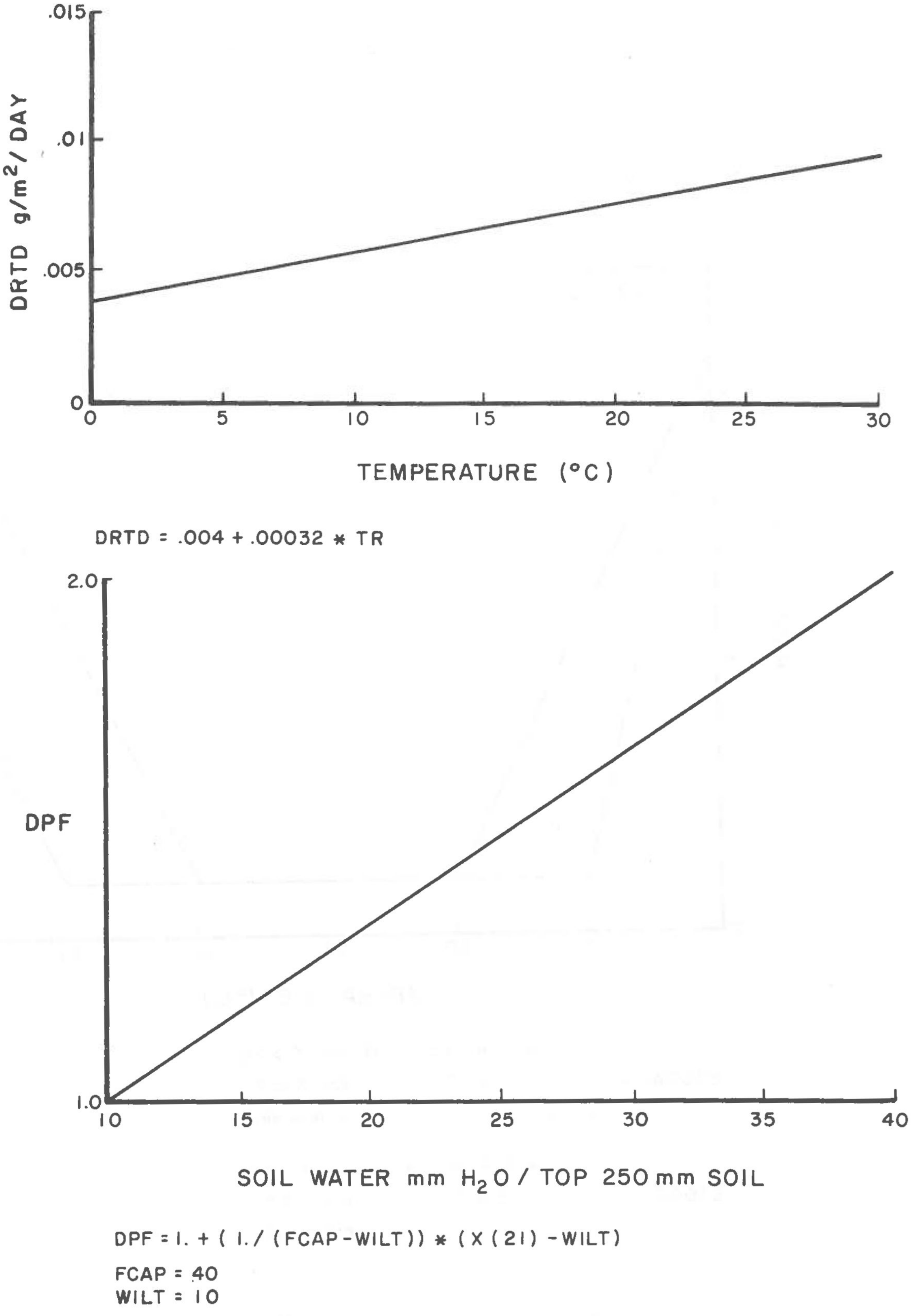
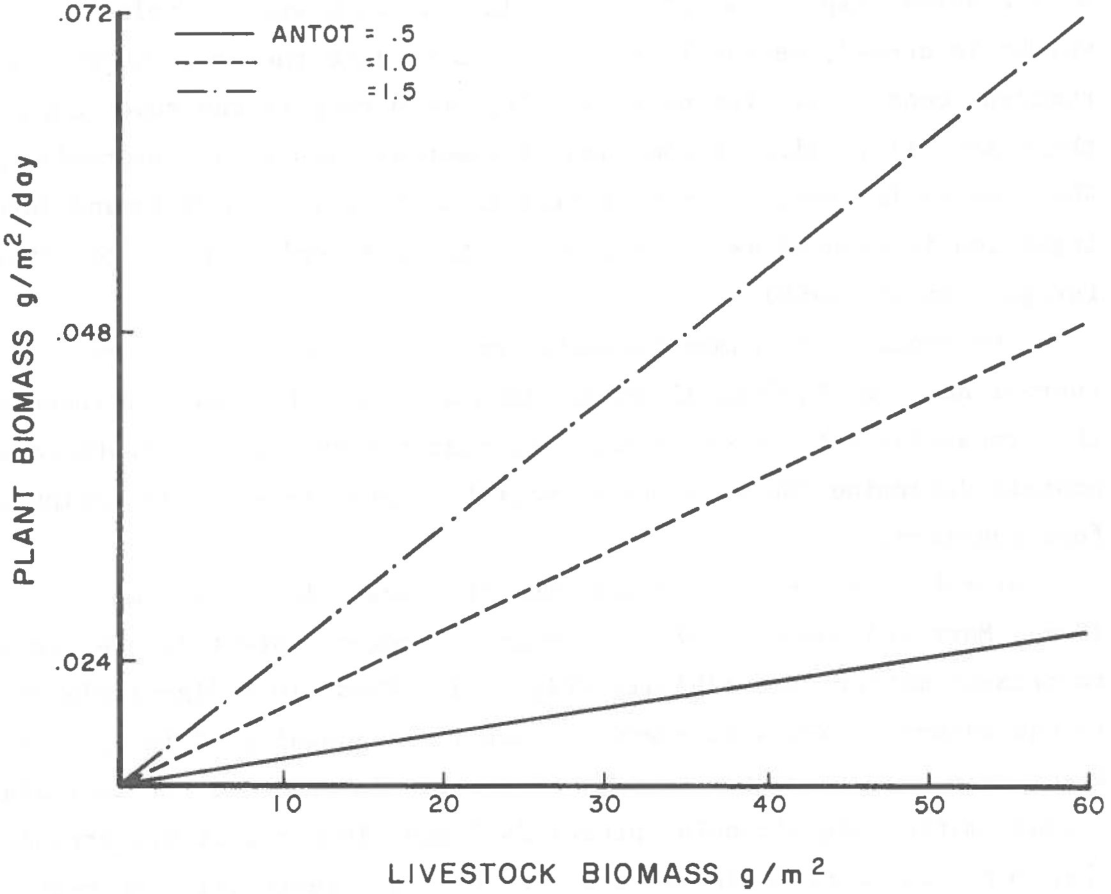


Fig. 9. (a) Dry plant material flow rate from standing dead to litter as a function of temperature.

(b) Increased flow from standing dead to litter due to the presence of water.



Plant loss $(g/m^2/day) = .0008 * X(12) * ANTOT$

Fig. 10. Trampling loss from livestock for 3 grazing intensities.

be a ruminant herbivore and that certain pertinent information be known, i.e. number of individuals in the consumer class, average weight per individual, effect of supplemental nitrogen on forage consumption, etc.

The consumer's diet is composed of live and dead plant material. Ingestion rate is determined by simulating rumen activity (Rice et al., 1974). Rumen capacity is determined as a constant proportion (.16) of the metabolic size of the animal (kg liveweight $^{.75}$). In all cases where a relationship to body weight is needed, metabolic size is used to make the model more general for ruminant consumers. The potential ingestion rate is the rumen capacity minus the rumen fill, which is composed of rumen dry matter and microbial protein. When available forage is not sufficient to meet livestock demand potential, ingestion is reduced as a function of the consumer's ability to utilize limited forage (Bement, 1969).

The dynamics of rumen fermentation are central to the herbivore intake subroutine. The biological assumptions are that the rumen microbes control the processing rate of foodstuffs and that the available carbohydrate and protein determine the quantity of microbial protein which is acting on the food ingested.

Microbial protein yield per 100g of digestible organic matter is about 8-20g (Hume, Morr and Somers, 1970). Dietary nitrogen content is used as an index to organic matter digestibility (Fig. 11). Thus, 100g digestible organic matter is equivalent to 430 kcal energy. Then if microbial protein yield is 8g/g digestible organic matter, .0186g of protein is produced per kcal digestible organic matter (8g microbial protein/430 kcal/100g digestible organic matter). Therefore, as nitrogen in the diet increases, digestibility increases and the growth of microbial protein increases. However, when dietary nitrogen content exceeds 1.8% microbial protein, growth is assumed to hold constant rather than to be continually increasing.

Rumen microbial protein content determines how much of the potentially digestible food will be digested. Four percent of ingested foods are digested per day for each gram of microbial protein per metabolic size as determined from digestion versus microbial population data. As microbial protein increases, the fermentation digestion rate increases, which increases the normal exit rate of digestible material, leaving more space in the rumen for increased ingestion. Also, higher fermentation rates add to the normal passage rate of material from the rumen to the intestine, again increasing potential intake.

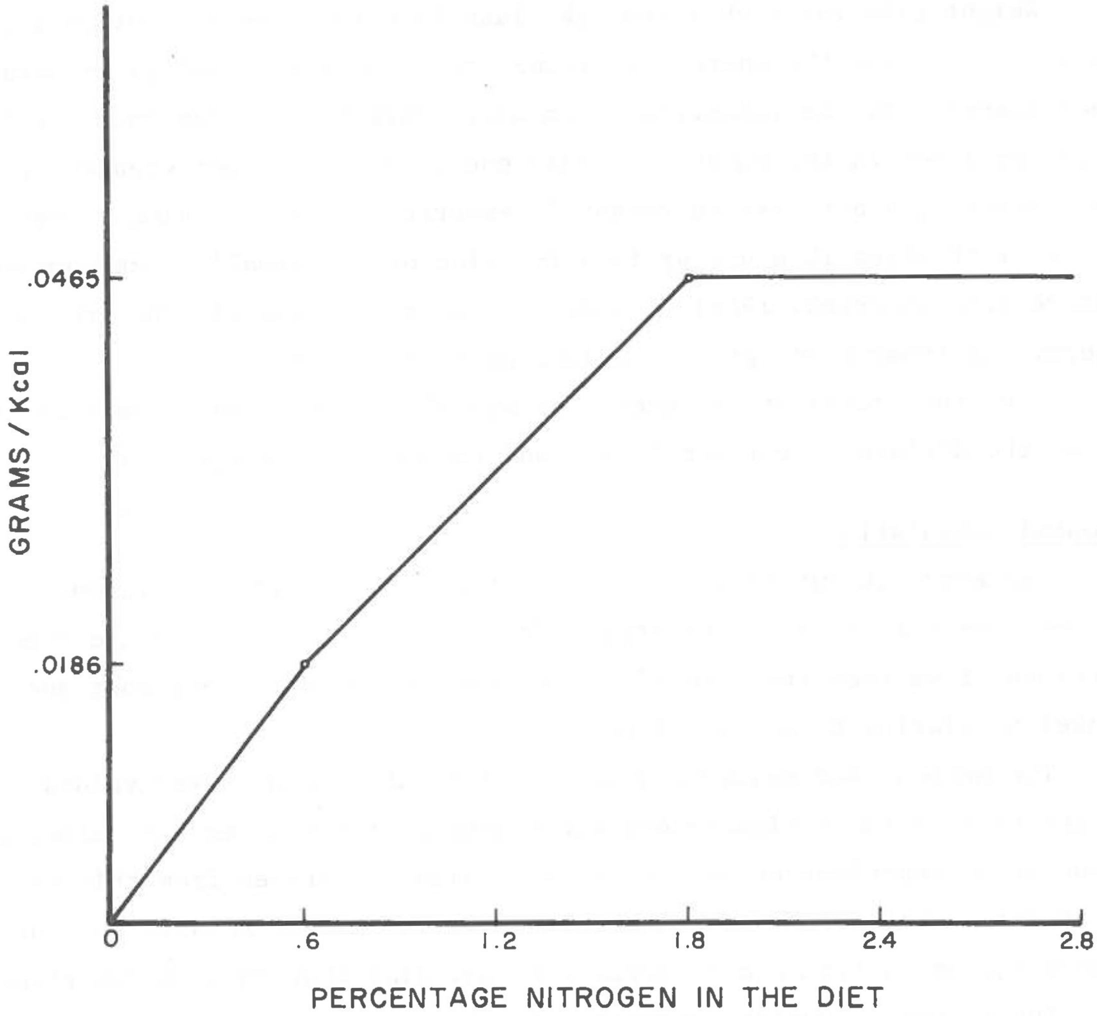


Fig. 11. Grams of microprotein resulting from each kcal of digestible organic matter as a function of percentage nitrogen in the diet.

The amount of nitrogen in the forage is determined by the relative plant growth rate (Fig. 12). The nitrogen in the diet is assumed to be higher than the average in the forage due to animal selectiveness. An approximate nitrogen value was obtained assuming herbivores consumed three times as much green as dry plant material. The nitrogen in the diet may also be increased by the presence of high protein supplemental feed.

Weight gain for each livestock class is determined by subtracting metabolic requirements from the energy resulting from digestion. Energy is obtained from food passed into the intestine, from microprotein, and from volatile fatty acids produced in the rumen. If this energy supply is not greater than metabolic requirements, a net loss in weight is experienced by the animal. Weight gain per kcal of digestible energy is a function of an animal's size compared to its mature size (Hedrick, 1968). Thus, the larger the animal, the greater the energy requirement per gram of weight gain (Fig. 13).

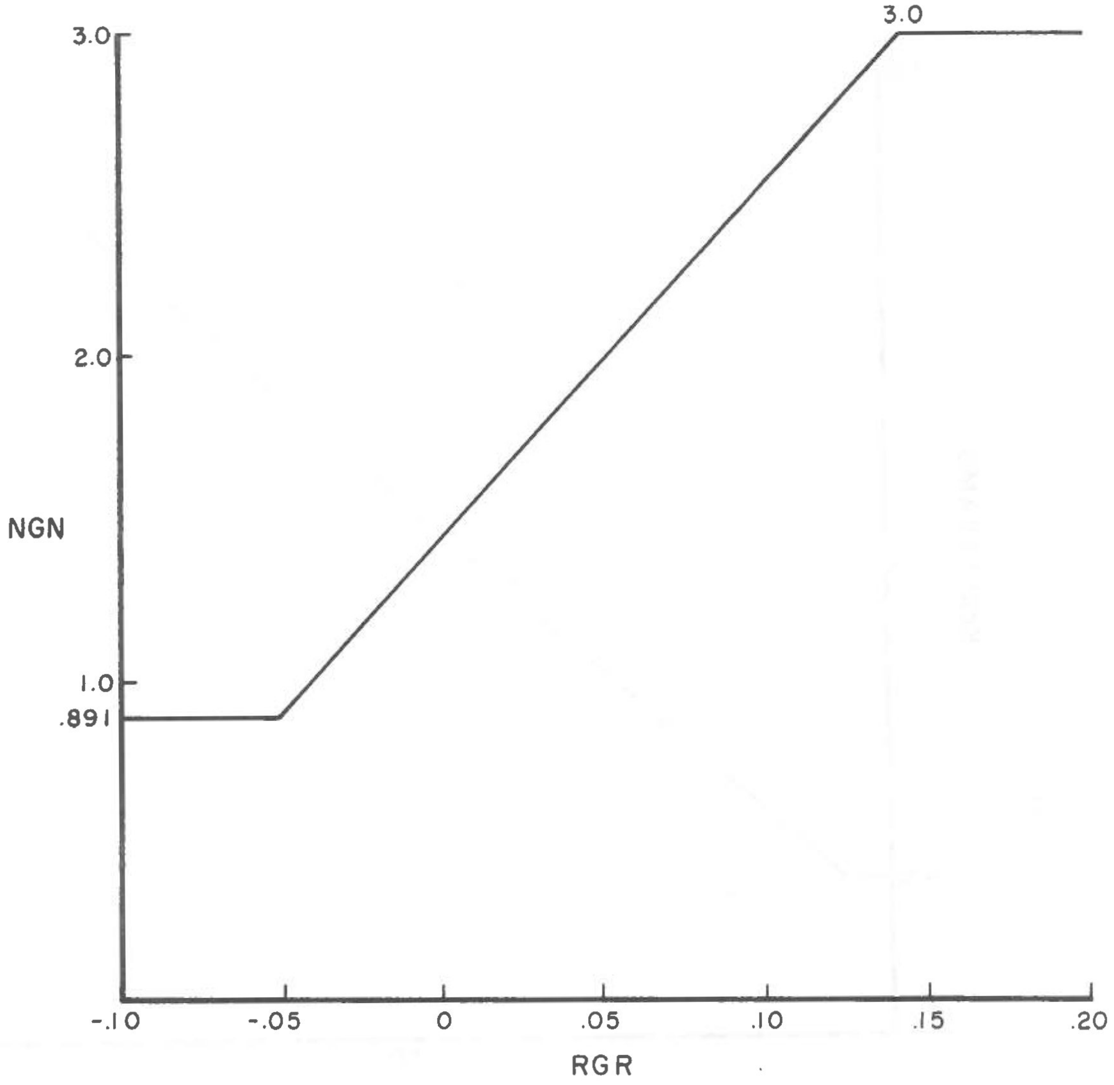
Livestock respiration, excretion and all other animal losses are assumed to be the difference between total food ingested and weight gain.

Economic Simulation

The economic submodel was developed to generate stochastic monthly market values from historical information. The value of livestock could then be determined if we know their initial cost and supplemental feed cost and assume rangeland grazing to be cost free.

The method used requires a vector of monthly mean market values and the square root of the variance-covariance matrix of the dependent market values. A vector of dependent normal random variables is derived from this information using the methods of Naylor et al. (1966). The method entails generating the random variables from a multivariate normal distribution with covariance.

The economic submodel assumes a cattle enterprise for determining the base market value. Some transformation would be necessary for conversion to sheep or other livestock values. During the simulation, prices are determined on a monthly basis and compared with the cash value of each livestock class enabling the current values of each class to be determined.



RGR = Growth rate per unit green plant biomass.
NGN = Percentage nitrogen in green plant biomass.

Fig. 12. Nitrogen content of green forage as a function of relative plant growth rate.

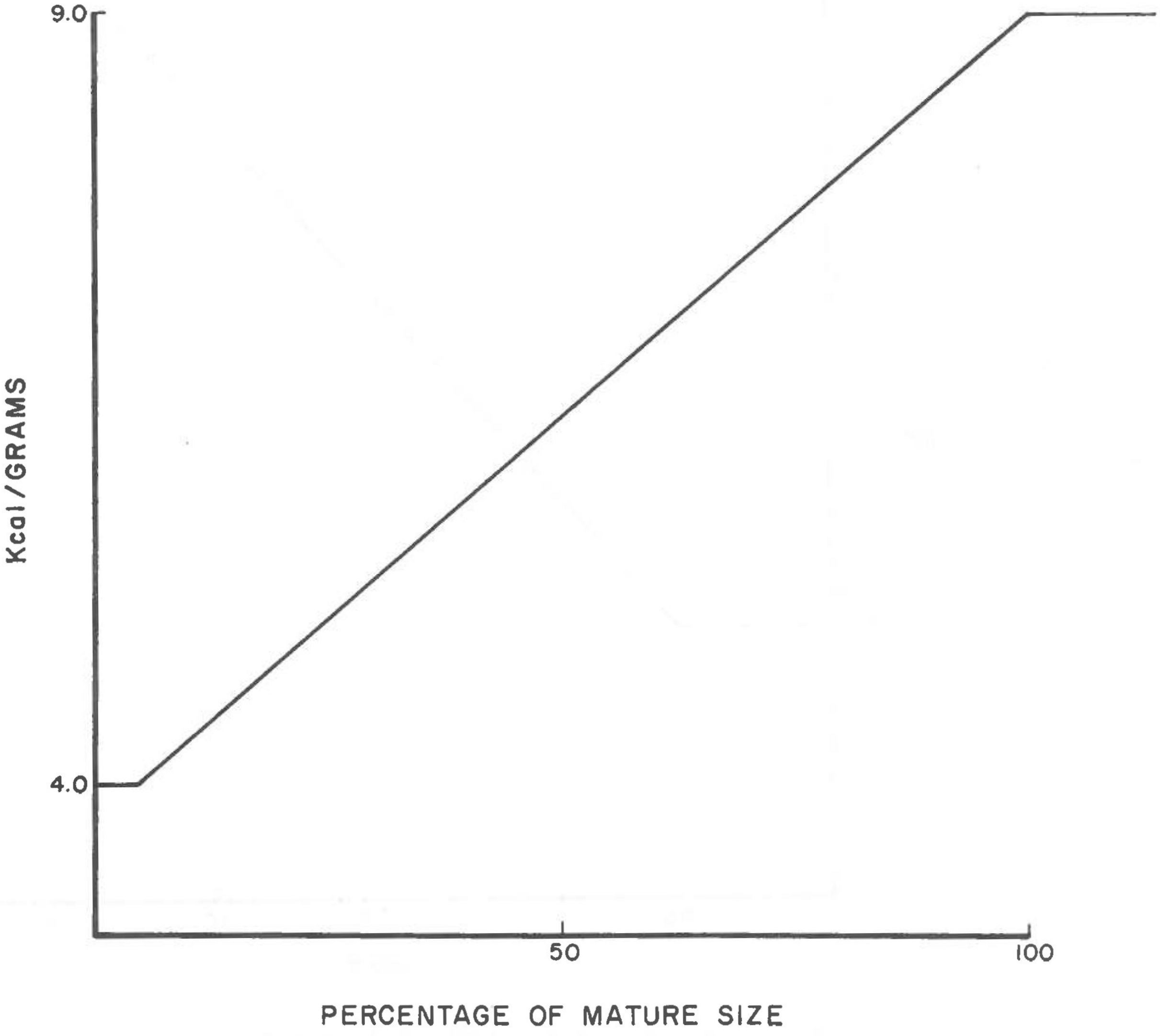


Fig. 13. kcal of available energy required to produce 1 gram of gain for animals at different stages of maturity.

USERS GUIDE TO THE RANGES GRASSLAND SIMULATION MODEL

RANGES is written in ANSI standard FORTRAN, and has been successfully exported to two other installations. A deck and this manual are available at cost from the Regional Systems Program, 325 Aylesworth Hall, Colorado State University, Fort Collins, Colorado, 80523.

The FORTRAN coded model is composed of two major parts. One is the biological simulation and the other is the simulation programming overhead (Fig. 14). In general, any simulation of interest could be substituted for the first section, while the second section would remain unchanged, except for the inputs.

Structural modifications to the model are described in a later section of this report, and give insight into the simulation method used. The model was developed in a modular fashion to allow changes and function substitutions. Subroutines describing biological functions such as plant growth (GROW), evapotranspiration (ET) and ruminant consumption (RUMEN) can be substituted directly for inadequate or obsolete functions, provided the subroutine parameter lists remain the same. A substitution for RUMEN illustrates this point (Appendix 3).

Deck Structure

There are three sections required for running this model: (i) control cards, (ii) FORTRAN program, and (iii) input data. The control cards required to run RANGES will vary from installation to installation. The control cards required at CSU are discussed below to give programmers at other installations an idea of the analogous commands to give other machines.

CARD NUMBER:

```
1 - HI527,AFNRCSCT,T40,PR100,CM60000.  RANGES.
2 - FTN,L=0.
3 - LGO.
4 - 7-8-9 (multipunched in column 1)
5 - (7 spaces)  PROGRAM RANGES (INPUT,OUTPUT,TAPE1=INPUT,
                TAPE2=OUTPUT,TAPE3=PLTSV,TAPE4,TAPE5)
```

Card 1 identifies the job and sets the physical requirements and limits for the job. It contains the job sequence number (HI527), the charge account number (AFNRCSCT), the time limit (40 central processor seconds), printed page limit (100 pages), core limit (60 K), and an identification code (RANGES).

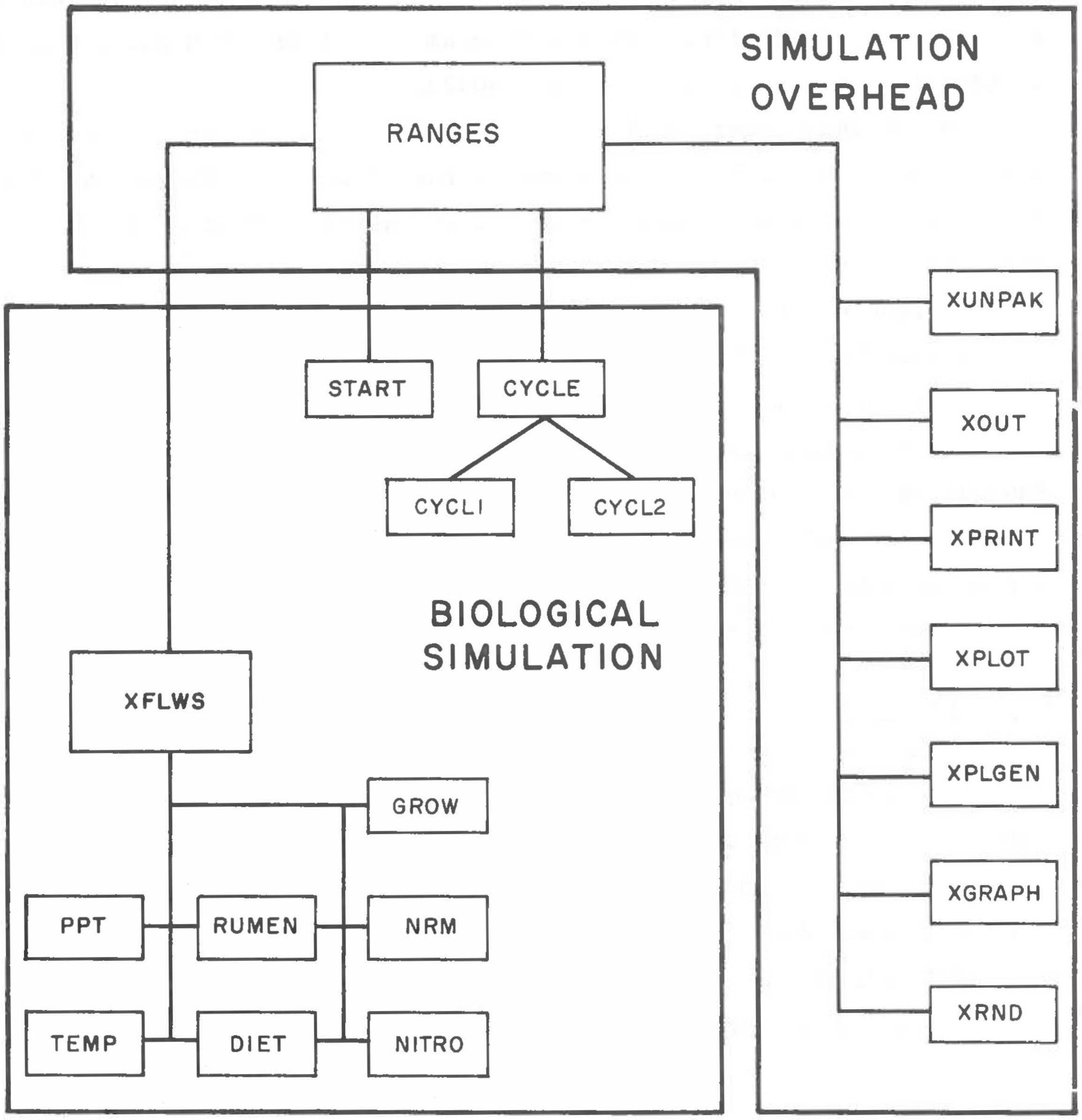


Fig. 14. RANGES program organization indicating modular design.

The model requires about 16 seconds to compile and about 16 seconds to execute for a one-year simulation with a one-day time step on a CDC 6400 computer. The number of printed pages varies according to the output requests and whether a program listing is obtained. The program listing alone requires about 60 pages including loader maps, and average printed simulation output is about 20 pages.

Card 2 (FTN,L=0.) loads the FORTRAN compiler into central memory and compiles the program. The L=0 parameter supresses the program listing to reduce printed output.

Card 3 (LGO.) loads the object code and executes the simulation with the results being printed on the system output device.

Card 4 (7-8-9) separates the control cards from the FORTRAN deck.

Card 5 (PROGRAM....) is the first card of the FORTRAN deck and specifies input and output devices. It is specific to CDC machines and is explained in Appendix 4.

The remainder of the FORTRAN deck (Appendix 2) follows Card 5 and is trailed by another separator card (7-8-9). The data section follows the end of record card (7-8-9) and can be composed of as many as four parts: (i) output control directives requesting variables to be printed and plotted, (ii) user supplied data cards which define the simulation input variables, (iii) data cards generated from the program PREGEN if stochastic precipitation is desired or sine function parameters for precipitation, and (iv) sine function parameters for temperature.

Different control cards are necessary if abiotic data is to be input from a magnetic tape; for example:

```
Job Card
FTN,L=0.
COPYCR,INPUT,DATA.
REQUEST,EXFILE,HY,ID-D0245,READ.
REWIND,EXFILE.
COPYCF,EXFILE,DATA,1.
REWIND,DATA.
LGO,DATA.
7-8-9
```

The gist of these control cards is to copy the data following the program deck, onto a file called DATA. Secondly, request a magnetic tape with the necessary abiotic data on it and copy that information onto the file called DATA following the information from the previous copy. Finally, execute the program with file DATA attached as the input file. It should be noted that the input format in SUBROUTINE RECRD should correspond to the format of the abiotic tape. Also, specification of stochastic weather generation requires running a separate program (PREGEN) to obtain input parameters for the geographic point of interest (Appendix 5).

Data Section

The data section follows the end of record card (7-8-9), which trails the FORTRAN deck, and consists of two parts. The first part of the data section is a set of output control directives requesting variables to be printed and plotted. The second part consists of user-supplied data cards which define the input variables for the simulation. The last data card must be a blank trailer.

Output Control Directives

All output control directives contain the following data fields:

COMMAND. n1,n2,n3,•••,n14.

The command begins in column 1 and contains no embedded blanks. Legal commands are PRINT, FLOW, PLOT, and END. The integer constants n1 through n14 are right-justified in fields of five columns each, starting in column 11. Note that some fields may not be used by some commands.

The card columns of each of the fields are:

Field	Card Column
COMMAND	1-10
n1	11-15
n2	16-20
n3	21-25
•	•
•	•
•	•
n14	76-80

The output directives PRINT, FLOW, and PLOT may appear in any order before the END. card.

PRINT Directives

The state variables X(J), J=1,99, may be requested for tabular output by PRINT directives. Upon encountering a PRINT command in columns 1 through 6, the card is scanned in numeric fields of five columns each, starting in column 11 and ignoring blank fields. All constants encountered are interpreted as the indices of the state variables to be printed. Note that the constants must have values which range from one to 99. There is no limit to the number of PRINT cards which may be used. The time interval between printouts is the value of DTPR.

FLOW Directives

The current values for any of the flows between state variables may be requested by the command FLOW.

The command FLOW appears in columns 1 through 5. Successive pairs of numeric fields are then interpreted as the indices of the flows to be printed. For example:

```
FLOW.     11  12
```

would print out the flows between X(11) and X(12). As many FLOW cards as necessary may be included. If a flow is requested which does not exist, the flow is ignored. The interval between FLOW printouts is the value of DTFL.

PLOT Directives

A graph of a state variable over time may be requested by a PLOT command. The command PLOT. appears in columns 1 through 5. The first numeric field in columns 11 through 15 is interpreted as the total number of plots or graphs which are needed. The rest of this card is left blank. The cards which follow the PLOT card must be blank in columns 1 through 10. The number of these cards must equal the total number of plots desired, as each card generates a graph. The indices of the state variables which are to be plotted together appear on the same card in the first five numeric fields. There can be at most five variables plotted per graph and at most 20 graphs, i.e. at most, 20 cards after the PLOT card.

Example:

```
                    PLOT.        2
                                80
                                21    11    12
```

In the above example, there will be two graphs. The first will plot state variable 80, and the second graph will plot state variables 21, 11 and 12.

The interval between plotted values is DTPL. This part of the data section must be terminated by an END. card beginning in column 1, whether or not tabular or graphical output is requested.

User Supplied Data

The second section contains initial values of variables for the simulation according to the following order and format (Note: the current state of the model requires DT = 1):

1st card: TSTRT, TEND, DT, DTPR, DTFL, DTPL
FORMAT (8F10.0)

2nd card: X(10), X(11), X(12), X(21)
FORMAT (8F10.0)

3rd card: NCLAS, MODT, MODP, INYR, AREA, FCAP, THOLD,
WILT, SOILA, SOILB, TINIT
FORMAT (3I1, I2, 5X, 7F10.0)

Note: SOILA and SOILB are not presently used in the model but are available to the user for future development.

If NCLAS is zero, i.e. there are no herbivores present, then the following cards up to the weather data input are not read and should *not* be present in the data deck. Otherwise, there should be J=1, NCLAS cards following plus the price variance-covariance matrix and mean price vector.

IN (J), MKT (J), INFD (J), NOFD (J), LVWT (J), SUPRT (J),
SUPPR (J), STKNO (J), SUPN (J), MSZ (J)
FORMAT (4I3,8X, 6F10.0)

The next 24 cards contain the animal class price variance-covariance matrix, six elements per card.

PROCOV(12, 12)

FORMAT(7X,6(E10.4,1X))

The user would probably want to use the supplied variance-covariance matrix as it represents several years data and then modify PRMN, the mean net price to correspond with the current market status.

Note: The matrix is actually read into a vector PRCOV(144) by columns and later passed to subroutine NRM as a matrix.

The next two cards contain the mean monthly animal price per hundredweight.

PRMN(12)

FORMAT(7X,6(E10.4,1X))

The last two card sets are read if MODP or MODT equals 2 or 3.

MODP = 2: three sine function coefficients: H1, H2, H3

FORMAT(3F10.0)

MODP = 3: seven cards, the title and three sets of Fourier coefficients:

(NAME(I), I=1,3)
FORMAT (2A4,A2)
3 pairs:
NT,A1
FORMAT(I3,F10.0)
(A(I), B(I), I=1,NT)
FORMAT(12F6.4)

The first two cards have the number of terms and coefficients for the Fourier representation of the weekly averages of the probability of a dry day. The second two cards represent the probability of a dry day preceded by a dry day and the third pair represents the average weekly storm size.

Last card: MODT = 2 or 3: three sine function coefficients: H4, H5, H6

FORMAT(3F10.0)

The variables used above are defined in Appendix 1. An example set of user supplied data is given in Table 1. These inputs are representative of the Pawnee National Grasslands.

Table 1. Example input variables to the RANGES grassland simulation model.

	Parameter	Input Value	Dimension	Description
System Control Parameters	TSTRT	0.	Day	Starting simulation time
	TEND	365.	Day	Ending simulation time
	DT	1.	Day	Simulation time step size
Abiotic Parameters	X(21)	20.	mm H_2O/top 250 mm soil	Initial soil water level
	WILT	10.	mm H_2O/top 250 mm soil	Soil wilting point
	FCAP	40.	mm H_2O/top 250 mm soil	Soil field capacity
	THOLD	50.	mm H_2O/top 250 mm soil	Total water holding capacity of soil
	MODT	2.	Control variable	Temperature generated by sine function
	MODP	2.	Control variable	Precipitation generated by a sine function
	AREA	320.	Acres	Area to be simulated
	H1	14.	Dimensionless	Amplitude of precipitation sine wave divided by two
	H2	60.	Dimensionless	Displacement along time axis
	H3	0.	Dimensionless	Displacement along vertical axis
	H4	30.	Dimensionless	Amplitude of temperature sine wave divided by two
	H5	120.	Dimensionless	Displacement along time axis
	H6	22.	Dimensionless	Displacement along vertical axis
Biotic Parameters	X(10)	0.	g/m^2	Initial warm season plant biomass
	X(11)	0.	g/m^2	Initial cool season plant biomass
	X(12)	20.	g/m^2	Initial standing dead plant biomass
	IN	0.	Day	Stocking day
	LVWT	400.	lb	Initial average body weight for animal class
	MKT	365.	Day	Market day
	SUPRT	.001	Body weight proportion	Supplemental feed rate
	INFD	280.	Day	Begin supplemental feeding
	NOFD	365.	Day	End supplemental feeding
	SUPPR	.025	$/lb	Supplemental feed cost
	STKNO	30.	Number	Number of animals
	SUPN	25.	%	Supplemental feed nitrogen content
	MSZ	1200.	lb	Mature body weight for animal class
	NCLAS	1.	Number	Number of livestock classes
Economic Input Parameters	PRMN	Matrix	$/cwt	Mean net prices
	PRCOV	Matrix	$/cwt	Price variance-covariance matrix

Modifications

Additional output. If additional output is needed, FORTRAN WRITE statements using logical unit number 2 or FORTRAN PRINT statements may be inserted in the program wherever they are desired. They must be accompanied by the appropriate FORMAT statements.

Additional state variables and flows. In order to add additional state variables, the variable XNST and the array XST(99) must be altered in the main program. XNST is the number of state variables used and XST(99) is a list of state variable indices.

For example, if the state variable 7 is to be added to the program, the value of XNST must be increased by one, and the number 7 entered at the end of the array XST(99).

In order to add additional flows to the program, the variable XNF and the array XFR must be altered in the main program. XNF is the number of flows and XFR is the flow reference table.

For example, if the flow from state variable 7 to state variable 39 is to be added, the value of XNF is increased by one and 739 is entered as the last entry in the array XFR.

In addition to the above changes, the FORTRAN code of the flow must be added in SUBROUTINE XFLWS as follows:

```
XN = XN + 1
   F = 0.
     .
     .
     .
FORTRAN CODE
     .
     .
     .
 XF(XN) = F
```

where F is the flow between 7 and 39.

Substituting Subroutines

RANGES was specifically designed to keep processes, as much as possible, within one routine. Thus, new subroutines can take the place of the original ones provided the subroutine argument list remains the same. Care must be taken that variables used in other parts of the program are not left undefined in a new subroutine. If additional variables are needed in a subroutine, some minor programming changes can be made to meet this end. This capability allows detailed site specific information to be readily entered into the model in functional form. An example of such a substitution is shown in Appendix 3.

CASE EXAMPLES

Applicability of the model to a wide range of conditions is presented as a series of case examples run on three different experimental range sites. First, the whole model is exercised using Pawnee Grasslands input parameters. The Pawnee Site is simulated again, only without grazing, and the results are then compared to simulations of the Dickinson Site in North Dakota and the Eastern Colorado Range Station (ECRS) in Colorado. The only alterations required when running the model on any of these sites are new soil water parameters and appropriate driving variables for the site, i.e. temperature and precipitation data.

Example 1: Pawnee Grasslands

This example illustrates all of the model components, i.e. soil water, plant production, livestock grazing and economic considerations. The discussion of the model function should help when interpreting the other case examples. Input parameters used to generate these results are given in Table 1.

A simple sinusoidal function was used to generate daily precipitation and temperature values for a one-year run. Soil water rises sharply during the spring when there is precipitation and little evaporation to remove water from the soil. As soon as the temperature increases, soil water is lost both through evaporation and transpiration (Fig. 15). Above 5°C cool season plants initiate growth and above 10°C warm season plants begin growing (Fig. 16). A five-day growth initiation period is allowed for the plants, after which 70% of the photosynthate per day is assumed to be translocated to the roots. Plant production responds to soil water level and average daily air temperature. Growth continues, contingent on soil water availability, until 45 days have elapsed or a decrease in production is experienced. Either of these conditions causes senecence to begin and the accumulation of standing

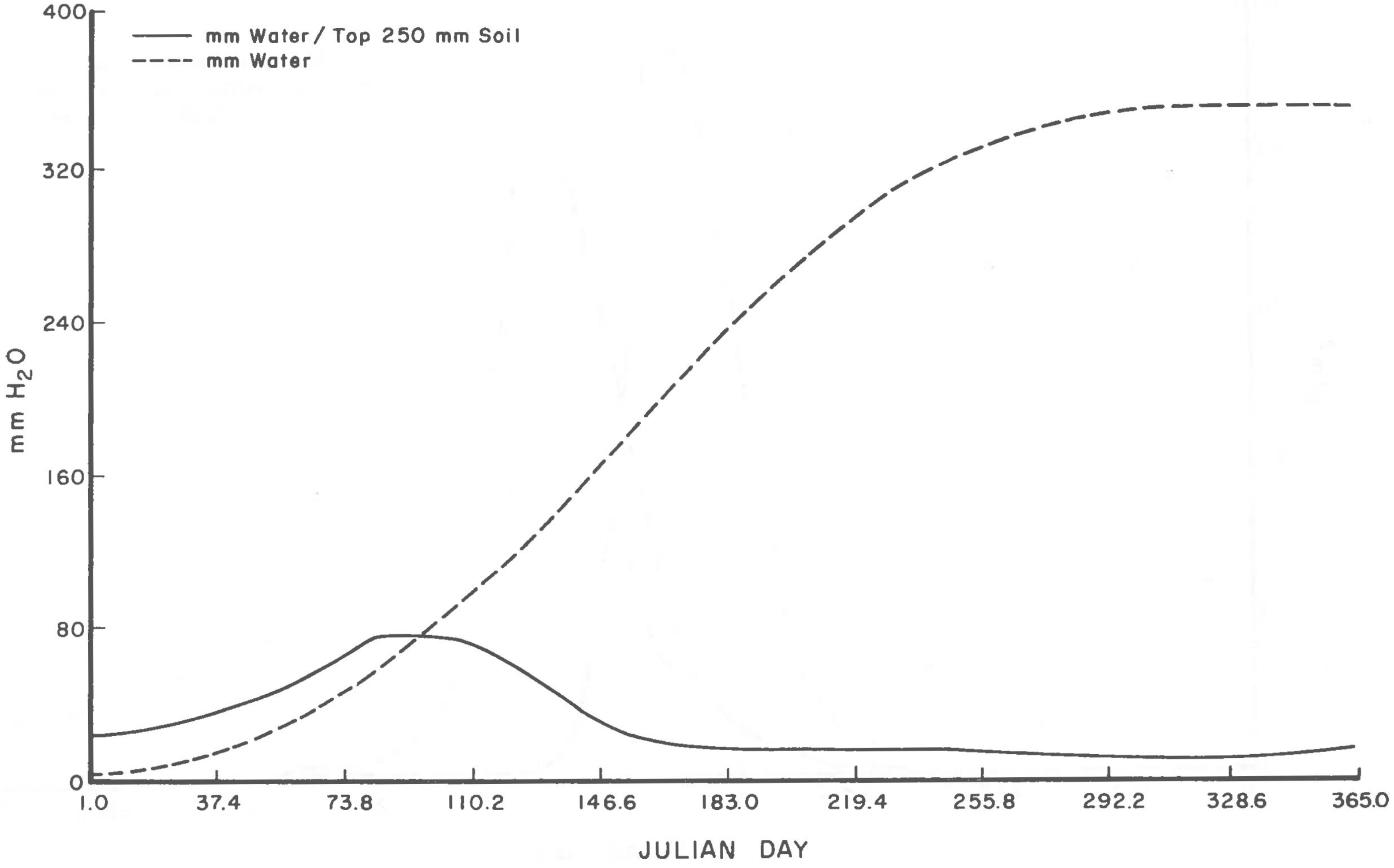


Fig. 15. Simulated soil water content and accumulated precipitation for the Pawnee National Grasslands.

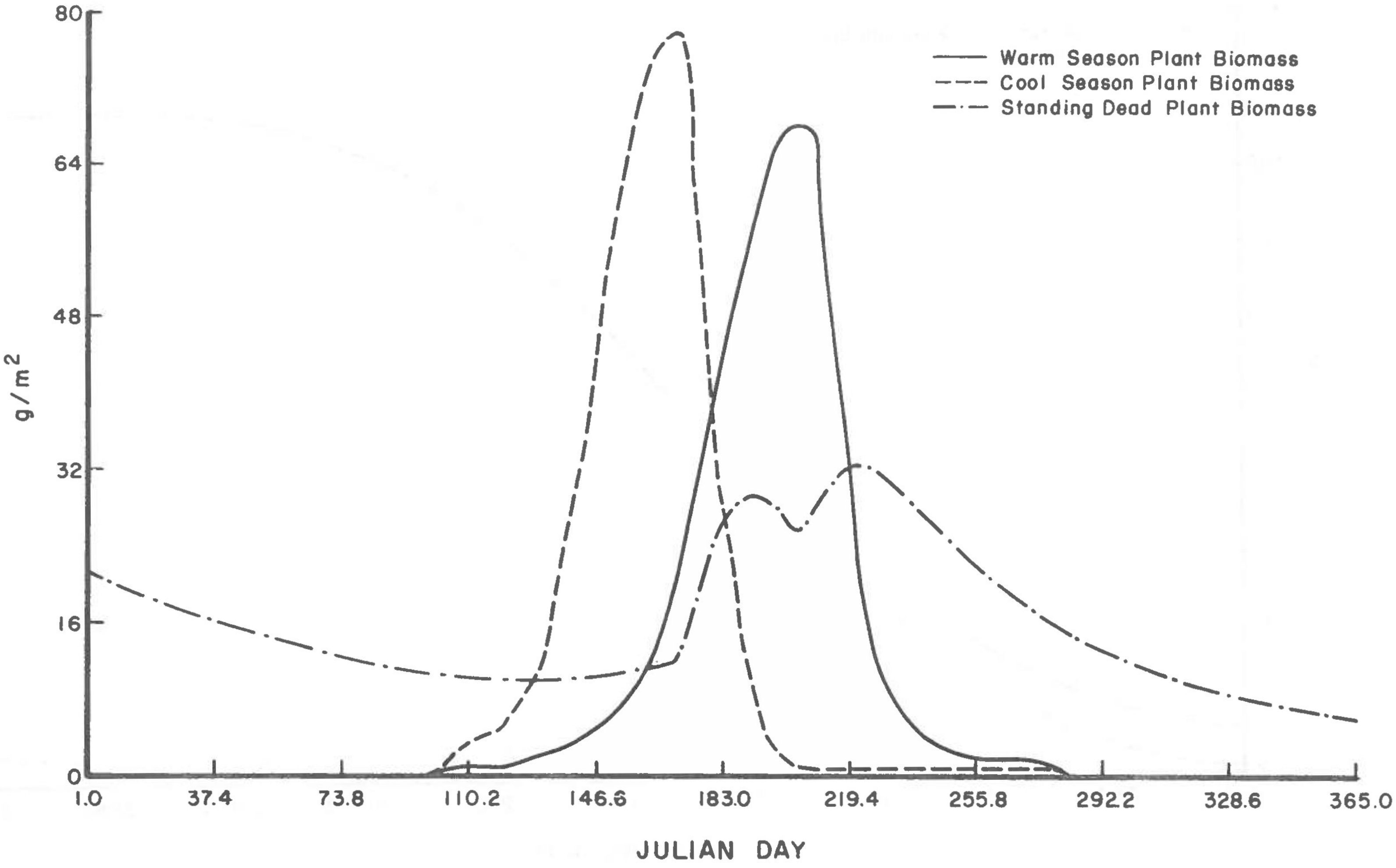


Fig. 16. Simulated cool and warm season plant production and standing dead plant biomass for the Pawnee National Grasslands.

dead plant biomass which results. Early in the year, the previous year's standing dead plant biomass is decayed by the presence of spring moisture. After the phenological peak, standing dead plant material begins accumulation until the decay rate exceeds the accumulation rate.

Temperature is used to regulate plant mortality with extremes of hot or cold increasing the death rate. Soil water is used as an index to the flow of standing dead plant material to litter in conjunction with temperature. Thus, standing dead plants decay rapidly in the spring when it is warm and wet, and in the fall, if precipitation occurs. Again, the presence of livestock contributes to litter accumulation as well as trampling loss to green plant material.

Simulated cattle grazing (30 steers averaging 400 lb each) was begun at day one and continued for the entire year (Fig. 17). Lack of available forage and low quality forage caused the cattle to lose weight rapidly until plant growth began. After this point they gained weight at a reasonable rate (1.6 lb/day) for the remainder of the growing season. At day 280 supplemental feed was given which kept the cattle from losing weight late in the season when forage was again at an inadequate level. Knowing the purchase price and the cost of supplemental feed allows the net cost of cattle to be determined, assuming no cost for grazing on private rangeland. Market value for each month of the year is generated stochastically (Fig. 18) in the model and allows net financial gain to be determined (Fig. 19). As many as five classes of ruminant livestock can be simulated per run; however, the market values presented here are based on cattle prices. Note that random number generators in different computers will result in different livestock prices and market values.

Example 2: Pawnee Grasslands

The Pawnee Site has sandy loam soils and a semiarid climate with an average yearly rainfall of approximately 11.7 inches. Warm season plants are generally predominant; however, the relatively large annual precipitation used in the simulation (14 inches) gave more cool season plant growth early in the season than warm season plants ever achieved (Fig. 20). This phenomenon has been observed at the Pawnee Site. Notice this run produced more than the previous run (Example 1) because livestock were absent.

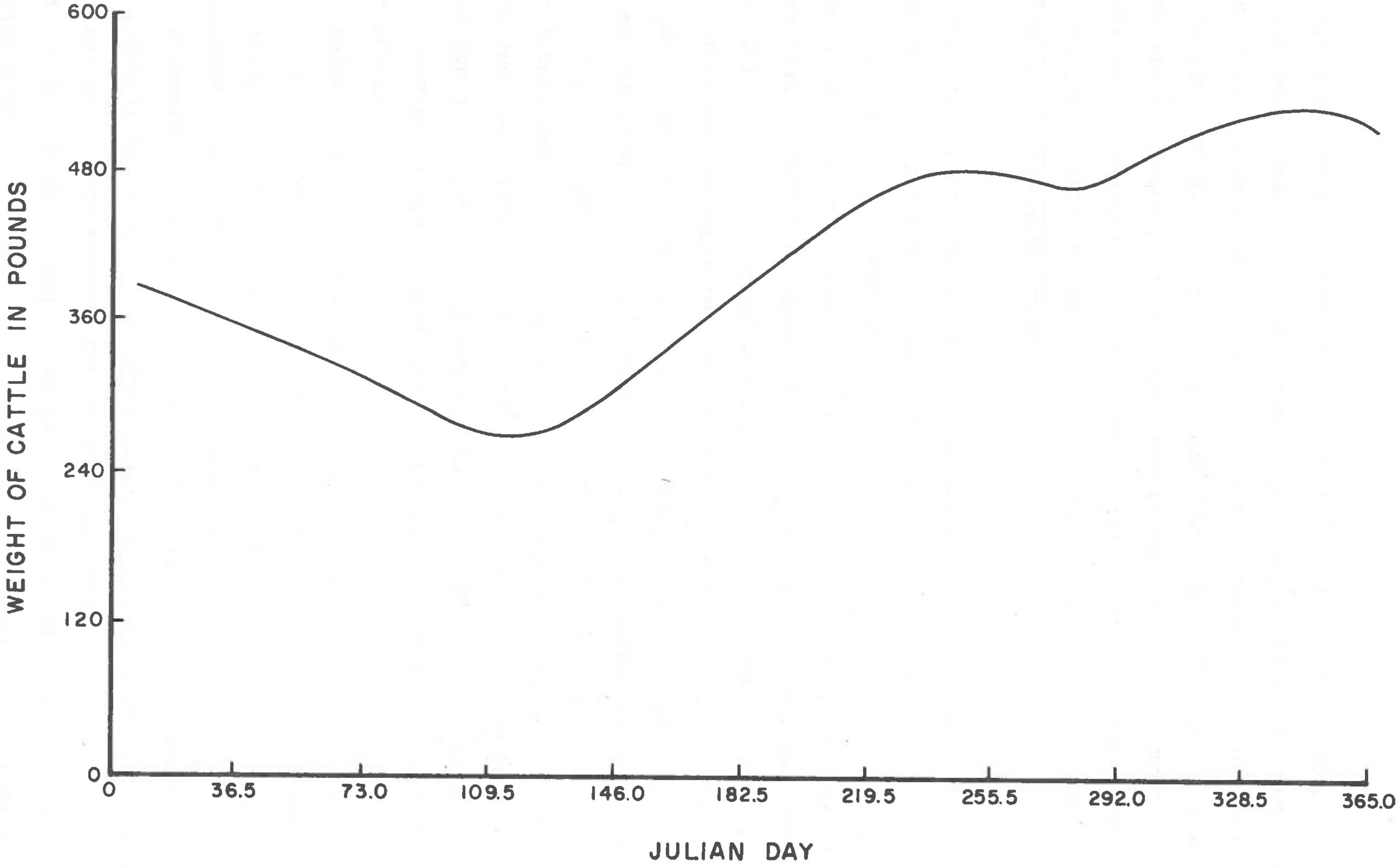


Fig. 17. Simulated weight for steers grazed for the entire year on the Pawnee National Grasslands.

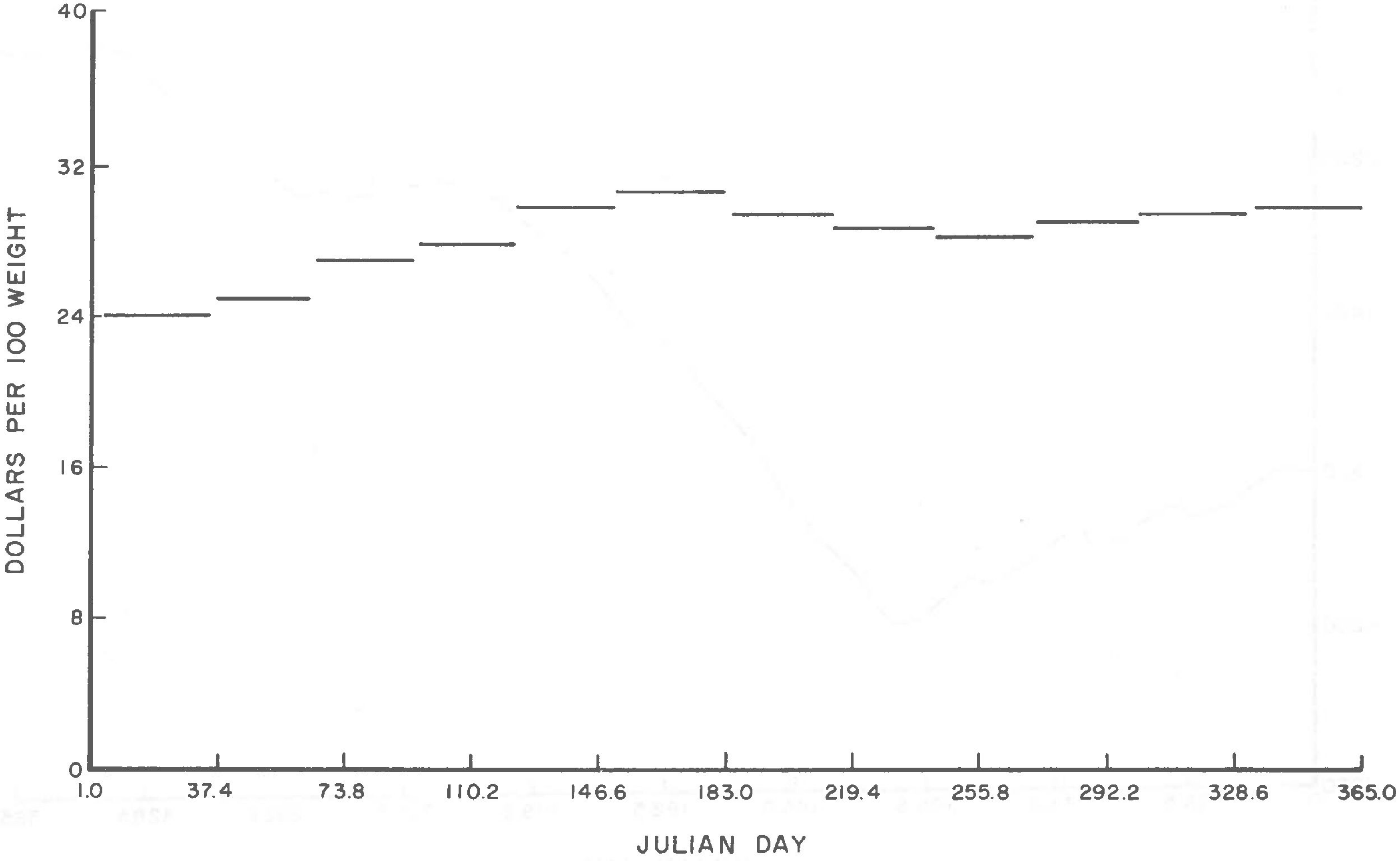


Fig. 18. Simulated stochastic market values for cattle.

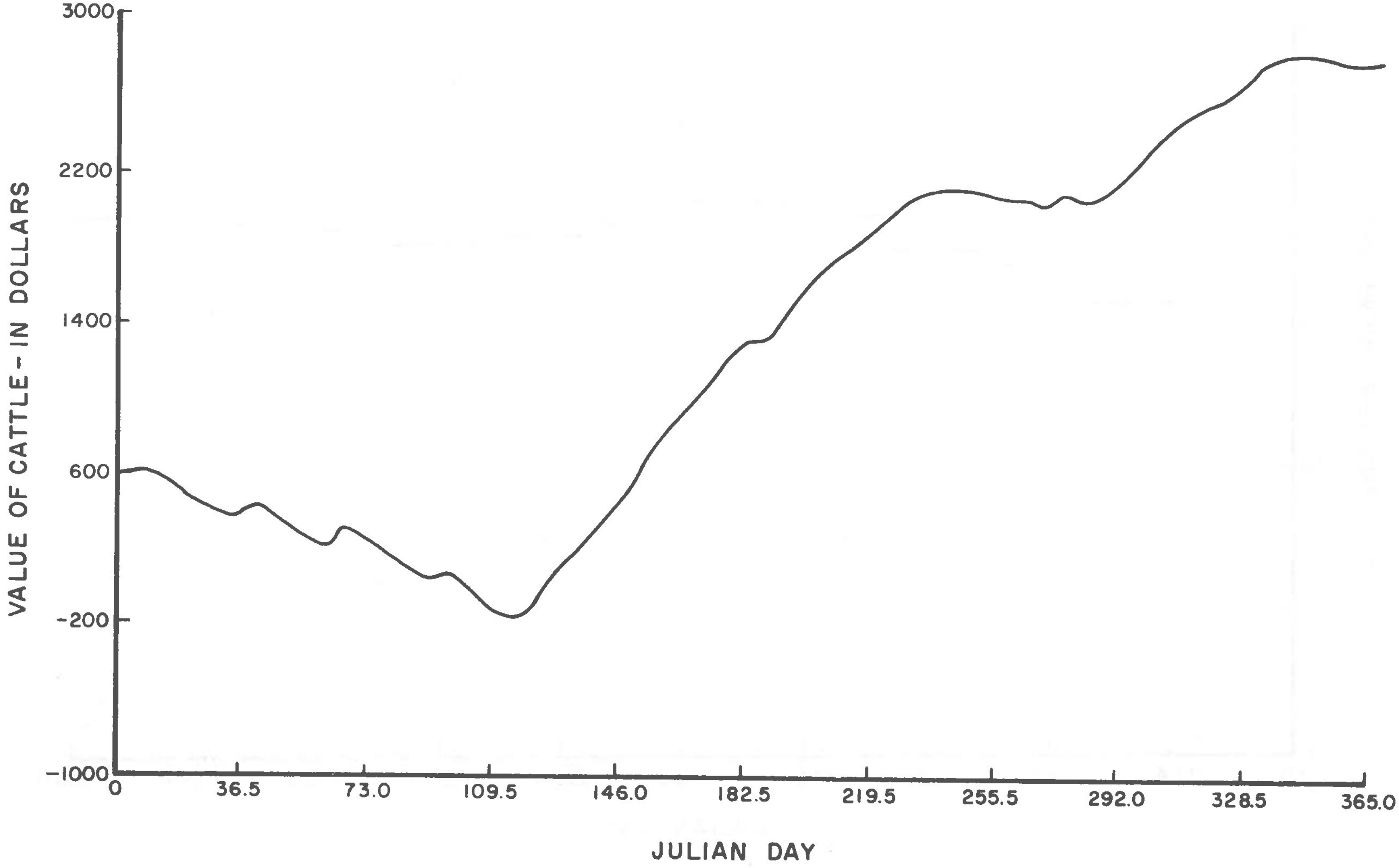


Fig. 19. Total market value of steers being grazed.

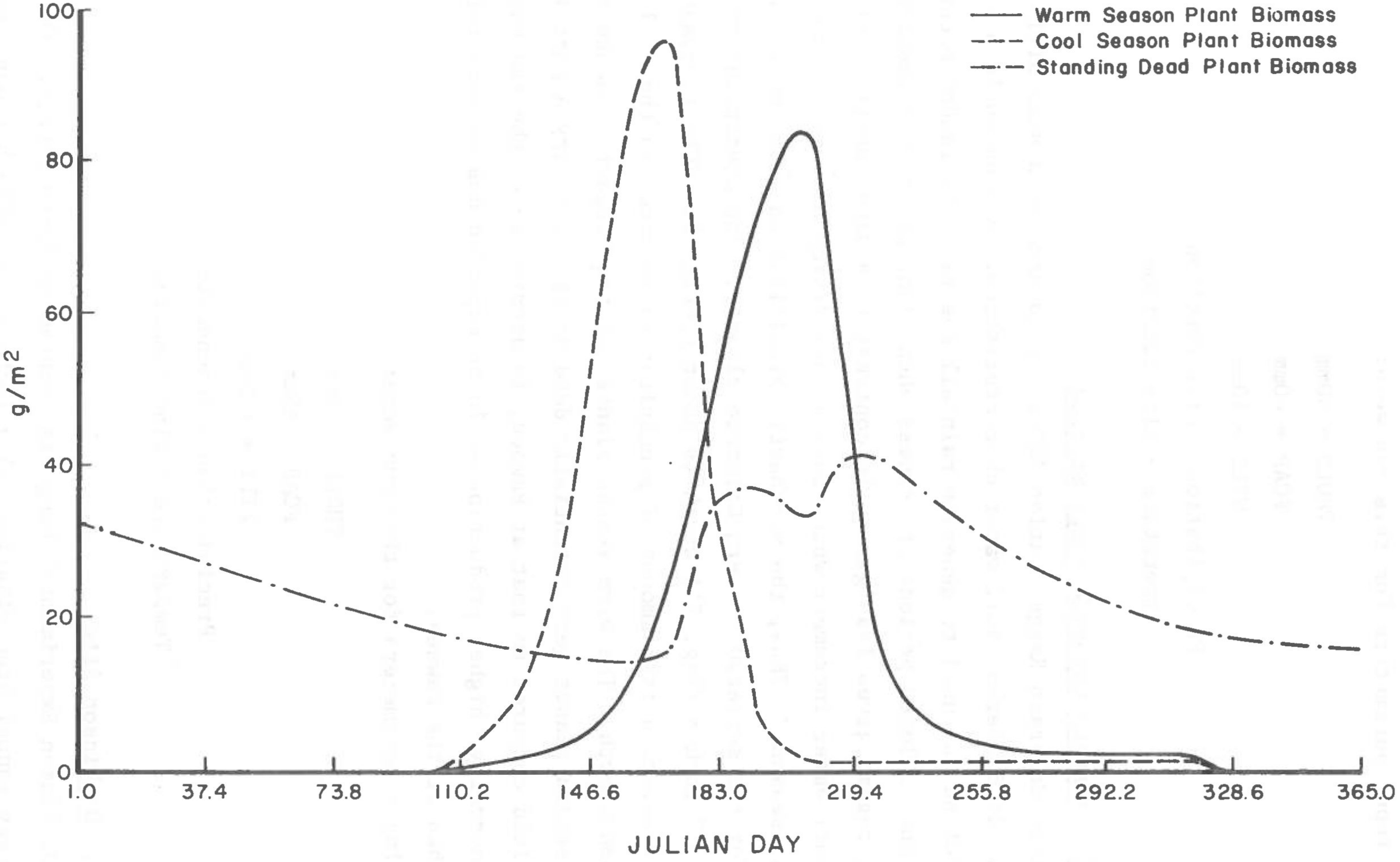


Fig. 20. Simulated cool and warm season plant production and standing dead plant biomass at the Pawnee National Grasslands without cattle grazing.

The input parameters for this run were:

THOLD = 50mm
FCAP = 40mm
WILT = 10mm
Precipitation - sine function
Temperature - sine function

Example 3: Eastern Colorado Range Station

Eastern Colorado Range Station (ECRS) is located on a sand hills site which gave it different soil water characteristics. A stochastic precipitation routine was used to generate rainfall events. The random occurrence of the events allowed periods of dryness when plant growth was reduced. The resulting regime gives less growth in contrast to a sine function which gives a daily soil water increment which only allows drying to occur at the end of the "rainy season." Thus, the stochastic precipitation gives more realistic results for the semiarid eastern Colorado climate. The precipitation regime yielded 15.7 inches (Fig. 21) which is about average for ECRS (average = 16.7 inches); however, a fair amount of precipitation occurred in the fall, giving late season growth. The warm season plants had the biggest response because the cool season plants were essentially dead after a hot, dry August (Fig. 22). Similar yield compared to that at Pawnee, is attributed to the stochastic weather; normally, higher production would be expected due to more rainfall at ECRS than at the Pawnee.

The input parameters for this run were:

THOLD = 35mm
FCAP = 25mm
WILT = 7.5mm
Precipitation - stochastic
Temperature - sine function

Example 4: Dickinson Site

The Dickinson Experimental Range is located in North Dakota. It receives slightly less annual precipitation (15.9 inches) than ECRS but has lower average temperatures and thus, lower evapotranspiration rates. Consequently, more

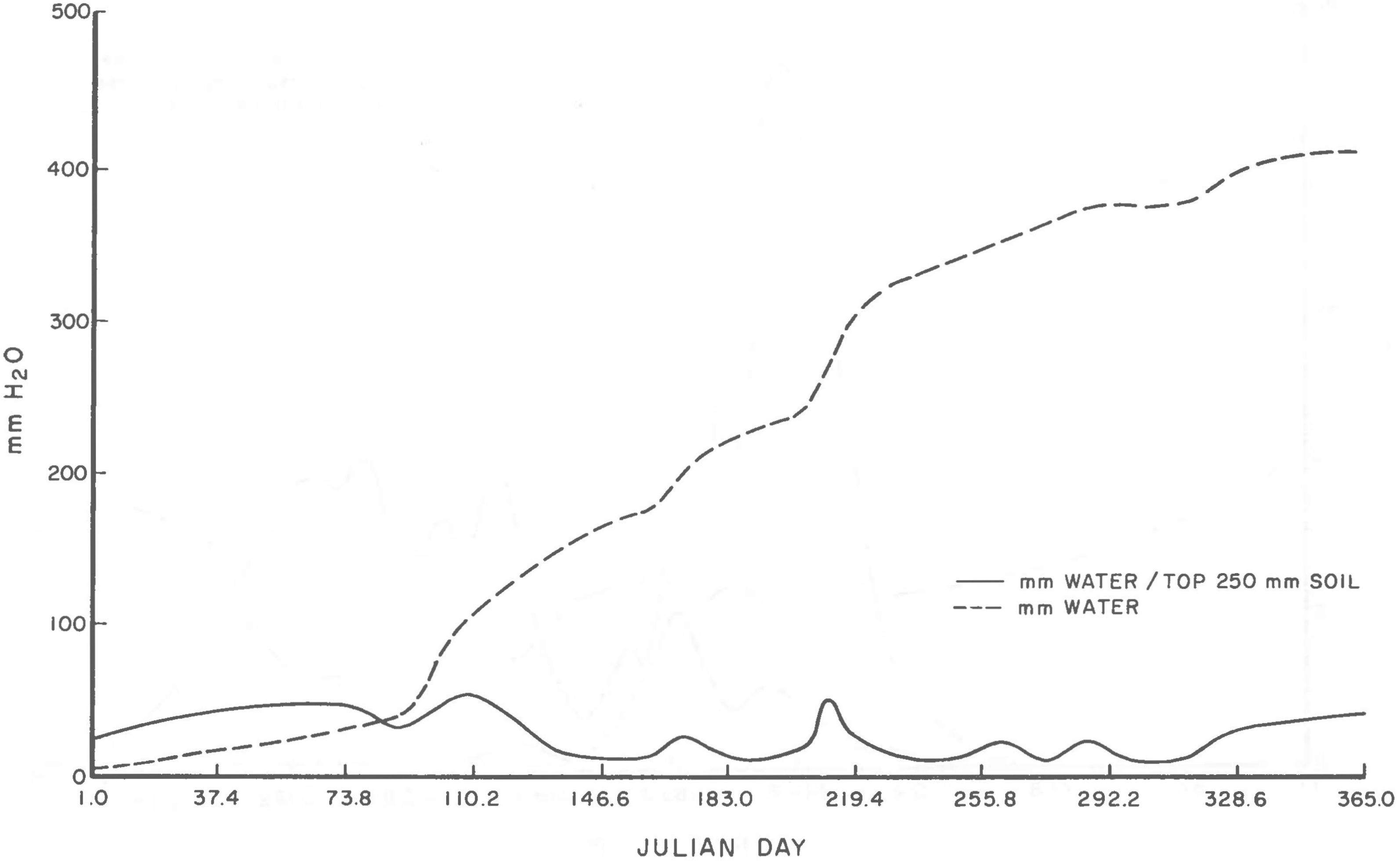


Fig. 21. Simulated stochastic precipitation and resulting soil water content at the Eastern Colorado Range Station.

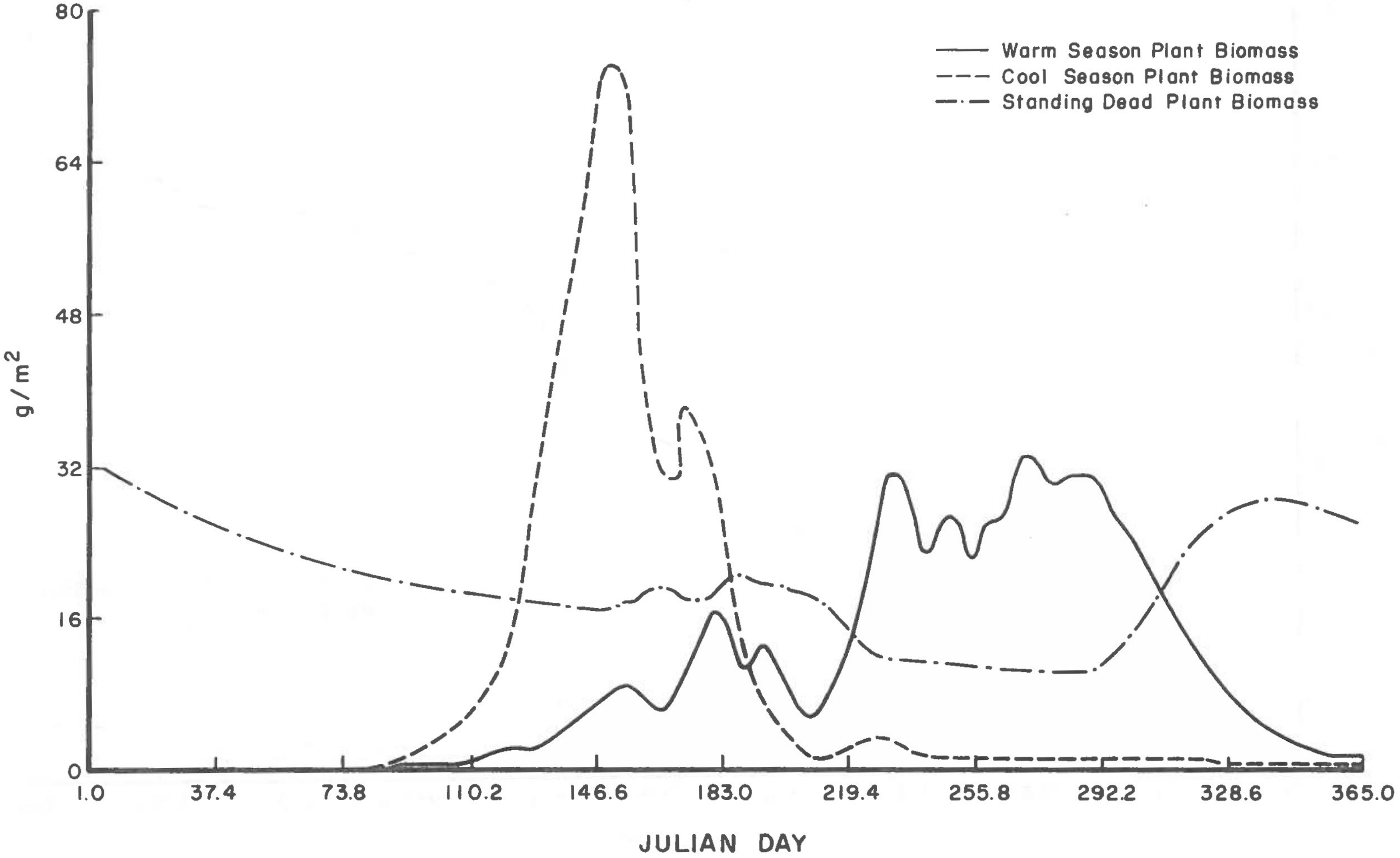


Fig. 22. Simulated cool and warm season plant production and standing dead plant biomass under a stochastic precipitation regime without cattle grazing at the Eastern Colorado Range Station.

growth can be expected at the Dickinson Site. In fact, this is what happens both in the field and in the model results (Fig. 23). Another expectation would be an increased ratio of cool season to warm season plants, which is accurately simulated by the model. The soils at Dickinson are of the Flasher Vebar Complex.

Input parameters were:

THOLD = 38.5mm
FCAP = 35mm
WILT = 15mm

Precipitation - Fourier series representation of historical data (a sine function was not used because of multiple dips in the average precipitation curve)
Temperature - sine function

CONCLUSIONS

The work reported here was based on a hypothesis regarding modeling ecosystems. The hypothesis is that complex ecosystem level models are, in general, too sophisticated to be readily usable by management personnel and that their heavy data requirements restrict the ease with which they can be used at different sites. Therefore, a management oriented model was proposed which would focus on the crucial mechanisms that were determined by ecosystem level modeling efforts in an attempt to represent a multitude of possible sites with only a minimal data requirement. The model developed from these hypotheses has been tried on three sites.

The submodel that is important in terms of representing a site adequately is the plant growth model. The livestock submodel will respond according to plant production so its variation is primarily a function of the plant model. The soil water submodel, on the other hand, is instrumental in simulating plant production. Therefore, the evapotranspiration and plant production functions must be very versatile and give reasonable results. The model was tested on three different sites for its ability to represent each site by changing only the specified input parameters. First attempts at running the

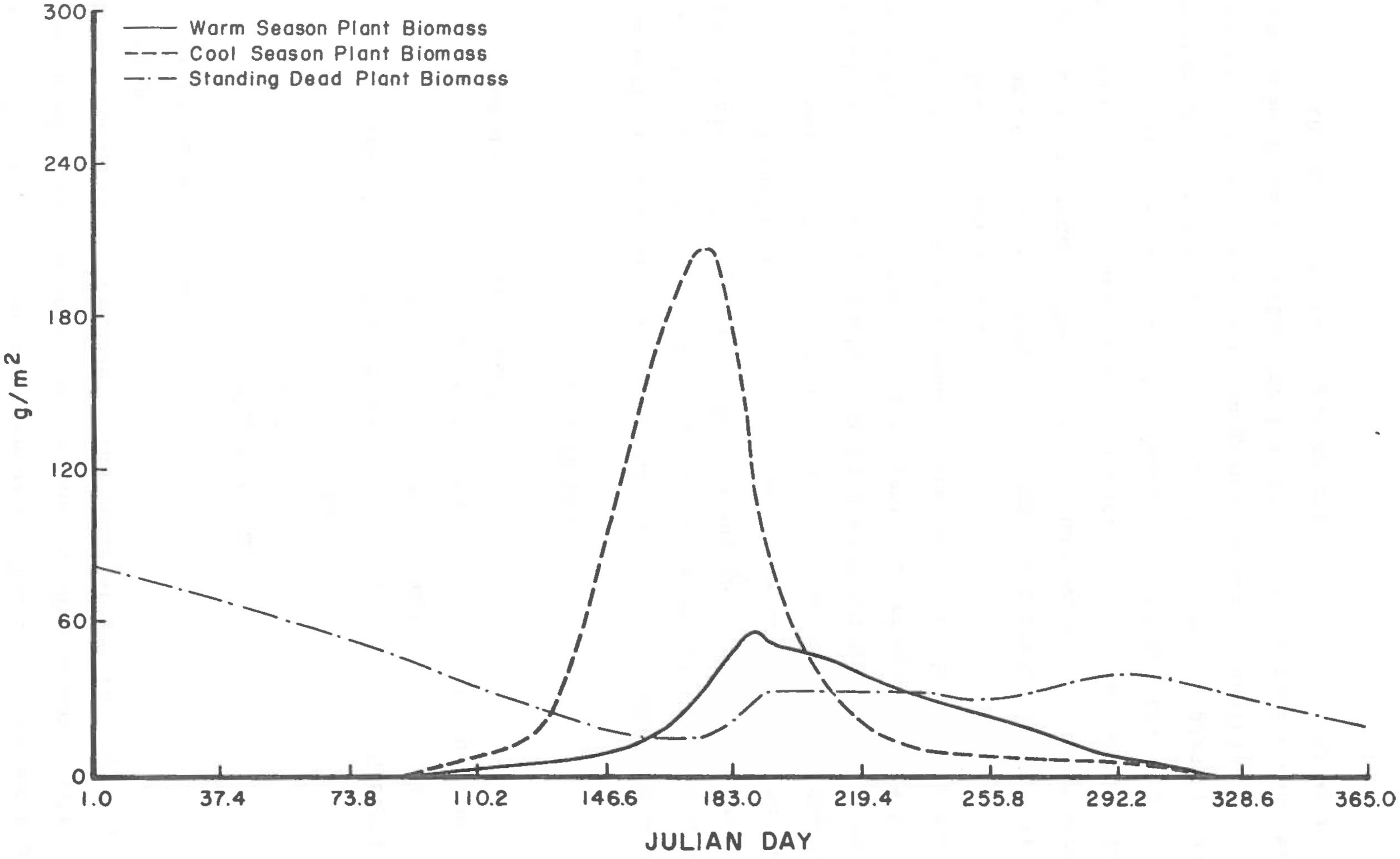


Fig. 23. Simulated cool and warm season plant production and standing dead plant biomass without cattle grazing at the Dickinson Experimental Range.

model at a new site using parameters tuned for the Pawnee site produced erroneous results. In order to adequately represent multiple sites compromises were made on some parameters. The resulting model gives reasonable results, but has some problems. For example, the simulated cool season plants exceed warm season plant production at the Pawnee site. This is not the usual case; however, under some conditions and on certain sites it does occur.

A second problem with the model is the timing of warm season growth. It begins growth at the appropriate time but often continues to produce well into the fall, a phenomenon not usually observed. Finally, the plant production at the Eastern Colorado Range Station appears a bit low considering the amount of rainfall received. However, given that the precipitation was stochastic, the effective moisture may not have been very high.

Such problems are to be expected when a model is built attempting to incorporate realism and generality at the expense of precision. The problems can usually be remedied by specifying some parameters for the particular site. But this should be done by a user and was not the intent of the modeling activity. Another possible solution to model inaccuracies is increased resolution in some of the mechanisms or additional mechanisms that may be important at a particular site.

With the results to date it would be impossible to state conclusively that a simple general model capable of handling a wide variety of sites can be built. On the other hand, it is not possible to say it cannot be done as this effort has shown. Rather it appears that with more knowledge about certain key mechanisms the results could be significantly improved on multiple sites and the hypothesis conclusively tested. A few of the important relationships that need further study follow.

Most general evapotranspiration functions are extremely complex and require many parameters that are not usually measured except at experimental sites. A general relationship with easily obtainable data inputs seems plausible with a semi-physical relationship such as evapotranspiration. Also some more general photosynthesis and respiration rate information would be helpful. Information currently available is species specific and it would be useful to know what factors are responsible for the observed variations in response trends.

If relationships such as the above can be developed then there is a good chance for simple general models; otherwise models will have to be site specific to be useful for management.

LITERATURE CITED

Bement, R. E. 1969. Stocking rate guide for beef production on blue grama range. J. Range Manage. 22:83-86.

Brown, L. W. 1974. Photosynthesis of two important grasses of the short grass prairie as affected by several ecological variables. Ph.D. Dissertation, Colo. State Univ., Ft. Collins. 191 p.

Connor, D. J., L. F. Brown and M. J. Trlica. 1974. Plant cover, light interception and photosynthesis of shortgrass prairie - a functional model. Photosynthetica (unpubl.).

Forrester, J. W. 1961. Industrial dynamics. MIT Press, Cambridge, Mass. 464 p.

Forrester, J. W. 1968. Principles of systems. Wright-Allen Press, Inc., Cambridge, Mass.

Hedrick, H. B. 1968. Bovine growth and composition. Miss. Univ. Agr. Exp. Sta. Bull. 928.

Heermann, D. F., M. D. Finkner and E. A. Hiler. 1971. Probability of sequences of wet and dry days for eleven western states and Texas. Colo. State Univ. Exp. Sta., Ft. Collins. Tech. Bull. 117.

Hume, I. D., R. J. Moir and M. Somers. 1970. Synthesis of microbial protein in the rumen. I. Influence of the level of nitrogen intake. Aust. J. Agr. Res. 21:283.

Innis, G. S. 1972. The second derivative and population modeling: another view. Ecology 53(4):720-723.

McGinnies, W. G. and J. F. Arnold. 1939. Relative water requirement of Arizona range plants. Ariz. Agr. Exp. Sta. Tech. Bull. 80:167-246.

Naylor, T. H., J. L. Balintfy, D. S. Burdick and K. Chu. 1966. Computer simulation techniques. John Wiley and Sons, Inc., New York. 340 p.

Rasmussen, J. L., G. Bertolin, and G. F. Almeyda. 1971. Grassland climatology of the Pawnee Grassland. US/IBP Tech. Rep. No. 127. Colo. State Univ., Ft. Collins. 79 p.

Rice, R. W., J. G. Morris, B. T. Maeda and R. L. Baldwin. 1974. Simulation of animal functions in models of production systems: ruminants on the range. Federation of Am. Soc. for Experimental Biology. Proc., Vol. 33, No. 2.

Singh, J. S. and D. C. Coleman. 1974. Evaluation of functional root biomass and translocation of photoassimilated carbon-14 in a shortgrass prairie ecosystem. *In* J. K. Marshall [ed.] The Belowground Ecosystem: A Synthesis of Plant Associated Processes. (unpubl.).

Todorovic, P. and D. A. Woolhiser. 1971. Stochastic model of daily rainfall. Proc. of the USDA-IASPS Symp. on Statistical Hydrology, Tucson, Ariz.

Whitman, W. D. 1974. Standing dead component in mixed grass prairie at the Dickinson Site. (unpubl.).

APPENDIX 1

Variable names, definitions and units

Name	Description	Unit
A1	The first parameter of a Fourier series - the mean value over the time interval	variable
A(I)	Array of Fourier sine coefficients for the terms remaining in the series I=1,2.	variable
AEN	Available energy for gain	kcal/day
AFAC	Grams microprotein growth per kcal digestible organic matter	g/kcal
ANOS(I)	Actual number of animals of class I, I=1,5	numbers
ANTOT	Total livestock numbers of all classes currently grazing	numbers
AREA	Size of grazing area - input in acres converted to m * m	m * m
ASZ	Proportion of mature size per livestock class	proportion
AVAIL	Forage availability	proportion
B	Grams of water needed to produce 1 gram of plant biomass	g H_2O/g plant
B(I)	Array of Fourier cosine coefficients - I=1,2	variable
BASE(I)	Price paid for stock when they are put on the pasture. I=1,5	$
CAP	Total rumen capacity	kg/day
CASH(I)	Amount invested in class I, I=1,5	$
CBM	Cool season green plant biomass	g/m^2
CH	Chlorophyll in green forage	mg/g
CMAX	Maximum cool season plant photosynthesis per day	g Ps/g/day
CP	Proportion of warm season plants for total growth	proportion

Name	Description	Unit
CRMAX	Maximum cool season plant respiration rate	g Rs/g/day
D	Percentage of green and dry forage that is digestible; second computation includes supplemental feed	proportion
DDAY	Julian day of the year, same as IDAY only a real variable	days
DG	Digestible component of all intake	kg
DIG	Total potential digestion, the sume of residual material digested from previous time step and current digested material	proportion
DIIT	Nitrogen content of animal diet	proportion
DMP	Microprotein growth	g/day
DMX	Green plant death loss	g lost/g/day
DOM	kcal contained in food eaten and digested	kcal/day
DPF	Effect of soil water on dead plant disappearance	proportion
DRDM	Change in rumen dry matter	kg/day
DRTD	Effect of temperature on disappearance of dead plant material	g lost/g/day
DS	Proportion of supplemental feed to total ingestion	proportion
DT	Solution time step for integration	days
DTFL	Time step between flow printouts	days
DTPL	Time step between plot value storage	days
DTPR	Time step between printouts	days
ECR	Effect of temperature on cool season respiration	proportion
EFAC	Proportion of volatile fatty acids to fermented energy	proportion
EO	Potential evapotranspiration	mm H_20/250mm soil

Name	Description	Unit
ESMC	Effect of soil water on cool season photosynthesis	proportion
ESMW	Effect of soil water on warm season photosynthesis	proportion
ET	Actual evapotranspiration	mm H_2O/250mm soil
ETC	Effect of temperature on cool season photosynthesis	proportion
ETODC	Effect of temperature on green plant death for cool season plants	proportion
ETODW	Effect of temperature on green plant death for warm season plants	proportion
ETW	Effect of temperature on warm season photosynthesis	proportion
EVAP	Evapotranspiration	mm/m^2/day
EWR	Effect of temperature on warm season respiration	proportion
EXR	Exit rate - combination of digested intake and constant passage rate, constrained to be less than .95	proportion
EYR	The rate at which eaten food passes through the digestive tract	proportion
FCAP	Soil water field capacity	mm H_2O/250mm soil
FCONS	Total forage consumption	g
FRDR	Fermentation digestion rate	proportion
G	Percentage of the forage that is green plant material	proportion
GAIN(I)	Animal weight gain by livestock class I, I=1,5	g/m^2/day
GAN	Animal weight gain	g/m^2/day
GC	Cool season plant growth returned to XFLWS from GROW	g/m^2
GFAC	kcal energy required for 1 gram of gain	kcal/gram
GN	Nitrogen content of green forage	proportion

Name	Description	Unit
GNIN(I)	Intake of live forage by livestock class I, I=1,5	g/day
GNN	Intake of live forage	g/day
GPLUSD	Total green and dry forage present	g/m^2
GR	Total cool and warm season growth used for constraining growth to a rate compatible with ET	$g/m^2/day$
GROW	A FORTRAN function subroutine	
GW	Warm season plant growth returned to XFLWS from GROW	g/m^2
H1	Amplitude of sine wave divided by two for precipitation	inches
H2	Sine wave displacement along time axis for precipitation	days
H3	Sine wave displacement along vertical axis for precipitation	inches
H4	Amplitude of sine wave divided by two for temperature	degrees
H5	Sine wave displacement along time axis for temperature	days
H6	Sine wave displacement along vertical axis for temperature	degrees
IDAY	Julian day of the year	days
IG	Ingestion rate	kg/day
IME	Time of simulation, same as TIME only an integer variable	days
IN(I)	Stocking day for livestock class I, I=1,5	day
INFD(I)	Beginning date of supplemental feed for livestock class I, I=1,5	day
INYR	First calendar year in which weather data is read from a tape (MODT = 1)	year
IYR	Year of simulation	year

Name	Description	Unit
JUDAY	Julian day read from weather tape	day
K	Counter which keeps track of days elapsed since growth initiation	days
KC	Counter to allow five days growth initiation for cool season plants	days
KW	Counter to allow five days growth initiation for warm season plants	days
KYR	Year of recorded data read from weather tape	years
LVWT(I)	Average individual animal weight. Input is in pounds, converted to grams in START. Total animal weight is found in each flow by multiplying by STKNO(I)/AREA	
LWT(I)	LVWT in pounds	pounds
MEN	Energy derived from microprotein	kcal/day
MIPRO	Microprotein level in the rumen	g
MIPROW	Microprotein per metabolic size	g/kg
MISS	Indicates when data is missing from weather tape	dimensionless
MKT(I)	Market day for livestock class I, I=1,5	day
MODP	Indicates method used for computing precipitation 1 = read from a tape 2 = compute from sine function 3 = compute using stochastic generator	dimensionless
MODT	Indicates method used for computing temperature 1 = read from a tape 2 = compute from sine function 3 = compute from sine function	dimensionless
MON	Month of the year	months
MPAS	Microprotein passage rate	g passed/g/day
MSZ	Mature size of livestock	g

Name	Description	Unit
N	Size of PRICE vector and PRCOV matrix	dimensionless
NAME	Station or ranch name for which weather parameters entered for stochastic generator	hollerith
NCLAS	The number of animal classes	numbers
NDIET	Percentage nitrogen in the diet, including supplement	proportion
NFOR	Percentage nitrogen in the forage	proportion
NFRDR	Proportion of intake digested per gram of microprotein, normal fermentation digestion rate	proportion
NGN	Nitrogen content of live forage	proportion
NITRO	A subroutine	
NOFD(I)	Ending date of supplemental feeding for livestock class I, I=1,5	days
NRA	1 if no rain on IDAY, 2 if rain occurred; can be used to lower temperature when rain occurs	dimensionless
NRM	A subroutine	
NT	Number of cycles used in Fourier series, generally 2 if PREGEN (Appendix 5) data used	dimensionless
O	Percentage of forage that is dry plant material	proportion
OMIN(I)	Intake of standing dead forage by livestock class I, I=1,5	g/day
OMN	Intake of standing dead forage	g/day
P	P(W/D) or P(W/W) depending on the occurrence of a storm the preceding day	proportion
PASFD	Passed food	kg/day
PEN	Energy obtained from passed food	kcal/day
PFL	Cumulative precipitation	mm
PH	Phosphorus content of forage	proportion

Name	Description	Unit
PHOS	Percentage phosphorus content in the diet	proportion
PP	Precipitation values read from weather tape	inches
PPT	A FORTRAN function subroutine	
PR	Protein content of live forage	proportion
PRC	Monthly livestock prices	\$/cwt
PRCOV(I)	Variance-covariance matrix for livestock class net prices: I, I=1,144	
PRICE(I)	Actual net price for livestock classes I, I=1,12 computed from PRMN(J) and PRCOV(I) for random normal deviates	\$/cwt
PRMN(I)	Mean monthly net price for livestock classes I=1,12	\$/cwt
PROT	Forage crude protein content	proportion
PT	Daily precipitation	mm
PW	Plant water	proportion
Q0	Probability of a dry day preceded by a dry day; converted to probability of a wet day preceded by a dry day	probability
Q1	Probability of a dry day; converted to probability of dry day preceded by a wet day; converted to a probability of a wet day preceded by a wet day	probability
QC	Scaling parameter for ESMC, a function of FCAP and WILT	dimensionless
QW	Scaling parameter for ESMW, a function of FCAP and WILT	dimensionless
R	Uniformly distributed random variable	proportion
RDM	Rumen dry matter	kg
RDMD	Rumen dry matter digestible	kg/day
RFL	Rumen fill	kg/day

Name	Description	Unit
RG	Relative growth rate of live forage	proportion
RGR	Relative growth rate of live forage	proportion
RUMEN	A subroutine	
S	Dummy variable computed in SUBROUTINE SER, used for Q0, Q1 and XLAM	dimensionless
SD	Nitrogen content of dry forage	proportion
SG	Supplemental feed ingested	kg/day
SITE	Station name for weather tape	hollerith
SOILA(SLA)	Input parameter for soil nutrient modifications	
SOILB(SLB)	Input parameter for soil nutrient modifications	
STKNO(I)	The number of animals in livestock class I, I = 1,5	numbers
SUPL	Proportion of supplemental feed to animal weight	proportion
SUPN	Nitrogen content of supplemental feed	proportion
SUPP(I)	Total price of supplemental feed for livestock class I, I=1,5	$
SUPPR(I)	Price of supplemental feed for livestock class I, I=1,5	$/1b
SUPRT(I)	Supplemental feed rate for livestock class I, I=1,5. Proportion of body weight	proportion
SUPWT(I)	Total supplemental feed given to livestock class I, I=1,5	lbs.
SW	Soil water	mm $H_2$0/250mm soil
T	Mean 5 day temperature	°C
T1	Variable to save the last good maximum daily temperature value in case a missing value is encountered when reading a weather tape	°F

Name	Description	Unit
T2	Variable to save the last good minimum daily temperature value in case a missing value is encountered when reading a weather tape	°F
TEMP	A FORTRAN function subroutine	
TEND	Ending time of simulation	days
THOLD	Total water holding capacity of soil	mm H_2O/250mm soil
TINIT	Parameter used to initialize random number generator	odd number
TMAX	Maximum daily temperature read from a weather tape	°F
TMIN	Minimum daily temperature read from a weather tape	°F
TOTAL	Sum of VALUE(I), I=1,5. Each value taken at MKT(I), I=1,5	
TP	Same as TP(I) in subroutine RECRD - current average daily temperature	°F
TP(I)	Stack of the past five days mean temperature used to compute five day average	°C
TR	Mean daily temperature	°C
TSTRT	Starting time of simulation	day
UNDFD	Undigested food	kg/day
V	Vitamin A in live forage	mg/gm
VALUE(I)	Amount received for livestock class I, I=1,5 on market day	$
VFA	Energy from volatile fatty acids	kcal/day
VITA	Vitamin A in live forage	mg/gm
VL75	Metabolic size, based on kg body weight for average sized animal from livestock class I, I=1,5	kg
WBM	Warm season green plant biomass	g/m^2

Name	Description	Unit
WILT	Soil wilting point	mm H_2O/250mm soil
WMAX	Maximum warm season plant photosynthesis per day	g Ps/g/day
WP	Percentage of total growth made up by warm season plants	proportion
WRMAX	Maximum warm season plant respiration per day	g Rs/g/day
X1011	Sum of the state variables X(10) and X(11) used to determine when phenological peak has been reached	g/m^2
X10T0	The amount of X(10) present at the previous time step	g/m^2
X11T0	The amount of X(11) present at the previous time step	g/m^2
XLAM	Daily lambda parameters generated by a Fourier series which when substituted into an experimental transformation yield storm size predictions	inches
ZZ(I)	Intermediate computation for price generation, I=1,12	dimensionless

State Variables

X(1)	Livestock class 1	g/m^2
X(2)	Livestock class 2	g/m^2
X(3)	Livestock class 3	g/m^2
X(4)	Livestock class 4	g/m^2
X(5)	Livestock class 5	g/m^2
X(10)	Warm season green plant biomass	g/m^2
X(11)	Cool season green plant biomass	g/m^2
X(12)	Standing dead plant biomass	g/m^2
X(21)	Soil water	mm H_2O/250mm soil

X(31)	Atmospheric water source	mm
X(32)	Water sink	mm
X(34)	Respired gases and litter	g/m^2
X(J)	J=41, 45. - VALUE(I), I=1,5	
X(J)	J=46, 50. - BASE(I), I=1,5	
X(J)	J=51, 55. - SUPP(I), I=1,5	
X(J)	J=56, 60. - CASH(I), I=1,5	
X(J)	J=61, 65. - SUPWT(I), I=1,5	
X(J)	J=66, 70. - LWT(I), I=1,5	
X(71)	SEASN	
X(72)	T	
X(73)	GN	
X(74)	PH	
X(75)	PR	
X(76)	CH	
X(77)	V	
X(78)	PW	
X(79)	FCONS	
X(80)	PFL	
X(J)	J=81, 85. - ANOS(I), I=1,5	

Function Subroutines

Name	Description	Argument List
ET	Calculates evapotranspiration	(SW, TR, FCAP, THOLD, WILT)
PPT	Calculates precipitation	(IDAY, MODP, TIME, TSTRT, NRA)
TEMP	Calculates temperature	(IDAY, MODT, NRA, TIME, TSTRT)

Subroutines

Name	Description	Argument List
GROW	Calculates plant growth	(WBM, CBM, SW, TR, SLA, SLB, GW, GC, FCAP, WILT)
NITRO	Calculates nitrogen content of the forage	(RGR, NGN, PH, PROT, CH, VITA, PW)
NRM	Stochastic price generator	(PRICE, PRCOV, PRMN, N, ZZ)
RECRD	Read a weather tape	(TIME, TSTRT, IDAY, INYR, TP, P)
RUMEN	Calculate forage intake and animal gain	(LVWT, SUPN, SUPL, OMN, GNN, GAN, MSZ, RDM, RFL, RDMD, MIPRO, VL75)
SER	Generates a Fourier series given Fourier coefficients	(NT, A1, A, B, S)

APPENDIX #2

FORTRAN listing of the RANGES grassland simulation model.

```
      PROGRAM RANGES
     1(INPUT,OUTPUT,PLTSV,TAPE5=INPUT,TAPE6=OUTPUT,TAPE3=PLTSV,TAPE4,
     2TAPE7)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C                                                                              C
C      XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    C
C       XX  XX      XX  XX      XX  XX      XX  XX      XX  XX      XX  XX     C
C        XXXX        XXXX        XXXX        XXXX        XXXX        XXXX      C
C     XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX   C
C     XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX   C
C        XXXX        XXXX        XXXX        XXXX        XXXX        XXXX      C
C       XX  XX      XX  XX      XX  XX      XX  XX      XX  XX      XX  XX     C
C      XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    C
C                                                                              C
C                                                                              C
C     XXXXXXXXX    XXXXXXXX   XX      XX   XXXXXXXX   XXXXXXXXXX   XXXXXXXX    C
C     XXXXXXXXXX  XXXXXXXXXX  XXX     XX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX   C
C     XX      XX  XX      XX  XXX     XX  XX      XX  XX          XX           C
C     XX      XX  XX      XX  XX X    XX  XX          XX          XX           C
C     XX      XX  XX      XX  XX XX   XX  XX          XX          XX           C
C     XXXXXXXXXX  XXXXXXXXXX  XX  X   XX  XX   XXXX   XXXXXX      XXXXXXXXX    C
C     XXXXXXXXX   XXXXXXXXXX  XX  X   XX  XX   XXXXX  XXXXXX       XXXXXXXXX   C
C     XX  XX      XX      XX  XX   X  XX  XX      XX  XX                  XX   C
C     XX   XX     XX      XX  XX   XX XX  XX      XX  XX                  XX   C
C     XX    XX    XX      XX  XX    XXXX  XX      XX  XX                  XX   C
C     XX     XX   XX      XX  XX    XXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX   C
C     XX      XX  XX      XX  XX     XXX   XXXXXXXX   XXXXXXXXXX   XXXXXXXX    C
C                                                                              C
C                                                                              C
C      XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    C
C       XX  XX      XX  XX      XX  XX      XX  XX      XX  XX      XX  XX     C
C        XXXX        XXXX        XXXX        XXXX        XXXX        XXXX      C
C     XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX   C
C     XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX   C
C        XXXX        XXXX        XXXX        XXXX        XXXX        XXXX      C
C       XX  XX      XX  XX      XX  XX      XX  XX      XX  XX      XX  XX     C
C      XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    C
C                                                                              C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C             THIS PROGRAM WAS ADAPTED BY BRAD J. GILBERT FROM A MODEL         C
C        DEVELOPED BY GEORGE S. INNIS AND RICHARD MISKIMINS.  THE INPUT        C
C        AND OUTPUT SUBROUTINES WERE WRITTEN BY JON GUSTAFSON.                 C
C                                                                              C
C                   *****************************************                  C
C                                                                              C
C             PURPOSE-   THE DEVELOPMENT OF A GRASSLAND ECOLOGICAL             C
C        SIMULATION MODEL COMPOSED OF A PLANT GROWTH SYSTEM, SOIL              C
C        WATER SYSTEM, ECONOMIC SYSTEM, AND A CONSUMER SYSTEM                  C
C        USED TO INVESTIGATE VARIOUS MARKETING STRATEGIES ALONG WITH           C
C        CATTLE MANAGEMENT PRACTICES AND TO SHOW THE EFFECTS OF THIS           C
C        ENTIRE COLLECTION ON INCOME.                                          C
C                                                                              C
C                   *****************************************                  C
C                                                                              C
C             PROGRAM UNITS-   (MAIN PROGRAM) - RANGES                         C
C        (SUBPROGRAM) - BLOCK DATA                                             C
C        (SUBROUTINES) - CYCLE, CYCL1, CYCL2, GROW, NITRO, NRM,                C
C                        RECRD, RUMEN, SER, START, XFLWS, XGRAPH,              C
C                        XOUT, XPLGEN, XPLOT, XPRINT, XRND, XUNPAK             C
C        (USER DEFINED FUNCTIONS) - ET, PPT, TEMP                              C
C        (CONTROL DATA UTILITY FUNCTIONS) - RANF                               C
C        (CONTROL DATA UTILITY SUBROUTINES) - RANSET                           C
C        (EXTERNAL ANSI FUNCTIONS) - ALOG, COS, EXP, SIN, SQRT                 C
C        (INTRINSIC ANSI FUNCTIONS) - ABS, AMAX1, AMIN1, INT, MOD              C
C             ABS FINDS THE ABSOLUTE VALUE OF A REAL ARGUMENT.                 C
C             ALOG FINDS THE NATURAL LOGARITHM OF A REAL VARIABLE.             C
C             AMAX1 CHOOSES THE LARGEST VALUE AMONG A LIST OF REAL             C
C        ARGUMENTS.  AS A RESULT THE NUMBER OF ARGUMENTS MUST BE TWO           C
C        OR GREATER.                                                           C
C             AMIN1 CHOOSES THE SMALLEST VALUE AMONG A LIST OF REAL            C
C        ARGUMENTS.  AS A RESULT THE NUMBER OF ARGUMENTS MUST BE TWO           C
C        OR GREATER.                                                           C
C             COS IS THE TRIGONOMETRIC COSINE WITH THE ARGUMENT                C
C        IN RADIANS AND TYPE REAL.                                             C
C             EXP IS AN EXPONENTIAL FUNCTION WITH A REAL ARGUMENT.             C
C             INT IS A FUNCTION THAT TRUNCATES THE VALUE OF A REAL             C
C        VARIABLE TO BECOME AN INTEGER VARIABLE.                               C
C             MOD COMPUTES THE REMAINDER FROM AN INTEGER DIVISION OF           C
C        INTEGER ARGUMENTS.                                                    C
C             RANF IS A RANDOM NUMBER GENERATOR WHICH RETURNS VALUES           C
C        UNIFORMLY DISTRIBUTED OVER THE INTERVAL BETWEEN ZERO AND ONE,         C
C        INCLUDING ZERO BUT NOT INCLUDING ONE.  THE ARGUMENT IS JUST           C
C        A DUMMY WHICH ENDS UP BEING IGNORED.                                  C
C             RANSET INITIALIZES A GENERATIVE VALUE FOR RANF.  THE             C
C        ARGUMENT IS A BIT PATTERN WITH BIT ZERO SET TO 1 (FORCED              C
C        ODD) AND BITS 59 THROUGH 48 SET TO 1717 OCTAL.                        C
C             SIN IS THE TRIGONOMETRIC SINE WITH THE ARGUMENT IN               C
C        RADIANS AND TYPE REAL.                                                C
C             SQRT CALCULATES THE SQUARE ROOT OF ITS REAL ARGUMENT.            C
C                                                                              C
```

```
C                   *****************************************           C
C                                                                       C
C             VARIABLE DEFINITIONS-   SOME VARIABLE DEFINITIONS APPEAR  C
C          BELOW WHILE OTHERS ARE DEFINED IN THE USERS GUIDE TO THE     C
C          RANGES GRASSLAND SIMULATION MODEL.  EXCEPT FOR SOME SUBSCRIPTS C
C          ALL THE VARIABLES ARE DEFINED.                               C
C          DUM      - REAL ARRAY CONTAINING THE SAME VALUES AS THE      C
C                       ARRAY (A) IN THE SUBROUTINE (XPRINT).           C
C          FLOW     - REAL VARIABLE CONTAINING A TEMPORARY STORAGE      C
C                       OF THE AMOUNT OF MATERIAL PER UNIT TIME ENTERING C
C                       OR LEAVING A LEVEL OR STATE VARIABLE.           C
C          IBLNK    - INTEGER VARIABLE CONTAINING A BLANK TO USE FOR    C
C                       CREATING THE PLOTS.                             C
C          ICHR     - INTEGER VARIABLE CONTAINING A TEMPORARY VALUE BEFORE C
C                       PLACING INTO THE OUTPUT CHARACTER STRING (IP).  C
C          IDASH    - INTEGER VARIABLE CONTAINING THE CHARACTER (-) TO  C
C                       USE FOR CREATING THE PLOTS.                     C
C          IEYEE    - INTEGER VARIABLE CONTAINING THE LETTER (I) TO USE C
C                       FOR CREATING THE PLOTS.                         C
C          IF       - INTEGER VARIABLE CONTAINING THE INDEX OF SOURCE   C
C                       COMPARTMENT IN THE SUBROUTINE (XUNPAK).         C
C          INFO     - INTEGER VARIABLE CONTAINING THE COMPUTER WORD     C
C                       PASSED TO THE SUBROUTINE (XUNPAK) FOR PROCESSING. C
C          IP       - INTEGER ARRAY CONTAINING THE OUTPUT CHARACTER     C
C                       STRING WITH BLANKS AND GRAPHICAL REFERENCE LINES C
C                       FOR THE PLOTTING OF STATE VARIABLES IN THE      C
C                       SUBROUTINE (XGRAPH).                            C
C          IP       - INTEGER VARIABLE CONTAINING A FLOW PRINT FLAG IN  C
C                       THE MAIN PROGRAM (RANGE4) AND THE SUBROUTINE    C
C                       (XUNPAK).                                       C
C          IQUAL    - INTEGER VARIABLE CONTAINING THE CHARACTER (=)     C
C                       TO USE FOR CREATING THE PLOTS.                  C
C          ISTOP    - INTEGER VARIABLE CONTAINING A FLAG TO SIGNAL THE  C
C                       LAST RECORD OF STORED PLOTTING DATA.            C
C          IT       - INTEGER VARIABLE CONTAINING THE INDEX OF          C
C                       DESTINATION COMPARTMENT IN THE SUBROUTINE (XUNPAK). C
C          IV       - INTEGER VARIABLE CONTAINING THE WORD IN THE       C
C                       SUBROUTINE (XUNPAK) TO BE UNPACKED.             C
C          IVERB    - INTEGER VARIABLE CONTAINING THE COMMAND VERBS     C
C                       INPUT BY THE CARD READER FOR OUTPUT CONTROL.    C
C          IZ       - INTEGER VARIABLE CONTAINING THE TRUNCATED VALUE   C
C                       FOR THE VARIABLE (ZI) USED IN SCALING THE Y AXIS C
C                       OF PLOTS GENERATED.                             C
C          I1       - INTEGER VARIABLE CONTAINING AN INDEX OF SOURCE    C
C                       COMPARTMENT.                                    C
C          I2       - INTEGER VARIABLE CONTAINING AN INDEX OF DESTINATION C
C                       COMPARTMENT.                                    C
C          JCHAR    - INTEGER ARRAY CONTAINING THE POSSIBLE NUMERIC VALUES C
C                       REPRESENTING THE VARIABLES PLOTTED.             C
C          JNT      - INTEGER VARIABLE CONTAINING A SUBSCRIPT VALUE FOR C
C                       PRINTING THE X AXIS INDICES ON THE PLOTS        C
C                       GENERATED.                                      C
C          JP       - INTEGER VARIABLE CONTAINING A TEMPORARY VALUE FROM C
C                       THE OUTPUT CHARACTER STRING (IP).               C
C          J1       - INTEGER VARIABLE CONTAINING THE INITIAL SUBSCRIPT C
C                       IN LISTING THE STATE VARIABLES FROM THE ARRAY (VAL) C
C                       FOR EACH LINE.                                  C
C          J2       - INTEGER VARIABLE CONTAINING THE ENDING SUBSCRIPT  C
C                       IN LISTING THE STATE VARIABLES FROM THE ARRAY (VAL) C
C                       FOR EACH LINE.                                  C
C          KEY      - INTEGER ARRAY CONTAINING THE POSSIBLE COMMAND     C
C                       VERBS TO CHECK AGAINST THE VARIABLE (IVERB).    C
C          KNT      - INTEGER VARIABLE CONTAINING A VERTICAL CONTROL    C
C                       FACTOR FOR BUILDING THE FRAME OF THE GRAPH USED C
C                       TO PLOT THE STATE VARIABLES.                    C
C          LOC      - INTEGER VARIABLE CONTAINING THE LOCATION POINTER  C
C                       IN THE STATE VARIABLE ARRAY (Z) FOR PLOTTING    C
C                       PURPOSES.                                       C
C          M        - INTEGER VARIABLE CONTAINING A TEMPORARY STORAGE   C
C                       VALUE USED TO ADJUST THE VALUE OF ARRAYS (IN    C
C                       AND MK1).                                       C
C          NFL      - INTEGER VARIABLE CONTAINING A SPECIAL FLAG TO     C
C                       DETERMINE FLOW PRINTOUTS BY THE TIME STEP BETWEEN C
C                       THE FLOW PRINTOUTS.                             C
C          NKNT     - INTEGER VARIABLE CONTAINING THE NUMBER OF STATE   C
C                       VARIABLES PER LINE IN THE LISTING.              C
C          NLINE    - INTEGER VARIABLE CONTAINING THE NUMBER OF LINES   C
C                       MAKING A LIST OF STATE VARIABLES WITH           C
C                       FOUR VARIABLES PER LINE.                        C
C          NN       - INTEGER VARIABLE CONTAINING THE NUMBER OF STATE   C
C                       VARIABLES TO BE PLOTTED.                        C
C          NPL      - INTEGER VARIABLE CONTAINING A SPECIAL FLAG TO     C
C                       DETERMINE PLOTTING TIME IN THE MAIN PROGRAM (RANGE4) C
C                       OR THE PLOT NUMBER IN THE SUBROUTINE (XGRAPH).  C
C          NPR      - INTEGER VARIABLE CONTAINING A SPECIAL FLAG TO     C
C                       DETERMINE TIME FOR PRINTOUTS.                   C
C          NUM      - INTEGER ARRAY CONTAINING THE INDICES OF THE       C
C                       STATE VARIABLES REQUESTED FOR PRINTING ON THE   C
C                       PRINT OUTPUT CONTROL CARD.                      C
C          NVARS    - INTEGER VARIABLE CONTAINING THE NUMBER OF VARIABLES C
C                       PER PLOT.                                       C
C          NVR      - INTEGER VARIABLE CONTAINING THE NUMBER OF VARIABLES C
C                       PER PLOT.                                       C
C          NXLOC    - INTEGER MATRIX CONTAINING THE LOCATION OF EACH    C
C                       VARIABLE IN THE LIST OF STATE VARIABLES FOR     C
C                       EACH PLOT.                                      .
```

```
                                                                               C
C         NXPL      - INTEGER ARRAY CONTAINING A LIST OF STATE VARIABLES      C
C                     TO BE PLOTTED.                                           C
C         P1        - REAL VARIABLE CONTAINING EITHER CURRENT DAY WEATHER      C
C                     INFORMATION OR PAST GOOD WEATHER DATA.                   C
C         Q         - REAL VARIABLE CONTAINING A TEMPORARY VALUE ADDED TO      C
C                     OR SUBTRACTED FROM A STATE VARIABLE.                     C
C         RNVAR     - REAL VARIABLE CONTAINING THE TRUNCATED VALUE OF          C
C                     THE VARIABLE (VAR) USED FOR SCALING PURPOSES.            C
C         RNZMAX    - REAL VARIABLE CONTAINING THE EXTREME MAXIMUM             C
C                     VALUE FOR THE GENERATED PLOTS.                           C
C         RNZMIN    - REAL VARIABLE CONTAINING THE EXTREME MINIMUM             C
C                     VALUE FOR THE GENERATED PLOTS.                           C
C         SUM       - REAL VARIABLE CONTAINING A SUMMATION VALUE USED          C
C                     IN THE EXPANSION OF FOURIER SERIES.                      C
C         TIME      - REAL VARIABLE CONTAINING THE CURRENT VALUE OF            C
C                     SIMULATED TIME.                                          C
C         TIMEFL    - REAL VARIABLE CONTAINING THE TIME VALUE USED AS A        C
C                     FLAG TO SIGNAL PRINTING THE FLOWS.                       C
C         TIMEPL    - REAL VARIABLE CONTAINING THE TIME VALUE USED AS A        C
C                     FLAG TO SIGNAL PLOTTING.                                 C
C         TIMEPR    - REAL VARIABLE CONTAINING THE TIME VALUE USED AS A        C
C                     FLAG TO SIGNAL PRINTING OF THE STATE VARIABLES.          C
C         TINIT     - REAL VARIABLE CONTAINING THE GENERATIVE VALUE FOR        C
C                     THE RANDOM NUMBER GENERATOR.                             C
C         TM        - REAL VARIABLE CONTAINING THE TEMPORARY VALUE FOR         C
C                     THE NEXT TIME STEP USED IN PRINTING THE FLOWS.           C
C         VAL       - REAL ARRAY CONTAINING THE WORKING STORAGE VALUES         C
C                     USED IN OUTPUT GENERATION.                               C
C         VAR       - REAL VARIABLE CONTAINING THE RANGE OF THE                C
C                     MINIMUM AND MAXIMUM VALUES IN SCALING FOR THE            C
C                     PLOTS.                                                   C
C         XF        - REAL ARRAY CONTAINING THE CURRENT VALUES OF FLOWS.       C
C         XFR       - INTEGER ARRAY CONTAINING THE FLOW REFERENCE              C
C                     TABLE WHERE THE COMPARTMENTAL INDICES OF THE             C
C                     FLOWS ARE STORED ACCORDING TO THE FOLLOWING              C
C                     FORMULA    IP*10000  FLOW PRINT FLAG,                    C
C                                  I1*100  INDEX OF SOURCE COMPARTMENT.        C
C                                      I2  INDEX OF DESTINATION                C
C                                          COMPARTMENT.                        C
C         XI        - REAL VARIABLE CONTAINING THE REAL VALUE OF A             C
C                     VARIABLE SUBSCRIPT (I) USED IN THE EXPANSION             C
C                     OF THE FOURIER SERIES.                                   C
C         XINC      - REAL VARIABLE CONTAINING THE INCREMENTAL VALUE           C
C                     BETWEEN INDICES OF THE X AXIS ON THE PLOTS GENERATED.    C
C         XLINE     - REAL ARRAY CONTAINING THE INDICES FOR THE X AXIS         C
C                     ON THE PLOTS GENERATED.                                  C
C         XMAX      - REAL VARIABLE CONTAINING A MAXIMUM VALUE FOR THE         C
C                     X AXIS ON THE PLOTS GENERATED.                           C
C         XMIN      - REAL VARIABLE CONTAINING A MINIMUM VALUE FOR THE         C
C                     X AXIS ON THE PLOTS GENERATED.                           C
C         XN        - INTEGER VARIABLE CONTAINING A COUNTER USED TO            C
C                     ENTER FLOWS INTO THE ARRAY (XF) IN THE SUBROUTINE        C
C                     (XFLWS) OR REAL VARIABLE CONTAINING THE REAL             C
C                     VALUE OF VARIABLE SUBSCRIPT (N) USED IN THE              C
C                     EXPANSION OF THE FOURIER SERIES IN THE SUBROUTINE        C
C                     (SER).                                                   C
C         XNF       - INTEGER VARIABLE CONTAINING THE NUMBER OF FLOWS          C
C                     WITH A MAXIMUM OF 300.                                   C
C         XNLOC     - INTEGER MATRIX CONTAINING THE LOCATION OF EACH           C
C                     VARIABLE IN THE LIST OF STATE VARIABLES                  C
C                     (I.E. XPL(K)) IN EACH PLOT.                              C
C         XNPL      - INTEGER VARIABLE CONTAINING THE NUMBER OF STATE          C
C                     VARIABLES TO BE PLOTTED.                                 C
C         XNPLT     - INTEGER VARIABLE CONTAINING THE NUMBER OF PLOTS          C
C                     TO BE GENERATED WITH A MAXIMUM OF 20.                    C
C         XNPR      - INTEGER VARIABLE CONTAINING THE NUMBER OF STATE          C
C                     VARIABLES TO BE PRINTED.                                 C
C         XNST      - INTEGER VARIABLE CONTAINING THE NUMBER OF STATE          C
C                     VARIABLES.                                               C
C         XNVRS     - INTEGER ARRAY CONTAINING THE NUMBER OF VARIABLES         C
C                     PER PLOT WITH A MAXIMUM OF 5.                            C
C         XPL       - INTEGER ARRAY CONTAINING A LIST OF STATE VARIABLES      C
C                     TO BE PLOTTED.                                           C
C         XPR       - INTEGER ARRAY CONTAINING A LIST OF STATE VARIABLES      C
C                     TO BE PRINTED.                                           C
C         XST       - INTEGER ARRAY CONTAINING A LIST OF STATE VARIABLE       C
C                     INDICES.                                                 C
C         XTST      - REAL VARIABLE CONTAINING A TEMPORARY VALUE USED          C
C                     TO CALCULATE INDICES OF THE X AXIS ON THE PLOTS          C
C                     GENERATED.                                               C
C         YINC      - REAL VARIABLE CONTAINING THE INCREMENTAL VALUE           C
C                     BETWEEN INDICES OF THE Y AXIS ON THE PLOTS GENERATED.    C
C         YLINE     - REAL ARRAY CONTAINING THE INDICES FOR THE Y AXIS ON      C
C                     THE PLOTS GENERATED.                                     C
C         YMAX      - REAL VARIABLE CONTAINING A MAXIMUM VALUE FOR THE Y       C
C                     AXIS ON THE PLOTS GENERATED.                             C
C         YMIN      - REAL VARIABLE CONTAINING A MINIMUM VALUE FOR THE Y       C
C                     AXIS ON THE PLOTS GENERATED.                             C
C         YTST      - REAL VARIABLE CONTAINING A TEMPORARY VALUE USED TO       C
C                     CALCULATE INDICES OF THE Y AXIS ON THE PLOTS             C
C                     GENERATED.                                               C
C         Z         - REAL VARIABLE CONTAINING THE SCALING FACTOR FOR          C
C                     SETTING THE PLOT VALUES IN THE SUBROUTINE (XRND).        C
C         Z         - REAL ARRAY CONTAINING THE STATE VARIABLES TO BE          C
C                     PLOTTED IN THE SUBROUTINE (XPLGEN).                      C
```

```
C        ZI      - REAL VARIABLE CONTAINING THE LOCATION OF THE                C
C                  PLOTTING CHARACTER IN THE OUTPUT STRING ACCORDING           C
C                  TO THE VALUE OF THE DEPENDENT VARIABLE (ZZ) BEFORE          C
C                  SCALING.                                                    C
C        ZJ      - REAL VARIABLE CONTAINING THE VALUE FOR THE INTEGER          C
C                  VARIABLE (IZ) USED IN SCALING THE Y AXIS OF PLOTS           C
C                  GENERATED.                                                  C
C        ZMAX    - REAL VARIABLE CONTAINING THE MAXIMUM GRAPHICAL              C
C                  SCALING VALUE IN THE SUBROUTINE (XRND).                     C
C        ZMIN    - REAL VARIABLE CONTAINING THE MINIMUM GRAPHICAL              C
C                  SCALING VALUE IN THE SUBROUTINE (XRND).                     C
C        ZRND    - REAL VARIABLE CONTAINING THE TRUNCATED VALUE FOR            C
C                  THE VARIABLE (Z) IN THE SUBROUTINE (XRND).                  C
C        ZZ      - REAL VARIABLE CONTAINING A TEMPORARY VALUE STORING          C
C                  THE STATE VARIABLE TO BE PLOTTED.                           C
C                                                                              C
C                   *****************************************                  C
C                                                                              C
C             PROGRAM FILES-   TAPE5 REFERENCED BY THE VARIABLE (U1)           C
C        IS EQUIVALENCED TO THE INPUT FILE, IN THIS CASE THE CARD              C
C        READER.  TAPE6 REFERENCED BY THE VARIABLE (U2) IS EQUIVALENCED        C
C        TO THE OUTPUT FILE, IN THIS CASE THE LINE PRINTER.  TAPE3             C
C        REFERENCED BY THE VARIABLE (U3) IS EQUIVALENCED TO A DISK             C
C        FILE THAT IS USED TO STORE PLOT GENERATION DATA.  TAPE4 AND           C
C        TAPE7 REFERENCED BY THE VARIABLE (LU) CAN BE EQUIVALENCED TO          C
C        DISK FILES USED AS ALTERNATIVE STORAGE SPACE.  ALL THESE              C
C        VARIABLES ARE INITIALIZED IN THE BLOCK DATA SUBPROGRAM.               C
C                                                                              C
C                   *****************************************                  C
C                                                                              C
C             FORMAT OF DATA ENTRIES ARE ALSO DESCRIBED IN THE                 C
C        USERS GUIDE TO THE RANGES GRASSLAND SIMULATION MODEL.                 C
C                                                                              C
C                   *****************************************                  C
C                                                                              C
C             REMARKS-   ALL CODING WITHIN THIS PROGRAM HAS BEEN               C
C        CONFORMED TO STANDARD ANSI FORTRAN EXCEPT THE FOLLOWING ITEMS.        C
C        THE PROGRAM CARD WHICH IS A PART OF CDC SCOPE 3.3 OPERATING           C
C        SYSTEM, SOME MIXED MODE OPERATIONS, AND THE CDC UTILITY               C
C        FUNCTION AND SUBROUTINE.  THE FUNCTION (RANF) IS REFERENCED           C
C        IN THE SUBROUTINE (NRM) LINES 18 AND 19, AND THE FUNCTION             C
C        (PPI) LINES 91 AND 94.  THE SUBROUTINE (RANSET) IS REFERENCED         C
C        IN THE SUBROUTINE (START) LINE 66.                                    C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       RANGE001
      INTEGER I,IDAY,INFO,INYR,IP,IYR,I1,I2,JUDAY,K,LU,MON              RANGE002
      INTEGER NFL,NPL,NPR,U1,U2,U3,XFR,XNF,XNLOC,XNPL,XNPLT             RANGE003
      INTEGER XNPR,XNST,XNVRS,XPL,XPR,XST                               RANGE004
      REAL DT,DTFL,DTPL,DTPR,U,TEND,TIME,TIMEFL,TIMEPL,TIMEPR           RANGE005
      REAL TM,TSTRT,VAL(300),X,XF,X10TO,X11TO                           RANGE006
C                                                                       RANGE007
      COMMON /TO/ DTFL,DTPL,DTPR,TEND                                   RANGE008
      COMMON /TT/ IDAY,INYR,IYR,JUDAY,K,MON                             RANGE009
      COMMON /TX/ DT,TIME,TSTRT,X(99),XNF,X10TO,X11TO                   RANGE010
      COMMON /XSYS/ XF(300),XFR(300),XNLOC(20,5),XNPL,XNPLT,XNPR,XNST,XNRANGE011
     1VRS(20),XPL(99),XPR(99),XST(99)                                   RANGE012
      COMMON /XUNT/ LU,U1,U2,U3                                         RANGE013
C                                                                       RANGE014
C . . . . . . INITIALIZE SIMULATION CONTROL VARIABLES.                  RANGE015
C                                                                       RANGE016
      DO 102 I=33,300                                                   RANGE017
         J=I+30                                                         RANGE018
         IF (J.GT.99) GO TO 101                                         RANGE019
         XST(J)=0                                                       RANGE020
 101     XFR(I)=0                                                       RANGE021
         ITEM=I-32                                                      RANGE022
         IF (ITEM.GT.99) GO TO 102                                      RANGE023
         X(ITEM)=0.0                                                    RANGE024
 102  CONTINUE                                                          RANGE025
      DO 103 I=1,300                                                    RANGE026
         XF(I)=0.0                                                      RANGE027
 103  CONTINUE                                                          RANGE028
      CALL XOUT                                                         RANGE029
      CALL START                                                        RANGE030
C                                                                       RANGE031
C . . . . . . CHECK AND INITIALIZE MORE SIMULATION CONTROL VARIABLES.   RANGE032
C                                                                       RANGE033
      IF (TEND.LE.TSTRT) GO TO 115                                      RANGE034
      IF (DT.LE.0.) GO TO 116                                           RANGE035
      IF (LU.GT.0) REWIND LU                                            RANGE036
      TIME=TSTRT                                                        RANGE037
      IF (XNPR.GT.0.AND.DTPR.LE.0.) GO TO 117                           RANGE038
      NPR=0                                                             RANGE039
      TIMEPR=0.                                                         RANGE040
      IF (XNPR.LE.0) GO TO 104                                          RANGE041
      NPR=1                                                             RANGE042
      TIMEPR=TSTRT                                                      RANGE043
 104  NPL=0                                                             RANGE044
      TIMEPL=0.                                                         RANGE045
      IF (XNPL.LE.0) GO TO 105                                          RANGE046
      NPL=1                                                             RANGE047
      TIMEPL=TSTRT                                                      RANGE048
      DTPL=(TEND-TSTRT)/99.                                             RANGE049
      IF (DTPL.LT.DT) DTPL=DT                                           RANGE050
```

```
  105 NFL=0
      TIMEFL=0.
      IF (DTFL.LE.0.) GO TO 106
      NFL=1
      TIMEFL=TIME
C
C . . . . . . OUTPUT SIMULATION CHARACTERISTICS AND CONTROL VARIABLES
C             AND INITIAL VALUES OF STATE VARIABLES.
C
  106 WRITE (U2,118) XNST,XNPR,XNPL,XNF
      WRITE (U2,119) TSTRT,TEND,DT,DTPR,DTPL,DTFL
      CALL XPRINT (VAL,XST,XNST)
      WRITE (U2,121)
C
C . . . . . . ENTER THE SIMULATION LOOP.
C
  107 CALL CYCLE
C
C . . . . . . GENERATE OUTPUT IF REQUESTED.
C
      IF (NPR.LE.0.OR.TIMEPR.GT.TIME) GO TO 108
      CALL XPRINT (VAL,XPR,XNPR)
      TIMEPR=TIMEPR+DTPR
  108 IF (NPL.LE.0.OR.TIMEPL.GT.TIME) GO TO 109
      CALL XPLOT (VAL,0)
      TIMEPL=TIMEPL+DTPL
C
C . . . . . . COMPUTE THE FLOWS.
C
  109 CALL XFLWS
      IF (NFL.LE.0.OR.TIMEFL.GT.TIME) GO TO 110
      TM=TIME+DT
C
C . . . . . . UPDATE THE STATE VARIABLES AND PRINT FLOWS IF REQUESTED.
C
  110 X11TO=X(11)
      X10TO=X(10)
      DO 111 I=1,XNF
          INFO=XFR(I)
          CALL XUNPAK (INFO,IP,I1,I2)
          Q=XF(I)*DT
          X(I1)=X(I1)-Q
          X(I2)=X(I2)+Q
          IF (NFL.LE.0.OR.TIMEFL.GT.TIME) GO TO 111
          IF (IP.LE.0) GO TO 111
          WRITE (U2,120) I1,I2,XF(I)
  111 CONTINUE
      IF (NFL.LE.0.OR.TIMEFL.GT.TIME) GO TO 112
      TIMEFL=TIMEFL+DTFL
  112 TIME=TIME+DT
      IF (INT(TIME).GT.INT(TEND)) GO TO 113
      GO TO 107
C
C . . . . . . SIMULATION IS COMPLETE, PERFORM END PROCESSING AND PLOT
C             GENERATION IF REQUESTED.
C
  113 IF (NPL.LE.0) GO TO 114
      CALL XPLOT (VAL,1)
      CALL XPLGEN
  114 STOP
C
C . . . . . . GENERATE DIAGNOSTICS FOR ILLEGAL CONDITIONS.
C
  115 WRITE (U2,122)
      WRITE (U2,123) TSTRT,TEND
      STOP
  116 WRITE (U2,122)
      WRITE (U2,124) DT
      STOP
  117 WRITE (U2,122)
      WRITE (U2,125) DTPR
      STOP
C
C . . . . . . FORMATS USED IN THIS PROGRAM.
C
  118 FORMAT (7H1RANGES,15X,18HINITIAL CONDITIONS///13X,37HNO. OF STATE
     1VARIABLES................,I2/16X,34HREQUESTED FOR PRINT............
     2...,I2/16X,34HREQUESTED FOR PLOT................,I2//13X,36HNO. OF
     3 FLOWS.........................,I3)
  119 FORMAT (///20X,20HTSTRT ..............,F12.1/20X,20HTEND..........
     1......,F12.1/20X,20HDT.................,F12.1/20X,20HDTPR........
     2........,F12.1/20X,20HDTPL.................,F12.1/20X,20HDTFL......
     3..........,F12.1//)
  120 FORMAT (10X,1H(,I2,1H,,I2,4H) = ,F12.5)
  121 FORMAT (7H1RANGES,15X,18HSIMULATION RESULTS//)
  122 FORMAT (42H0*****ILLEGAL CONDITION - PARAMETER VALUES)
  123 FORMAT (13H0     TSTRT (,F12.1,12H) .GE. TEND(,F12.1,1H))
  124 FORMAT (9H0     DT(,F12.1,9H) .LE. 0.)
  125 FORMAT (1H0,5X,38HPRINT REQUESTS ENCOUNTERED WHILE DTPR(,F12.1,9H)
     1 .LE. 0.)
C
      END
```

```
      BLOCK DATA                                                        BLKDT001
C                                                                       BLKDT002
C . . . . . . THIS SUBPROGRAM (BLOCK DATA) ENTERS INITIAL VALUES INTO    BLKDT003
C        VARIABLES IN THE LABELED COMMON BLOCKS PRIOR TO PROGRAM        BLKDT004
C        EXECUTION.                                                     BLKDT005
C                                                                       BLKDT006
      INTEGER IBLNK,IDASH,IFYEE,IN,INFD,IQUAL,JCHAR,KEY,LU              BLKDT007
      INTEGER MKT,MODP,MODT,NCLAS,NOFD,U1,U2,U3,XFR,XNF                 BLKDT008
      INTEGER XNLOC,XNPL,XNPLT,XNPR,XNST,XNVRS,XPL,XPR,XST              BLKDT009
      REAL ANOS,ANTOT,AREA,CH,CMAX,CRMAX,DIIT,DT,FCONS,GAIN             BLKDT010
      REAL GN,GNTN,LVWT,LWT,MSZ,OMIN,PFL,PH,PR,PRC,PT,PW                BLKDT011
      REAL RG,SD,STKNO,SUPL,SUPN,SUPPR,SUPRT,T,TIME,TOTAL,TP            BLKDT012
      REAL TR,TSTRT,V,WMAX,WRMAX,X,XF,X10T0,X11T0                       BLKDT013
C                                                                       BLKDT014
      COMMON /CAT/ ANOS(5),ANTOT,AREA,DIIT,FCONS,GAIN(5),GNTN(5),IN(5),IBLKDT015
     1NFD(5),LVWT(5),LWT(5),MKT(5),MSZ(5),NCLAS,NOFD(5),OMIN(5),STKNO(5)BLKDT016
     2,SUPL(5),SUPN(5),SUPPR(5),SUPRT(5)                                BLKDT017
      COMMON /RESP/ CMAX,CRMAX,WMAX,WRMAX                               BLKDT018
      COMMON /TAP/ MODP,MODT,PFL,PT,T,TP(6),TR                          BLKDT019
      COMMON /TX/ DT,TIME,TSTRT,X(99),XNF,X10T0,X11T0                   BLKDT020
      COMMON /UX/ CH,GN,PH,PR,PRC,PW,RG,SD,TOTAL,V                      BLKDT021
      COMMON /XFIG/ KEY(5),IBLNK,IDASH,IFYEE,IQUAL,JCHAR(5)             BLKDT022
      COMMON /XSYS/ XF(300),XFR(300),XNLOC(20,5),XNPL,XNPLT,XNPR,XNST,XNBLKDT023
     1VRS(20),XPL(99),XPR(99),XST(99)                                   BLKDT024
      COMMON /XUNT/ LU,U1,U2,U3                                         BLKDT025
C                                                                       BLKDT026
C . . . . . . INITIALIZE SIMULATION CONTROL VARIABLES.                  BLKDT027
C                                                                       BLKDT028
      DATA XNF/32/                                                      BLKDT029
      DATA XFR(1)/3121/,XFR(2)/2132/,XFR(3)/3310/,XFR(4)/3311/          BLKDT030
      DATA XFR(5)/1012/,XFR(6)/1112/,XFR(7)/1234/,XFR(8)/1001/          BLKDT031
      DATA XFR(9)/1101/,XFR(10)/1201/,XFR(11)/3501/,XFR(12)/0134/       BLKDT032
      DATA XFR(13)/1002/,XFR(14)/1102/,XFR(15)/1202/,XFR(16)/3502/      BLKDT033
      DATA XFR(17)/0234/,XFR(18)/1003/,XFR(19)/1103/,XFR(20)/1203/      BLKDT034
      DATA XFR(21)/3503/,XFR(22)/0334/,XFR(23)/1004/,XFR(24)/1104/      BLKDT035
      DATA XFR(25)/1204/,XFR(26)/3504/,XFR(27)/0434/,XFR(28)/1005/      BLKDT036
      DATA XFR(29)/1105/,XFR(30)/1205/,XFR(31)/3505/,XFR(32)/0534/      BLKDT037
      DATA XNST/62/                                                     BLKDT038
      DATA XST(1)/01/,XST(2)/02/,XST(3)/03/,XST(4)/04/,XST(5)/05/       BLKDT039
      DATA XST(6)/10/,XST(7)/11/,XST(8)/12/,XST(9)/21/,XST(10)/31/      BLKDT040
      DATA XST(11)/32/,XST(12)/33/,XST(13)/34/,XST(14)/35/,XST(15)/41/  BLKDT041
      DATA XST(16)/42/,XST(17)/43/,XST(18)/44/,XST(19)/45/,XST(20)/46/  BLKDT042
      DATA XST(21)/47/,XST(22)/48/,XST(23)/49/,XST(24)/50/,XST(25)/51/  BLKDT043
      DATA XST(26)/52/,XST(27)/53/,XST(28)/54/,XST(29)/55/,XST(30)/56/  BLKDT044
      DATA XST(31)/57/,XST(32)/58/,XST(33)/59/,XST(34)/60/,XST(35)/61/  BLKDT045
      DATA XST(36)/62/,XST(37)/63/,XST(38)/64/,XST(39)/65/,XST(40)/66/  BLKDT046
      DATA XST(41)/67/,XST(42)/68/,XST(43)/69/,XST(44)/70/,XST(45)/71/  BLKDT047
      DATA XST(46)/72/,XST(47)/73/,XST(48)/74/,XST(49)/75/,XST(50)/76/  BLKDT048
      DATA XST(51)/77/,XST(52)/78/,XST(53)/79/,XST(54)/80/,XST(55)/81/  BLKDT049
      DATA XST(56)/82/,XST(57)/83/,XST(58)/84/,XST(59)/85/,XST(60)/87/  BLKDT050
      DATA XST(61)/88/,XST(62)/99/                                      BLKDT051
      DATA XNPR/0/,XNPL/0/,XNPLT/0/,X11T0/0./,X10T0/0./                 BLKDT052
      DATA KEY(1)/4HEND./,KEY(2)/4HPRIN/,KEY(3)/4HFLOW/,KEY(4)/4HPLOT/  BLKDT053
      DATA KEY(5)/4H    /                                               BLKDT054
      DATA JCHAR(1)/1H1/,JCHAR(2)/1H2/,JCHAR(3)/1H3/,JCHAR(4)/1H4/      BLKDT055
      DATA JCHAR(5)/1H5/                                                BLKDT056
      DATA IBLNK/1H /,IDASH/1H-/,IFYEE/1HI/,IQUAL/1H=/                  BLKDT057
C                                                                       BLKDT058
C     . . . . INITIALIZE NON-READ VARIABLES.                            BLKDT059
C                                                                       BLKDT060
      DATA PRC/25.0/,RG/-.04/,PFL/0./,GN/1.0/,FCONS/0./,PH/.19/         BLKDT061
      DATA PR/6.25/,CH/0./,V/0./,PW/10./,DIIT/0./,T/0./,TOTAL/0./       BLKDT062
      DATA SD/.06/                                                      BLKDT063
      DATA WMAX/.39/,CMAX/.41/                                          BLKDT064
      DATA WRMAX/.13/,CRMAX/.18/                                        BLKDT065
      DATA U1/5/,U2/6/,U3/3/,LU/4/                                      BLKDT066
C                                                                       BLKDT067
      END                                                               BLKDT068
```

```
      SUBROUTINE CYCLE
C
C . . . . . . THIS SUBROUTINE (CYCLE) IS CALLED BY THE MAIN PROGRAM
C           (RANGES) TO RECALCULATE THE VALUES OF IMPORTANT VARIABLES
C           BEFORE COMPUTATION OF FLOWS AT EACH TIME STEP DURING
C           SIMULATION.
C
      INTEGER IDAY,IN,INFU,INYR,IYR,J,JUDAY,K,MKT,MODP
      INTEGER MODT,MON,NCLAS,NOFU,XNF
      REAL ANOS,ANTOT,AREA,BASE,CASH,CH,DIIT,DT,FCONS,GAIN
      REAL GN,GNIN,LVWT,LWT,MSZ,OMIN,PFL,PH,PR,PRC,PRCOV
      REAL PRICE,PRMN,PT,PW,RG,SD,STKND,SUPL,SUPN,SUPP
      REAL SUPPR,SUPRT,SUPWT,T,TIME,TOTAL,TP,TR,TSTRT,V
      REAL VALUE,X,X10T0,X11T0,ZZ
C
      COMMON /CAT/ ANOS(5),ANTOT,AREA,DIIT,FCONS,GAIN(5),GNIN(5),IN(5),I
     1NFU(5),LVWT(5),LWT(5),MKT(5),MSZ(5),NCLAS,NOFU(5),OMIN(5),STKND(5)
     2,SUPL(5),SUPN(5),SUPPR(5),SUPRT(5)
      COMMON /ECON/ BASE(5),CASH(5),PRCOV(144),PRICE(12),PRMN(12),SUPP(5
     1),SUPWT(5),VALUE(5),ZZ(12)
      COMMON /TAP/ MODP,MODT,PFL,PT,T,TP(6),TR
      COMMON /TT/ IDAY,INYR,IYR,JUDAY,K,MON
      COMMON /TX/ DT,TIME,TSTRT,X(99),XNF,X10T0,X11T0
      COMMON /UX/ CH,GN,PH,PR,PRC,PW,RG,SD,TOTAL,V
C
      IF (NCLAS.LE.0) GO TO 101
C
C . . . . . . CHECK FOR EQUALITY OF TIME AND TSTRT.
C
      IF (TIME.LE.TSTRT) GO TO 101
      CALL CYCL2
  101 CALL CYCL1
C
C . . . . . . TRANSFER VALUES OF SYSTEMS VARIABLES TO STATE VARIABLES
C           FOR PLOT AND PRINT OUTPUTS.
C
      X(71)=TR
      X(72)=T
      X(73)=GN
      X(74)=PH
      X(75)=PR
      X(76)=CH
      X(77)=V
      X(78)=PW
      X(79)=FCONS
      X(80)=PFL
      X(87)=PRICE(MON)
      X(88)=TOTAL
      IF (NCLAS.LE.0) RETURN
      DO 102 J=1,NCLAS
         X(J+40)=VALUE(J)
         X(J+45)=BASE(J)
         X(J+50)=SUPP(J)
         X(J+55)=CASH(J)
         X(J+60)=SUPWT(J)
         X(J+65)=LWT(J)
         X(J+80)=ANOS(J)
  102 CONTINUE
      RETURN
C
      END
```

```
      SUBROUTINE CYCLI
C
C . . . . . . THIS SUBROUTINE (CYCLI) IS CALLED BY THE SUBROUTINE
C        (CYCLE) TO DETERMINE WHICH METHOD TO BE USED FOR WEATHER
C        GENERATION AND TO ESTABLISH A LIVESTOCK SCHEDULE.
C
      INTEGER I,IDAY,IN,INFD,INYR,IYR,J,JUDAY,K,M,MKT
      INTEGER MODP,MODT,MON,NCLAS,NOFD,NRA,XNF
      REAL ANOS,ANTOT,AREA,BASE,CASH,DIIT,DT,EVAP,ECAP,ECONS
      REAL GAIN,GNIN,LVWT,LWT,MSZ,OMIN,PFL,PRCOV,PRICE,PRMN
      REAL PT,SOILA,SOILB,STKNO,SUPL,SUPN,SUPP,SUPPR,SUPRT
      REAL SUPWT,T,THOLD,TIME,TOTAL,TP,TR,TSTRT,VALUE,WILT
      REAL X,X10T0,X11T0,ZZ
C
      COMMON /CAT/ ANOS(5),ANTOT,AREA,DIIT,ECONS,GAIN(5),GNIN(5),IN(5),I
     1NFD(5),LVWT(5),LWT(5),MKT(5),MSZ(5),NCLAS,NOFD(5),OMIN(5),STKNO(5)
     2,SUPL(5),SUPN(5),SUPPR(5),SUPRT(5)
      COMMON /ECON/ BASE(5),CASH(5),PRCOV(144),PRICE(12),PRMN(12),SUPP(5
     1),SUPWT(5),VALUE(5),ZZ(12)
      COMMON /PLNT/ EVAP,ECAP,SOILA,SOILB,THOLD,WILT
      COMMON /TAP/ MODP,MODT,PFL,PT,T,TP(6),TR
      COMMON /TT/ IDAY,INYR,IYR,JUDAY,K,MON
      COMMON /TX/ DT,TIME,ISTRT,X(99),XNF,X10T0,X11T0
C
C . . . . . . CALCULATE YEAR AND DAY OF YEAR OF SIMULATION.
C
      TOTAL=0.
      IYR=TIME/365.
      IDAY=TIME-IYR*365.
      IYR=IYR+1
C
C . . . . . . CALCULATE MONTH OF THE YEAR.
C
      MON=IDAY/30.1667+1.0
      IF (MON.GT.12) MON=12
      M=DT+0.5
      IF (IDAY.GT.M) GO TO 101
      PFL=0.
      K=0
  101 IF (MODT-2) 102,103,103
C
C . . . . . . WEATHER DATA READ FROM A TAPE.
C
  102 CALL RECRD (TIME,ISTRT,IDAY,INYR,TP(1),PT)
      GO TO 104
C
C . . . . . . WEATHER DATA CALCULATED BY EITHER A SINE FUNCTION OR
C        GENERATION STOCHASTICALLY IN THE FUNCTIONS (PPT AND TEMP).
C
  103 PT=PPT(IDAY,MODP,TIME,ISTRT,NRA)
      TP(1)=TEMP(IDAY,MODT,NRA,TIME,ISTRT)
  104 TR=TP(1)
C
C . . . . . . COMPUTE 5 DAY TEMPERATURE AVERAGE (T) FROM TEMPERATURE
C        STACK (TP).
C
      T=0.0
      DO 105 I=1,5
         J=7-I
         TP(J)=TP(J-1)
  105 T=T+TP(J)
      IF (IDAY-5) 107,107,106
  106 T=T*.2
C
C . . . . . . ESTABLISH A LIVESTOCK SCHEDULE WITH THE VARIABLES OF
C        TOTAL MASS OF ANIMALS ON PASTURE (X(I)), ACTUAL NUMBER OF
C        ANIMALS ON PASTURE (ANOS(I)), TOTAL ANIMALS PER ACRE (ANTOT),
C        AVERAGE ANIMAL WEIGHT (LVWT(I)), AND ON MARKET DAY, CASH
C        AMOUNT RECEIVED TO THE DATE.
C
  107 ANTOT=0.
      IF (NCLAS.LE.0) GO TO 114
      DO 113 I=1,NCLAS
         ANOS(I)=0.0
         IF (STKNO(I).LE.0.0) GO TO 112
C
C . . . . . . DETERMINE IF ANIMALS OF ANY CLASS ARE ON PASTURE.
C
         IF (IDAY-MKT(I)) 108,111,112
  108    IF (IDAY-IN(I)) 112,109,110
  109    X(I)=LVWT(I)*STKNO(I)/AREA
         GO TO 113
  110    ANOS(I)=STKNO(I)
         ANTOT=ANTOT+ANOS(I)/AREA
         LVWT(I)=X(I)/STKNO(I)*AREA
         GO TO 113
  111    TOTAL=TOTAL+VALUE(I)
C
C . . . . . . TAKE ANIMALS OFF PASTURE.
C
  112    X(I)=0.0
  113 CONTINUE
  114 IF (X(10).LT.0.) X(10)=0.
      IF (X(11).LE.0.0) X(11)=0.0
      IF (X(12).LE.0.0) X(12)=0.0
      IF (X(21).LE.0.0) X(21)=0.0
      RETURN
C
      END
```

```
      SUBROUTINE CYCL2                                                  CYCL2001
C                                                                       CYCL2002
C . . . . . . . THIS SUBROUTINE (CYCL2) IS CALLED BY THE SUBROUTINE     CYCL2003
C        (CYCLE) TO CALCULATE AUXILLARY FINANCIAL VARIABLES SHOWING     CYCL2004
C        STATUS OF THE LIVESTOCK ENTERPRISE BY THE USE OF MONTHLY       CYCL2005
C        PRICE GENERATOR (NRM).                                         CYCL2006
C                                                                       CYCL2007
      INTEGER I,IDAY,IN,INFD,INYR,IYR,JUDAY,K,M,MKT,MON                 CYCL2008
      INTEGER NCLAS,NOFD,XNF                                            CYCL2009
      REAL ANOS,ANTOT,AREA,BASE,CASH,DTIT,DT,FCONS,GAIN,GNIN            CYCL2010
      REAL LVWT,LWT,MSZ,OMIN,PRCOV,PRICE,PRMN,STKNO,SUPL,SUPN           CYCL2011
      REAL SUPP,SUPPR,SUPRT,SUPWT,TIME,ISTRT,VALUE,X,X10TO              CYCL2012
      REAL X11TO,ZZ                                                     CYCL2013
C                                                                       CYCL2014
      COMMON /CAT/ ANOS(5),ANTOT,AREA,DTIT,FCONS,GAIN(5),GNIN(5),IN(5),ICYCL2015
     1NFD(5),LVWT(5),LWT(5),MKT(5),MSZ(5),NCLAS,NOFD(5),OMIN(5),STKNO(5)CYCL2016
     2,SUPL(5),SUPN(5),SUPPR(5),SUPRT(5)                                CYCL2017
      COMMON /ECON/ BASE(5),CASH(5),PRCOV(144),PRICE(12),PRMN(12),SUPP(5CYCL2018
     1),SUPWT(5),VALUE(5),ZZ(12)                                        CYCL2019
      COMMON /TT/ IDAY,INYR,IYR,JUDAY,K,MON                             CYCL2020
      COMMON /TX/ DT,TIME,ISTRT,X(99),XNF,X10TO,X11TO                   CYCL2021
C                                                                       CYCL2022
      M=DT*0.5                                                          CYCL2023
      IF (IDAT.GT.M) GO TO 101                                          CYCL2024
      CALL NRM (PRICE,PRCOV,PRMN,12,ZZ)                                 CYCL2025
  101 IF (PRICE(MON).LE.1.0) PRICE(MON)=1.0                             CYCL2026
      IF (NCLAS.LE.0) RETURN                                            CYCL2027
C                                                                       CYCL2028
C . . . . . . . WHEN LIVESTOCK ARE PUT OUT TO PASTURE (IDAY=IN(I)) THE  CYCL2029
C        PRICE OF CATTLE IS UTILIZED AS A BASE PRICE FOR THE CLASS I.   CYCL2030
C        CALCULATIONS ARE MADE FOR TOTAL AMOUNT PAID FOR                CYCL2031
C        SUPPLEMENTAL FEED (SUPP(I)) BY MULTIPLYING AMOUNT OF FEED      CYCL2032
C        (SUPWT(I)) AND PRICE OF FEED (SUPPR(I)), ALSO CALCULATIONS     CYCL2033
C        ARE MADE FOR AMOUNT INVESTED (CASH(I)), AVERAGE WEIGHT         CYCL2034
C        (LWT(I)) AND POTENTIAL PROFIT, IF ANIMALS SOLD NOW (VALUE(I)). CYCL2035
C                                                                       CYCL2036
      DO 103 I=1,NCLAS                                                  CYCL2037
         IF (IDAY.EQ.IN(I)) BASE(I)=PRICE(MON)*LVWT(I)*STKNO(I)/45400.0 CYCL2038
         SUPP(I)=SUPWT(I)*SUPPR(I)                                      CYCL2039
         CASH(I)=BASE(I)+SUPP(I)                                        CYCL2040
         LWT(I)=0.0                                                     CYCL2041
         IF (INT(STKNO(I)).LE.0) GO TO 102                              CYCL2042
         IF (X(I).LT.0.01) GO TO 102                                    CYCL2043
         LWT(I)=X(I)*AREA/(454.0*STKNO(I))                              CYCL2044
         VALUE(I)=PRICE(MON)*LVWT(I)*STKNO(I)/45400.0-CASH(I)           CYCL2045
         GO TO 103                                                      CYCL2046
  102    VALUE(I)=0.0                                                   CYCL2047
         X(I)=0.0                                                       CYCL2048
  103 CONTINUE                                                          CYCL2049
      RETURN                                                            CYCL2050
C                                                                       CYCL2051
      END                                                               CYCL2052
```

```
      REAL FUNCTION ET(SW,TR,FCAP,THOLD,WILT)
C
C . . . . . . THIS FUNCTION (ET) IS CALLED BY THE SUBROUTINE (XFLWS)
C        TO CALCULATE EVAPOTRANSPIRATION USING MEAN AIR TEMPERATURE
C        (TR) AS THE DETERMINANT.
C
      REAL EO,FCAP,SW,THOLD,TR,WILT
C
      ET=0.0
      IF (TR.LE.0.1) GO TO 102
      EO=.6+.18*TR
      IF (FCAP.LE.0.0) GO TO 101
C
C . . . . . . IF SOIL WATER (SW) IS LESS THAN FIELD CAPACITY (FCAP),
C        ACTUAL EVAPOTRANSPIRATION (ET) IS PROPORTIONAL TO THE RELATIVE
C        WATER CONTENT OF THE SOIL (EO).
C
      ET=EO*SW/FCAP
C
C . . . . . . IF SOIL WATER (SW) IS GREATER THAN OR EQUAL TO FIELD
C        CAPACITY (FCAP), ACTUAL EVAPOTRANSPIRATION (ET) IS TAKEN TO BE
C        THE SAME AS POTENTIAL EVAPOTRANSPIRATION (EO).
C
  101 IF (SW.GE.FCAP) ET=EO
C
C . . . . . . IF SOIL WATER (SW) EXCEEDS TOTAL WATER HOLDING CAPACITY
C        OF SOIL (EO), THE EXCESS IS SET EQUAL TO ET.
C
  102 IF (SW.GT.THOLD) ET=((2.*SW-THOLD)/SW)*ET
      IF (ET.LE.0.0) ET=0.0
C
C . . . . . . IF SOIL WATER (SW) IS LESS THAN OR EQUAL TO THE MINIMUM
C        WATER NEEDS OF THE PLANTS (WILT), ET IS SET EQUAL TO 1/100 OF
C        ET.
C
      IF (SW.LE.WILT) ET=.01*ET
      RETURN
C
      END

      SUBROUTINE GROW (WBM,CBM,SW,TR,SLA,SLB,GW,GC,FCAP,WILT,K)
C
C . . . . . . THIS SUBROUTINE (GROW) IS CALLED BY THE SUBROUTINE
C        (XFLWS) TO CALCULATE COOL AND WARM SEASON PLANT GROWTH.
C
      INTEGER K,KC,KW
      REAL CBM,CMAX,CRMAX,ECR,ESMC,ESMW,ETC,ETW,EWR,FCAP
      REAL GC,GW,QC,SLA,SLB,SW,TR,WBM,WILT,WMAX,WRMAX
C
      COMMON /RESP/ CMAX,CRMAX,WMAX,WRMAX
C
C . . . . . . INITIALIZE GROWTH DAYS COUNTER AND COMPUTE SCALING
C        BASES ON SOIL-WATER PARAMETERS IF FIRST CALL TO THE
C        SUBROUTINE.
C
      IF (K.LE.0) GO TO 101
      GO TO 102
  101 QW=5.0/(FCAP-WILT)
      QC=1./(FCAP-WILT)
      KW=1
      KC=1
C
C . . . . . . CALCULATIONS ARE MADE ON THE EFFECT OF SOIL WATER AND
C        TEMPERATURE ON WARM SEASON GROWTH, AND NET PHOTOSYNTHESIS
C        (GW) BY SUBTRACTING RESPIRATION FROM PHOTOSYNTHESIS.  IF
C        FIVE DAYS GROWTH INITIATION IS PAST, THEN SEVENTY PERCENT
C        OF PHOTOSYNTHATE IS TRANSLOCATED TO THE ROOTS.
C
  102 IF (WBM.LE.0.) GO TO 104
      ESMW=1.-EXP(-QW*(SW-WILT))
      IF (ESMW.LT.0.) ESMW=0.
      ETW=1.-(TR-25.)**2/(15.*15.)
      IF (ETW.LT.0.) ETW=0.
      EWR=.01*EXP(.161*TR)
      GW=ESMW*ETW*WBM*WMAX-EWR*WBM*WRMAX
      IF (GW) 104,104,103
  103 KW=KW+1
      IF (KW.GT.5) GW=GW*.3
C
C . . . . . . CALCULATIONS ARE MADE THE SAME WAY FOR COOL SEASON
C        PLANTS AS FOR WARM SEASON PLANTS ABOVE.
C
  104 IF (CBM.LE.0.) GO TO 106
      ESMC=QC*(SW-WILT)
      IF (ESMC.LT.0.) ESMC=0.
      IF (ESMC.GT.1.) ESMC=1.
      ETC=1.-(TR-18.)**2/(16.*16.)
      IF (ETC.LT.0.) ETC=0.
      ECR=.03*EXP(.126*TR)
      GC=ESMC*ETC*CBM*CMAX-ECR*CBM*CRMAX
      IF (GC) 106,106,105
  105 KC=KC+1
      IF (KC.GT.5) GC=GC*.3
  106 RETURN
C
      END
```

```fortran
      SUBROUTINE NITRO (RGR,NGN,PH,PROT,CH,VITA,PW)
C
C . . . . . . THIS SUBROUTINE (NITRO) IS CALLED BY THE SUBROUTINE
C        (AFLWS) TO COMPUTE PLANT NUTRIENTS IN GREEN FORAGE FROM
C        KNOWN RELATIONS WITH NITROGEN.  NUTRIENT CONTENTS OTHER
C        THAN NITROGEN ARE NITROGEN BASED.  THE NGN/RGR RELATIONSHIP
C        IS FROM URESK 1972.
C
      REAL CH,NGN,PH,PROT,PW,RGR,VITA
C
      NGN=1.0
C
C . . . . . . RELATIONSHIP DEVELOPED BETWEEN RELATIVE GROWTH RATE OF
C        FORAGE (RGR) AND NITROGEN CONTENT OF FORAGE (NGN).
C
      IF (RGR.LT.-0.05) NGN=0.891
      IF (RGR.GT.0.142) NGN=3.0
      IF (RGR.GE.-0.05.AND.RGR.LE.0.142) NGN=10.9*RGR+1.436
      PH=NGN*0.19
      PROT=NGN*6.25
      CH=2.0*NGN-1.0
      VITA=CH*0.5
      PW=30.0*NGN
      RETURN
C
      END
```

```fortran
      SUBROUTINE NRM (PRICE,PRCOV,PRMN,N,ZZ)
C
C . . . . . . THIS SUBROUTINE (NRM) IS CALLED BY THE SUBROUTINE
C        (CYCL2) AS A PRICE GENERATOR CALCULATING A VECTOR OF
C        DEPENDENT NORMAL RANDOM VARIABLES FROM THE SQUARE ROOT
C        MATRIX (PRCOV), WHERE PRCOV * PRCOV = V, IS THE COVARIANCE
C        MATRIX.  THE MATRIX (PRCOV) AND THE ARRAY (PRMN) ARE
C        DISCUSSED IN THE METHODS OF NAYLOR, BALINTFY, BURDICK,
C        AND CHU, 1966.
C
      INTEGER I,J,N
      REAL PRCOV(N,N),PRICE(N),PRMN(N),R1,R2,ZZ(N)
C
      DO 101 I=1,N
C
C . . . . . . GENERATE RANDOM FLUCTUATIONS.
C
         R1=RANF(1.1)
         R2=RANF(1.1)
  101 ZZ(I)=SIN(6.283185*R1)*SQRT(-2.0*ALOG(R2))
      DO 102 I=1,N
         PRICE(I)=0.0
      DO 102 J=1,N
C
C . . . . . . CALCULATE DEVIATIONS USING COVARIANCE MATRIX.
C
  102 PRICE(I)=PRICE(I)+ZZ(J)*PRCOV(I,J)
      DO 103 I=1,N
C
C . . . . . . CALCULATE PRICE FROM DEVIATIONS AND MEAN PRICE.
C
  103 PRICE(I)=PRICE(I)+PRMN(I)
      RETURN
C
      END
```

```
      REAL FUNCTION PPT(IDAY,MODP,TIME,ISTRT,NRA)
C
C . . . . . . THIS FUNCTION (PPT) IS CALLED BY THE SUBROUTINE (CYCLI)
C        TO CALCULATE PRECIPITATION.  THE ALGORITHM REQUIRES A WET DAY
C        PRECEDED BY A DRY DAY AND A WET DAY PRECEDED BY A WET DAY.
C        THUS THE CONVERSION IS MADE BY TAKING THE COMPLEMENT OF THE
C        INPUT PROBABILITIES.
C
      INTEGER I,IDAY,K,LU,MODP,NAME(3),NRA,NT,U1,U2,U3
      REAL A(4),A1,B(4),DDAY,H1,H2,H3,P,Q0(365),Q1(365)
      REAL R,TIME,ISTRT,XLAM(365)
C
      COMMON /XUNT/ LU,U1,U2,U3
C
C . . . . . . CHECK FOR EQUALITY OF TIME AND ISTRT.
C
      IF (TIME.GT.ISTRT) GO TO 104
      IF (MODP.NE.3) GO TO 103
C
C . . . . . . READ FOURIER COEFFICIENTS REPRESENTING PRECIPITATION
C        PARAMETERS IF TIME = START AND STOCHASTIC WEATHER WANTED.
C
      READ (U1,117) (NAME(I),I=1,3)
      WRITE (U2,118) (NAME(I),I=1,3)
C
C . . . . . . READ COEFFICIENT FOR Q0.
C
      READ (U1,119) NT,A1
      READ (U1,120) (A(I),B(I),I=1,NT)
      WRITE (U2,122)
      WRITE (U2,121) NT,A1,(A(I),B(I),I=1,NT)
      CALL SER (NT,A1,A,B,Q0)
C
C . . . . . . READ COEFFICIENT FOR Q1.
C
      READ (U1,119) NT,A1
      READ (U1,120) (A(I),B(I),I=1,NT)
      WRITE (U2,123)
      WRITE (U2,121) NT,A1,(A(I),B(I),I=1,NT)
      CALL SER (NT,A1,A,B,Q1)
C
C . . . . . . CALCULATION OF PROBABILITY (DRY PRECEDED BY WET DAY)
C        FROM PROBABILITY (DRY DAY) AND PROBABILITY (DRY PRECEDED BY
C        DRY) WHICH ARE DERIVED FROM DATA.
C
      Q1(1)=(Q1(1)-Q1(365)*Q0(1))/(1.-Q1(365))
      DO 101 K=2,365
  101 Q1(K)=(Q1(K)-Q1(K-1)*Q0(K))/(1.-Q1(K-1))
      DO 102 K=1,365
C
C . . . . . . DATA DRY/DRY CONVERT TO WET/DRY.
C
         Q0(K)=1.-Q0(K)
C
C . . . . . . DATA DRY/WET CONVERT TO WET/WET.
C
         Q1(K)=1.-Q1(K)
  102 CONTINUE
C
C . . . . . . READ COEFFICIENT FOR RAINFALL DISTRIBUTION.
C
      READ (U1,119) NT,A1
      READ (U1,120) (A(I),B(I),I=1,NT)
      WRITE (U2,124)
      WRITE (U2,121) NT,A1,(A(I),B(I),I=1,NT)
      CALL SER (NT,A1,A,B,XLAM)
      P=Q0(1)
  103 IF (MODP.NE.2) GO TO 104
C
C . . . . . . READ IN PARAMETERS FOR PRECIPITATION CURVE.  THE WAVE-
C        LENGTH IS CONSIDERED TO BE 365 DAYS (.0172 = I*PI/365.)
C
      READ (U1,115) H1,H2,H3
      WRITE (U2,116) H1,H2,H3
  104 IF (IDAY.EQ.0) GO TO 114
      IF (MODP-2) 106,105,107
C
C . . . . . . GENERATE DAILY PRECIPITATION USING A SINE FUNCTION.
C
  105 DDAY=IDAY
      PPT=H1*(SIN((DDAY-H2)*.0172)+1.)+H3
      PPT=PPT/365.
      IF (PPT.LE.0.) PPT=0.
      GO TO 113
  106 RETURN
C
C . . . . . . GENERATE DAILY PRECIPITATION STOCHASTICALLY USING A
C        MARKOV CHAIN FOR DETERMINING WHEN A STORM OCCURS AND AN
C        EXPONENTIALLY DISTRIBUTED EVENT SIZE GIVEN A STORM DID OCCUR.
C
  107 R=RANF(0)
      IF (R-P) 108,108,109
  108 NRA=2
      R=RANF(0)
      PPT=-ALOG(R)*XLAM(IDAY)
      GO TO 110
  109 NRA=1
      PPT=0.
C
```

```
C . . . . . . ESTABLISH NEW CONDITIONAL PROBABILITIES.
C
  110 GO TO (111,112), NRA
  111 P=Q0(IDAY)
      GO TO 113
  112 P=Q1(IDAY)
C
C . . . . . . CONVERT INCHES PRECIPITATION TO MM.
C
  113 PPT=PPT*25.4
      RETURN
  114 PPT=0.
      RETURN
C
C . . . . . . FORMATS USED IN THIS SUBROUTINE.
C
  115 FORMAT (3F10.0)
  116 FORMAT (27H0PRECIP SINE FCN COEF H1 = ,F10.4,5X,5HH2 = ,F10.4,5X,5
     1HH3 = ,F10.4)
  117 FORMAT (2A4,A2)
  118 FORMAT (1H0,2A4,A2)
  119 FORMAT (I3,F10.0)
  120 FORMAT (12F6.4)
  121 FORMAT (1H ,5HNT = ,I2,2X,5HA1 = ,F10.4,2X,18H(A(I),B(I),I=1,NT),1
     10(F6.4,2X))
  122 FORMAT (8H000 COFF)
  123 FORMAT (8H001 COFF)
  124 FORMAT (11H0LAMDA COEF)
C
      END
```

```
      SUBROUTINE RECRD (TIME,TSTRT,IDAY,INYR,TP,P)
C
C . . . . . . THIS SUBROUTINE (RECRD) IS CALLED BY THE SUBROUTINE
C              (CYCL1) TO READ WEATHER DATA RECORDS FROM A TAPE.
C
      INTEGER IDAY,INYR,JUDAY,KYR,LU,MISS,U1,U2,U3
      REAL P,PP,P1,SITE,TIME,TMAX,TMIN,TP,TSTRT,T1,T2
C
      COMMON /XUNT/ LU,U1,U2,U3
C
C . . . . . . CHECK FOR EQUALITY OF TIME AND TSTRT.
C
      IF (TIME.GT.TSTRT) GO TO 101
C
C . . . . . . VALUES P1, T1, T2 ARE INITIALIZED TO FORM A TABLE IN
C              WHICH THE VALUES OF THE PREVIOUS -GOOD DATA- ARE STORED.
C
      P1=0.
      T1=0.
      T2=0.
      MISS=0
  101 IF (MISS.EQ.1) GO TO 103
      READ (U1,108) SITE,KYR,JUDAY,TMAX,TMIN,PP
      IF (KYR) 107,107,102
C
C . . . . . . FIND BEGINNING OF SIMULATION RUN.
C
  102 IF (KYR.LT.INYR) GO TO 101
  103 MISS=0
      IF (IDAY.EQ.0) GO TO 106
      IF (IDAY.EQ.1.AND.JUDAY.EQ.366) GO TO 101
      IF (INT(TIME).EQ.INT(TSTRT+1.)) WRITE (U2,110) KYR,JUDAY,IDAY
C
C . . . . . . DATA RECORD FOUND FOR IDAY BY THREE OPTIONS#
C              JUDAY .LT. IDAY THEN NEW DATA CARD IS READ#
C              JUDAY .GT. IDAY THEN MISSING A RECORD, SET MISS=1, SKIP READ#
C              JUDAY .EQ. IDAY THEN DETERMINE IF ANY MISSING INFORMATION
C              ON THE RECORDS.  T1, T2, AND P1 CONTAIN EITHER CURRENT DAY
C              WEATHER INFORMATION OR PAST GOOD DATA.  IF ACTUAL DATA FOR
C              THE DAY IS MISSING, THE WEATHER DATA FROM THE PREVIOUS
C              -GOOD DATA- DAY IS USED.
C
      IF (JUDAY-IDAY) 101,105,104
  104 MISS=1
      GO TO 106
  105 IF (INT(TMAX).EQ.-0) TMAX=T1
      IF (INT(TMIN).EQ.-0) TMIN=T2
      IF (PP.LT.0.) PP=P1
C
C . . . . . . RESET DATA TABLE.
C
      T1=TMAX
      T2=TMIN
      P1=PP
  106 TP=(T1+T2)/2.
C
C . . . . . . CONVERSION FROM FAHRENHEIT TO CENTIGRADE OF TP AND
C              CONVERSION FROM INCHES TO MM OF P.
C
      TP=(TP-32.)*.55555
      P=P1
      P=P*25.4
      RETURN
  107 WRITE (U2,109) KYR,JUDAY,IDAY
      RETURN
C
C . . . . . . FORMATS USED IN THIS SUBROUTINE.
C
  108 FORMAT (A4,9X,I2,I3,1X,F4.0,1X,F4.0,1X,F5.2)
  109 FORMAT (1X,3I7)
  110 FORMAT (10H    KYR = ,I7,5X,8HJUDAY = ,I7,5X,7HIDAY = ,I7)
C
      END
```

```
      SUBROUTINE RUMEN (LVWT,SUPN,SUPL,OMN,GNN,GAN,MSZ,RDM,RFL,RDMD,MIPR
     1O,VL75)
C
C . . . . . . THIS SUBROUTINE (RUMEN) IS CALLED BY THE SUBROUTINE
C          (AFLWS) TO DETERMINE FORAGE INTAKE AND ANIMAL GAIN AS A
C          FUNCTION OF NITROGEN CONTENT IN THE FORAGE.
C
      REAL AEN,AFAC,ASZ,AVAIL,CAP,D,DG,DIG,DMP,DOM,DRDM
      REAL DS,EFAC,EXR,EYR,FRDR,G,GAN,GFAC,GNN,GPLUSD,IG
      REAL LVWT,MEN,MIPRO,MIPROW,MPAS,MSZ,NDIET,NFOR,NFRDR
      REAL O,OMN,PASFD,PEN,RDM,RDMD,RFL,SG,SUPL,SUPN
      REAL UNDFD,VFA,VL75
C
      COMMON /COW/ AVAIL,EYR,G,GPLUSD,MPAS,NFOR,NFRDR,O
C
C . . . . . . COMPUTATION OF INGESTION BASED ON THE AMOUNT AVAILABLE,
C          AND THE DIGESTION FACTORS.
C
      CAP=.16*VL75
      IG=CAP-RFL
      IF (IG.LT.0.) IG=0.
      IG=AVAIL*IG
      D=D*.45+G*.7
      SG=SUPL*LVWT*.001
      DS=0.
      IF (SG+IG) 102,102,101
  101 DS=SG/(SG+IG)
      D=D*(1.-DS)+.75*DS
  102 NDIET=NFOR*(1.-DS)+SUPN*DS
      DG=D*(IG+SG)
      DOM=DG*4300.
      IF (NDIET.LT.0.) AFAC=0.
      IF (NDIET.LT..6.AND.NDIET.GE.0.) AFAC=.031*NDIET
      IF (NDIET.GE..6.AND.NDIET.LT.1.8) AFAC=.023*(NDIET-.6)+.0186
      IF (NDIET.GE.1.8) AFAC=.0465
      DMP=DOM*AFAC
      MIPRO=MIPRO+DMP-MPAS*MIPRO
      MIPROW=MIPRO/VL75
      FRDR=NFRDR*MIPROW
      EXR=FRDR*EYR
      IF (EXR.GT..95) EXR=.95
      DIG=RDMD+DG
      UNDFD=IG*(1.-D)+RDM-RDMD
      PASFD=(1.0-FRDR)*EYR*DG+EYR*UNDFD
C
C . . . . . . COMPUTATIONS MADE FOR THE RUMEN FACTORS.
C
      DRDM=IG-EXR*DIG-PASFD
      RDM=RDM+DRDM+SUPL*LVWT*.001
      IF (RDM.LT.0.) RDM=0.
      RDMD=FAR*RDM
      IF (RDMD.LT.0.) RDMD=0.
      RFL=RDM+MIPRO*.001
C
C . . . . . . COMPUTATIONS MADE FOR ALL THE ENERGY RELATED VARIABLES.
C
      EFAC=.747-.15*(NDIET-.6)
      IF (NDIET.LT..6) EFAC=.747
      IF (NDIET.GE.1.8) EFAC=.591
      VFA=FRDR*DOM*EFAC
      PEN=.25*PASFD*4300.
      MEN=.65*MIPRO*4.3*MPAS
      AEN=VFA+PEN+MEN-125.*VL75
      ASZ=LVWT/MSZ
      IF (ASZ.LT..05) GFAC=4.
      IF (ASZ.GE..05.AND.ASZ.LT.1.0) GFAC=3.74+5.26*ASZ
      IF (ASZ.GE.1.0) GFAC=9.
      GAN=AEN/GFAC
      OMN=D*IG*1000.
      GNN=G*IG*1000.
      RETURN
C
      END
```

```
      SUBROUTINE SER (NT,A1,A,B,S)
C
C . . . . . . THIS SUBROUTINE (SER) IS CALLED BY THE FUNCTION (PPT)
C          TO OBTAIN DAILY PARAMETER VALUES FOR EXPANSION OF FOURIER
C          SERIES.
C
      INTEGER I,N,NT
      REAL A(10),A1,B(10),S(365),SUM,XI,XN
C
      DO 102 I=1,365
         XI=FLOAT(I)
         SUM=0.
         DO 101 N=1,NT
            XN=FLOAT(N)
  101    SUM=SUM+A(N)*COS(.01725*XI*XN)+B(N)*SIN(.01725*XI*XN)
  102 S(I)=A1+SUM
      RETURN
C
      END
```

```
      SUBROUTINE START                                                  START001
C                                                                       START002
C . . . . . . THIS SUBROUTINE (START) IS CALLED BY THE MAIN PROGRAM    START003
C        (RANGES) TO READ USER DEFINED INITIAL VALUES OR TO             START004
C        INITIALIZE SIMULATION CONTROL VARIABLES.                       START005
C                                                                       START006
      INTEGER I,IDAY,IN,INFD,INYR,IYR,J,JUDAY,K,LU,M,MKT                START007
      INTEGER MODP,MODT,MON,NCLAS,NOFD,U1,U2,U3,XNF                     START008
      REAL ANOS,ANTOT,AREA,BASE,CASH,CH,DIIT,DT,DTFL,DTPL               START009
      REAL DTPR,EVAP,FCAP,FCONS,GAIN,GN,GNIN,LVWT,LWT,MSZ               START010
      REAL OMIN,PFL,PH,PR,PRC,PRCOV,PRICE,PRMN,PT,PW,RG                 START011
      REAL SD,SOILA,SOILB,STKNO,SUPL,SUPN,SUPP,SUPPR,SUPRT              START012
      REAL SUPWT,T,TEND,THOLD,TIME,TINIT,TOTAL,TP,TR,TSTRT              START013
      REAL V,VALUE,WILT,X,X1010,X1110,ZZ                                START014
C                                                                       START015
      COMMON /CAT/ ANOS(5),ANTOT,AREA,DIIT,FCONS,GAIN(5),GNIN(5),IN(5),ISTART016
     1NFD(5),LVWT(5),LWT(5),MKT(5),MSZ(5),NCLAS,NOFD(5),OMIN(5),STKNO(5)START017
     2,SUPL(5),SUPN(5),SUPPR(5),SUPRT(5)                                START018
      COMMON /ECON/ BASE(5),CASH(5),PRCOV(144),PRICE(12),PRMN(12),SUPP(5START019
     1),SUPWT(5),VALUE(5),ZZ(12)                                        START020
      COMMON /IO/ DTFL,DTPL,DTPR,TEND                                   START021
      COMMON /PLNT/ EVAP,FCAP,SOILA,SOILB,THOLD,WILT                    START022
      COMMON /TAP/ MODP,MODT,PFL,PT,T,TP(6),TR                          START023
      COMMON /TT/ IDAY,INYR,IYR,JUDAY,K,MON                             START024
      COMMON /TX/ DT,TIME,TSTRT,X(99),XNF,X1010,X1110                   START025
      COMMON /UX/ CH,GN,PH,PR,PRC,PW,RG,SD,TOTAL,V                      START026
      COMMON /XUNT/ LU,U1,U2,U3                                         START027
C                                                                       START028
C . . . . . . SET ALL NON-READ ARRAYS TO ZERO.                          START029
C                                                                       START030
      DO 102 I=1,12                                                     START031
         PRICE(I)=0.0                                                   START032
         IF (I.GT.6) GO TO 101                                          START033
         TP(I)=0.0                                                      START034
 101     IF (I.GT.5) GO TO 102                                          START035
         ANOS(I)=0.0                                                    START036
         SUPP(I)=0.0                                                    START037
         CASH(I)=0.0                                                    START038
         GAIN(I)=0.0                                                    START039
         BASE(I)=0.0                                                    START040
         GNIN(I)=0.0                                                    START041
         LWT(I)=0.0                                                     START042
         OMIN(I)=0.0                                                    START043
         SUPL(I)=0.0                                                    START044
         VALUE(I)=0.0                                                   START045
         SUPWT(I)=0.0                                                   START046
  102 CONTINUE                                                          START047
C                                                                       START048
C . . . . . . READ DATA FROM DATA SECTION.                              START049
C                                                                       START050
      READ (U1,105) TSTRT,TEND,DT,DTPR,DTFL,DTPL                        START051
      READ (U1,105) X(10),X(11),X(12),X(21)                             START052
      READ (U1,107) NCLAS,MODT,MODP,INYR,AREA,FCAP,THOLD,WILT,SOILA,SOILSTART053
     1B,TINIT                                                           START054
      IF (NCLAS.LE.0) GO TO 103                                         START055
      READ (U1,108) (IN(J),MKT(J),INFD(J),NOFD(J),LVWT(J),SUPRT(J),SUPPRSTART056
     1(J),STKNO(J),SUPN(J),MSZ(J),J=1,NCLAS)                            START057
      READ (U1,106) (PRCOV(I),I=1,144)                                  START058
      READ (U1,106) (PRMN(I),I=1,12)                                    START059
C                                                                       START060
C . . . . . . AREA IS INPUT IN ACRES AND CONVERTED TO SQUARE METERS     START061
C        THEN THE RANDOM NUMBER GENERATOR IS INITIALIZED.               START062
C                                                                       START063
  103 AREA=AREA*4047.                                                   START064
      IF (AREA.LE.0.0) AREA=1.0                                         START065
      CALL RANSET (TINIT)                                               START066
C                                                                       START067
C   . . . . . SET DEFAULT VALUES.                                       START068
C                                                                       START069
      IF (TEND.LE.0.) TEND=365.                                         START070
      IF (DT.LE.0.) DT=1.                                               START071
      IF (NCLAS.GT.5) NCLAS=5                                           START072
      IF (NCLAS.LE.0) RETURN                                            START073
      DO 104 J=1,NCLAS                                                  START074
C                                                                       START075
C . . . . . . ADJUST VALUE OF IN(J) AND MKT(J).                         START076
C                                                                       START077
         M=(IN(J)-TSTRT)/DT+.5                                          START078
         IN(J)=TSTRT+M*DT                                               START079
         M=(MKT(J)-TSTRT)/DT+.5                                         START080
         MKT(J)=TSTRT+M*DT                                              START081
C                                                                       START082
C . . . . . . LVWT IS CONVERTED TO GRAMS AT THIS POINT.                 START083
C                                                                       START084
         MSZ(J)=MSZ(J)*454.                                             START085
         LVWT(J)=LVWT(J)*454.0                                          START086
         IF (STKNO(J).LE.0.0) LVWT(J)=0.0                               START087
  104 CONTINUE                                                          START088
      RETURN                                                            START089
C                                                                       START090
C . . . . . . FORMATS USED IN THIS SUBROUTINE.                          START091
C                                                                       START092
  105 FORMAT (8F10.0)                                                   START093
  106 FORMAT ((7X,6(F10.4,1X)))                                         START094
  107 FORMAT (3I1,I2,5X,7F10.0)                                         START095
  108 FORMAT (4I3,8X,6F10.0)                                            START096
C                                                                       START097
      END                                                               START098
```

```
      REAL FUNCTION TEMP(IDAY,MODT,NRA,TIME,TSTRT)
C
C . . . . . . THIS FUNCTION (TEMP) IS CALLED BY THE SUBROUTINE (CYCL1)
C        TO CALCULATE TEMPERATURE.  A TEMPERATURE SINE FUNCTION CAN
C        BE EASILY DERIVED FOR A LOCATION IF THE AVERAGE YEARLY
C        TEMPERATURE REGIME FOR A NEARBY LOCATION IS KNOWN.  NRA IS
C        A VARIABLE ALLOWING TEMPERATURE TO BE REDUCED WHEN RAIN
C        OCCURS. IF SOME REGIME OTHER THAN SINE IS USED.
C
      INTEGER IDAY,LU,MODT,NRA,U1,U2,U3
      REAL DDAY,H4,H5,H6,TIME,TSTRT
C
      COMMON /XUNT/ LU,U1,U2,U3
C
      IF (MODT-2) 101,102,102
  101 RETURN
C
C . . . . . . CHECK FOR EQUALITY OF TIME AND TSTRT.
C
  102 IF (TIME.GT.TSTRT) GO TO 103
C
C . . . . . . READ IN PARAMETERS FOR TEMPERATURE CURVE.
C        THE WAVELENGTH IS CONSIDERED TO BE 365 DAYS (.0172 =
C        (2*PI/365.).
C
      READ (U1,105) H4,H5,H6
      WRITE (U2,106) H4,H5,H6
  103 IF (IDAY.EQ.0) GO TO 104
      DDAY=IDAY
      TEMP=H4*(SIN((DDAY-H5)*.0172)+1.)+H6
      TEMP=(TEMP-32.)*.55555
      RETURN
  104 TEMP=0.
      RETURN
C
C . . . . . . FORMATS USED IN THIS SUBROUTINE.
C
  105 FORMAT (3F10.0)
  106 FORMAT (25H0TEMP SINE FCN COFF H1 = ,F10.4,5X,5HH5 = ,F10.4,5X,5HH
     16 = ,F10.4)
C
      END
```

```
      SUBROUTINE XFLWS                                                  XFLWS001
C                                                                       XFLWS002
C . . . . . . THIS SUBROUTINE (XFLWS) IS CALLED BY THE MAIN PROGRAM     XFLWS003
C        (RANGES) TO COMPUTE THE VALUES OF THE FLOWS AND STORE THE      XFLWS004
C        VALUES IN THE COMPUTED FLOW STACK.                             XFLWS005
C                                                                       XFLWS006
      INTEGER I,IDAY,IMF,IN,INFD,INYR,IYR,JDDAY,K,MNT                   XFLWS007
      INTEGER MODP,MODT,MON,NCLAS,NOFD,XFR,XN,XNF,XNLOC                 XFLWS008
      INTEGER XNPL,XNPLT,XNPR,XNST,XNVRS,XPL,XPR,XST                    XFLWS009
      REAL ANOS,ANTOT,AREA,AVAIL,B,BASE,CASH,CH,CP,DDAY                 XFLWS010
      REAL DITT,DMX,DFF,DRID,DT,DTFL,DTPL,DTPR,ETOUC,ETODW              XFLWS011
      REAL EVAP,FYR,FCAP,FCONS,FLOW,G,GAIN,GC,GN,GNIN                   XFLWS012
      REAL GPLUSD,GR,GW,LVWT,LWT,MIPRD(5),MPAS,MSZ,NFOR                 XFLWS013
      REAL NFRDR,O,OMIN,PFL,PH,PR,PRC,PRCOV,PRICE,PRMN,PT               XFLWS014
      REAL PW,RDM(5),RDMD(5),REL(5),RG,SD,SOILA,SOILB,STKNO             XFLWS015
      REAL SUPL,SUPN,SUPP,SUPPR,SUPPT,SUPWT,T,TEND,THOLD                XFLWS016
      REAL TIME,TOTAL,TP,TR,TSTRT,V,VALUE,VLZS,WILT,WP,X                XFLWS017
      REAL XF,X10TO,X1011,X11TO,ZZ                                      XFLWS018
C                                                                       XFLWS019
      COMMON /CAT/ ANOS(5),ANTOT,AREA,DITT,FCONS,GAIN(5),GNIN(5),IN(5),IXFLWS020
     1NFD(5),LVWT(5),LWT(5),MKT(5),MSZ(5),NCLAS,NOFD(5),OMIN(5),STKNO(5)XFLWS021
     2,SUPL(5),SUPN(5),SUPPR(5),SUPPT(5)                                XFLWS022
      COMMON /CON/ AVAIL,FYR,G,GPLUSD,MPAS,NFOR,NFKUR,O                 XFLWS023
      COMMON /ECON/ BASE(5),CASH(5),PRCOV(144),PRICE(12),PRMN(12),SUPP(5XFLWS024
     1),SUPWT(5),VALUE(5),ZZ(12)                                        XFLWS025
      COMMON /IO/ DTFL,DTPL,DTPR,TEND                                   XFLWS026
      COMMON /PLNT/ EVAP,FCAP,SOILA,SOILB,THOLD,WILT                    XFLWS027
      COMMON /TAP/ MODP,MODT,PFL,PT,T,TP(6),TR                          XFLWS028
      COMMON /TT/ IDAY,INYR,IYR,JDDAY,K,MON                             XFLWS029
      COMMON /TX/ DT,TIME,TSTRT,X(99),XNF,X10TO,X11TO                   XFLWS030
      COMMON /UX/ CH,GN,PH,PR,PRC,PW,RG,SD,TOTAL,V                      XFLWS031
      COMMON /XSYS/ XF(300),XFR(300),XNLOC(20,5),XNPL,XNPLT,XNPR,XNST,XNXFLWS032
     1VRS(20),XPL(99),XPR(99),XST(99)                                   XFLWS033
C                                                                       XFLWS034
      XN=1                                                              XFLWS035
C                                                                       XFLWS036
C . . . . . . PROCESS = PRECIPITATION.                                  XFLWS037
C                                                                       XFLWS038
      FLOW=PT                                                           XFLWS039
      IF (FLOW.LT.0.) FLOW=0.                                           XFLWS040
      PFL=PFL+FLOW*DT                                                   XFLWS041
      XF(XN)=FLOW                                                       XFLWS042
      XN=XN+1                                                           XFLWS043
C                                                                       XFLWS044
C . . . . . . PROCESS = EVAPOTRANSPIRATION.                             XFLWS045
C                                                                       XFLWS046
      EVAP=ET(X(21),TR,FCAP,THOLD,WILT)                                 XFLWS047
      FLOW=EVAP                                                         XFLWS048
      IF (FLOW.LT.0.) FLOW=0.                                           XFLWS049
      XF(XN)=FLOW                                                       XFLWS050
      XN=XN+1                                                           XFLWS051
C                                                                       XFLWS052
C . . . . . . PROCESS = NET ACCUMULATION OF GREEN FORAGE.               XFLWS053
C        CALCULATION OF PLANT USE OF ATMOSPHERIC CO2 INCLUDING BOTH     XFLWS054
C        GROWTH AND LOSS TERMS WHICH MAYBE LESS THAN ZERO.              XFLWS055
C                                                                       XFLWS056
      GW=0.                                                             XFLWS057
      GC=0.                                                             XFLWS058
      X1011=X(10)+X(11)                                                 XFLWS059
C                                                                       XFLWS060
C . . . . . . WHEN DAILY TEMPERATURES ARE BELOW 10 DEGREES CENTIGRADE,  XFLWS061
C        THERE IS NO GROWTH OF FORAGE.                                  XFLWS062
C                                                                       XFLWS063
      IF (T.LT.5.) GO TO 104                                            XFLWS064
C                                                                       XFLWS065
C . . . . . . ONE GRAM GROWTH INITIATION FOR COOL SEASON IF TEMP .GE.   XFLWS066
C        5, WARM SEASON IF TEMP .GE. 10.                                XFLWS067
C                                                                       XFLWS068
      IF (T.GE.5..AND.X(11).LE..00001) X(11)=1.                         XFLWS069
      IF (T.GE.10..AND.X(10).LE..00001) X(10)=1.                        XFLWS070
      CALL GROW (X(10),X(11),X(21),TR,SOILA,SOILB,GW,GC,FCAP,WILT,K)    XFLWS071
      K=K+1                                                             XFLWS072
C                                                                       XFLWS073
C . . . . . . ONLY CHECK GROWTH IF AT LEAST ONE COMPONENT HAD NET       XFLWS074
C        GROWTH OVER THE LAST DT.                                       XFLWS075
C                                                                       XFLWS076
      IF (GW.LT.0..AND.GC.LT.0.) GO TO 104                              XFLWS077
      EVAP=EVAP*1000.                                                   XFLWS078
      GR=0.                                                             XFLWS079
      DDAY=IDAY+40                                                      XFLWS080
      B=550.+200.*SIN((DDAY)/365.*6.28)                                 XFLWS081
      IF (GW.GT.0..AND.GC.GT.0.) GO TO 103                              XFLWS082
      IF (GW) 101,101,102                                               XFLWS083
C                                                                       XFLWS084
C . . . . . . RECOMPUTE WARM AND COOL SEASON GROWTH ASSUMING GROWTH     XFLWS085
C        WAS REDUCED.                                                   XFLWS086
C                                                                       XFLWS087
  101 IF (GC*B.GT.EVAP) GC=EVAP/B                                       XFLWS088
      GR=GC                                                             XFLWS089
      GO TO 104                                                         XFLWS090
  102 IF (GW*B.GT.EVAP) GW=EVAP/B                                       XFLWS091
      GR=GW                                                             XFLWS092
      GO TO 104                                                         XFLWS093
  103 GR=GW+GC                                                          XFLWS094
```

```
C                                                                       XFLWS095
C . . . . . . . COMPUTE PERCENT OF TOTAL WARM AND COOL SEASON PLANTS    XFLWS096
C           COMPRISE.                                                   XFLWS097
C                                                                       XFLWS098
      WP=GW/GR                                                          XFLWS099
      CP=GC/GR                                                          XFLWS100
C                                                                       XFLWS101
C . . . . . . . IF AMOUNT OF WATER EVAPOTRANSPIRED IS LESS THAN AMOUNT  XFLWS102
C           USED TO PRODUCE FLOW OF BIOMASS, REDUCE FLOW TO MAXIMUM     XFLWS103
C           POSSIBLE.                                                   XFLWS104
C                                                                       XFLWS105
      IF (GR*B.GT.EVAP) GR=EVAP/B                                       XFLWS106
      GW=WP*GR                                                          XFLWS107
      GC=CP*GR                                                          XFLWS108
C                                                                       XFLWS109
C . . . . . . . FLOW CAN BE NEGATIVE IF RESPIRATION .GT. GROWTH.        XFLWS110
C                                                                       XFLWS111
  104 FLOW=GW                                                           XFLWS112
      XF(XN)=FLOW                                                       XFLWS113
      XN=XN+1                                                           XFLWS114
      FLOW=GC                                                           XFLWS115
      XF(XN)=FLOW                                                       XFLWS116
      XN=XN+1                                                           XFLWS117
      RG=0.                                                             XFLWS118
      IF (X1011.LT..00001) GO TO 105                                    XFLWS119
      RG=GR/X1011                                                       XFLWS120
  105 CALL NITRO (RG,GN,PH,PR,CH,V,PW)                                  XFLWS121
C                                                                       XFLWS122
C . . . . . . . PROCESS = PLANT SENESCENCE, DRYING, AND FREEZING.       XFLWS123
C                                                                       XFLWS124
      DMX=.05                                                           XFLWS125
      ETODW=0.                                                          XFLWS126
C                                                                       XFLWS127
C . . . . . . . IF K IS LESS THAN 45 OR THE GROWTH RATE IS LESS THAN AT XFLWS128
C           THE PREVIOUS TIME STEP THEN NO DEATH.                       XFLWS129
C                                                                       XFLWS130
      IF (K.LT.45.OR.X(10).GT.X10T0) GO TO 106                          XFLWS131
C                                                                       XFLWS132
C . . . . . . . WARM SEASON DRYING.                                     XFLWS133
C                                                                       XFLWS134
      ETODW=.1                                                          XFLWS135
      IF (TR.GT.25.) ETODW=.10*(TR-25.)+.10                             XFLWS136
      IF (TR.LT.10.) ETODW=1.1-.1*TR                                    XFLWS137
C                                                                       XFLWS138
C . . . . . . . SECOND TERM REPRESENTS TRAMPLING LOSS.                  XFLWS139
C                                                                       XFLWS140
  106 FLOW=ETODW*X(10)*DMX+.0004*X(10)*ANTOT                            XFLWS141
      IF (FLOW.LT.0.) FLOW=0.                                           XFLWS142
      XF(XN)=FLOW                                                       XFLWS143
      XN=XN+1                                                           XFLWS144
C                                                                       XFLWS145
C . . . . . . . COOL SEASON DRYING.                                     XFLWS146
C                                                                       XFLWS147
      ETODC=0.                                                          XFLWS148
      IF (K.LT.45.OR.X(11).GT.X11T0) GO TO 107                          XFLWS149
      ETODC=.1                                                          XFLWS150
      IF (TR.GT.20.) ETODC=.10*(TR-20.)+.10                             XFLWS151
      IF (TR.LT.5.) ETODC=1.1-.2*TR                                     XFLWS152
  107 FLOW=ETODC*X(11)*DMX+.0004*X(11)*ANTOT                            XFLWS153
      IF (FLOW.LT.0.) FLOW=0.                                           XFLWS154
      XF(XN)=FLOW                                                       XFLWS155
      XN=XN+1                                                           XFLWS156
C                                                                       XFLWS157
C . . . . . . . PROCESS = DISAPPEARANCE OF STANDING DEAD FORAGE.        XFLWS158
C           DISAPPEARANCE OF DEAD FORAGE IS CONTROLLED AS A FUNCTION    XFLWS159
C           OF THE MEAN DAILY TEMPERATURE, SOIL MOISTURE AND TRAMPLING  XFLWS160
C           BY LIVESTOCK.                                               XFLWS161
C                                                                       XFLWS162
      DRTD=.004+.00032*TR                                               XFLWS163
      IF (TR.LT.0.) DRTD=.004                                           XFLWS164
      DPF=1.0+(1./(FCAP-WILT))*(X(21)-WILT)                             XFLWS165
      IF (X(21).GT.FCAP) DPF=2.                                         XFLWS166
      IF (X(21).LT.WILT) DPF=1.                                         XFLWS167
      FLOW=DRTD*DPF*X(12)+.0008*X(12)*ANTOT                             XFLWS168
      IF (FLOW.LT.0.) FLOW=0.                                           XFLWS169
      XF(XN)=FLOW                                                       XFLWS170
C                                                                       XFLWS171
C . . . . . . . LIVESTOCK LOOP.                                         XFLWS172
C                                                                       XFLWS173
      IF (NCLAS.LE.0) RETURN                                            XFLWS174
      IF (X1011.GT..00001) GO TO 108                                    XFLWS175
      WP=0.                                                             XFLWS176
      CP=0.                                                             XFLWS177
      GO TO 109                                                         XFLWS178
  108 WP=X(10)/X1011                                                    XFLWS179
      CP=1.-WP                                                          XFLWS180
C                                                                       XFLWS181
C . . . . . . . COMPUTE TOTAL FORAGE PRESENT.                           XFLWS182
C                                                                       XFLWS183
  109 GPLUSD=X1011+X(12)                                                XFLWS184
      IF (GPLUSD) 110,110,111                                           XFLWS185
  110 NFOR=0.                                                           XFLWS186
      AVAIL=0.                                                          XFLWS187
      GO TO 112                                                         XFLWS188
C                                                                       XFLWS189
C . . . . . . . COMPUTE PERCENTAGE GREEN AND DRY COMPRISE.              XFLWS190
```

```
C
  111 WP=3.*X(10)/(3.*X(10)+X(12))
      CP=1.-WP
      NFOR=WP*GN+CP*SD
C
C . . . . . . . COMPUTE FORAGE AVAILABLE FOR CONSUMPTION.
C
      AVAIL=1.0-EXP(-.167*GPLUSD)
  112 DO 117 I=1,NCLAS
         XN=XN+1
C
C . . . . . . . PROCESS = INTAKE OF GREEN FORAGE.
C
         FLOW=GNIN(I)*WP*STKNO(I)/AREA
         IF (FLOW.LT.0.) FLOW=0.
         FCONS=FCONS+FLOW*DT
         XF(XN)=FLOW
         XN=XN+1
         FLOW=GNIN(I)*CP*STKNO(I)/AREA
         IF (FLOW.LT.0.) FLOW=0.
         FCONS=FCONS+FLOW*DT
         XF(XN)=FLOW
         XN=XN+1
C
C . . . . . . . PROCESS = INTAKE OF STANDING DEAD FORAGE.
C
         FLOW=DMIN(I)*STKNO(I)/AREA
         IF (FLOW.LT.0.) FLOW=0.
         FCONS=FCONS+FLOW*DT
         XF(XN)=FLOW
         XN=XN+1
C
C . . . . . . . PROCESS = INTAKE OF SUPPLEMENTAL FEED.
C
         SUPL(I)=0.0
         FLOW=0.
C
C . . . . . . . DURING SUPPLEMENTAL FEED PERIOD, WEIGHT GAIN DUE TO
C        SUPPLEMENTAL FEEDING IS DETERMINED BY FEED RATE AND BODY MASS.
C
         IF (IDAY.GE.INFD(I).AND.IDAY.LE.NOFD(I)) FLOW=SUPRT(I)*X(I)
C
C . . . . . . . DURING SUPPLEMENTAL FEED PERIOD, CALCULATE PROPORTION OF
C        WEIGHT GAINED FROM SUPPLEMENTAL FEEDING TO DATE.
C
         IF (IDAY.GE.INFD(I).AND.IDAY.LE.NOFD(I)) SUPL(I)=SUPRT(I)
         IF (FLOW.LT.0.) FLOW=0.
         SUPWT(I)=SUPWT(I)+FLOW*DT*AREA/454.0
         XF(XN)=FLOW
         XN=XN+1
C
C . . . . . . . PROCESS = TOTAL OF ANIMAL LOSSES.
C        IF THE DAY IS OUTSIDE THE GRAZING PERIOD, INTAKE OF FORAGE
C        AND WEIGHT GAIN ARE SET EQUAL TO 0.
C
         IF (IDAY.LT.IN(I).OR.IDAY.GT.MKT(I)) GO TO 114
C
C . . . . . . . INITIALIZE SUMMATION VARIABLES.
C
         IME=TIME+.5
         VL75=(.001*LVWT(I))**.75
         IF (IME.NE.IN(I)) GO TO 113
         RDM(I)=.08*VL75
         RFL(I)=RDM(I)
         RDMD(I)=RDM(I)
         MIPRO(I)=200.
C
C . . . . . . . INITIALIZE CONSTANTS.
C
         MPAS=.52
         NFRDR=.04
         EYR=.25
C
C . . . . . . . IF THERE ARE NO ANIMALS, INTAKE OF FORAGE AND WEIGHT
C        GAIN ARE SET TO ZERO.
C
  113    IF (INT(STKNO(I)).GT.0) GO TO 115
  114    OMIN(I)=0.0
         GAIN(I)=0.0
         GNIN(I)=0.0
         GO TO 116
  115    CALL RUMEN (LVWT(I),SUPN(I),SUPL(I),OMIN(I),GNIN(I),GAIN(I),MSZ
     1   (I),RDM(I),RFL(I),RDMD(I),MIPRO(I),VL75)
C
C . . . . . . . ANIMAL LOSSES ARE THE DIFFERENCE BETWEEN TOTAL FOOD
C        INGESTED AND WEIGHT GAINED.
C
  116    FLOW=(GNIN(I)+OMIN(I)-GAIN(I))*STKNO(I)/AREA
         IF (FLOW.LT.0.) FLOW=0.
         XF(XN)=FLOW
  117 CONTINUE
      XNF=XN
      RETURN
C
      END
```

```
      SUBROUTINE XGRAPH (NPL,NVR,NXLOC,XLINE,YLINE,Z,YMIN,YMAX,NN,NXPL)
C
C . . . . . . THIS SUBROUTINE (XGRAPH) IS CALLED BY THE SUBROUTINE
C        (XPLGEN) TO GENERATE AND OUTPUT ONE PRINTER PLOT AT EACH
C        CALL.
C
      INTEGER I,IBLNK,ICHR,IDASH,IEYEE,IP(100),IQUAL,ISTOP,IZ
      INTEGER J,JCHAR,JNT,JP,KEY,KNT,LOC,LU,NN,NPL,NVR
      INTEGER NXLOC(20,5),NXPL(99),U1,U2,U3
      REAL TIME,XLINE(11),YLINE(6),YMAX,YMIN,Z(100),ZI,ZJ,ZZ
C
      COMMON /XFIG/ KEY(5),IBLNK,IDASH,IEYEE,IQUAL,JCHAR(5)
      COMMON /XUNT/ LU,U1,U2,U3
C
      KNT=0
      JNT=0
      DO 101 I=1,NVR
         J=NXLOC(NPL,I)
  101 IP(I)=NXPL(J)
      WRITE (U2,110) NPL,(I,IP(I),I=1,NVR)
      WRITE (U2,111) (YLINE(I),I=1,6)
C
C . . . . . . EACH PASS THROUGH THE FOLLOWING EXPLICIT LOOP (STATEMENT
C        102 TO STATEMENT 108) GENERATES ONE LINE OF THE PRINTED PLOT
C        ON THE OUTPUT.
C        READ IN ONE TIME STEP OF PLOTTING DATA.
C
  102 READ (U3) ISTOP,TIME,(Z(I),I=1,NN)
      IF (ISTOP.GE.1) GO TO 109
C
C . . . . . . INITIALIZE THE OUTPUT CHARACTER STRING (IP) TO CONTAIN
C        BLANKS AND GRAPHICAL REFERENCE LINES.
C
      KNT=KNT+1
      ICHR=IBLNK
      IF (MOD(KNT,10).NE.0) GO TO 103
      IF (KNT.EQ.100) GO TO 103
      ICHR=IDASH
  103 DO 104 I=1,100
         IP(I)=ICHR
         IF (MOD(I,20).NE.0) GO TO 104
         IF (I.EQ.100) GO TO 104
         IP(I)=IEYEE
  104 CONTINUE
C
C . . . . . . INSERT PLOTTING CHARACTERS INTO STRING WHICH REPRESENTS
C        THE PLOTTED VARIABLE.
C
      DO 107 I=1,NVR
         LOC=NXLOC(NPL,I)
         ZZ=Z(LOC)
C
C . . . . . . DETERMINE LOCATION OF PLOTTING CHARACTER IN STRING
C        ACCORDING TO VALUE OF DEPENDENT VARIABLE (ZZ) AND SCALING
C        PARAMETERS.
C
         IZ=100
C
C . . . . . . CHECK FOR THE EQUALITY OF YMAX AND YMIN.
C
         IF (ABS(YMAX-YMIN).LT.1.0E-20) GO TO 105
         ZI=1.+99.*(ZZ-YMIN)/(YMAX-YMIN)
         IZ=ZI
         ZJ=IZ
         IF ((ZI-ZJ).GE.0.5) IZ=IZ+1
         IF (IZ.GT.100) IZ=100
         IF (IZ.LT.1) IZ=1
C
C . . . . . . STORE PLOTTING CHARACTER IN STRING.
C
  105    JP=IP(IZ)
         ICHR=JCHAR(I)
         IF (JP.EQ.IBLNK) GO TO 106
         IF (JP.EQ.IDASH) GO TO 106
         IF (JP.EQ.IEYEE) GO TO 106
         IF (JP.EQ.ICHR) GO TO 107
         ICHR=IQUAL
  106    IP(IZ)=ICHR
  107 CONTINUE
C
C . . . . . . OUTPUT CHARACTER STRING.
C
      IF (KNT.EQ.1.OR.MOD(KNT,10).EQ.0) GO TO 108
      WRITE (U2,112) (IP(I),I=1,100)
      GO TO 102
  108 JNT=JNT+1
      WRITE (U2,113) XLINE(JNT),(IP(I),I=1,100)
      GO TO 102
  109 WRITE (U2,114)
      RETURN
C
C . . . . . . FORMATS USED IN THIS SUBROUTINE.
C
  110 FORMAT (1H1,11H  PLOT NO. ,I2//20X,5(I1,5H = X(,I2,1H),4X))
  111 FORMAT (////13X,F12.1,5(8X,F12.1)/20X,1H+,18X,1H+,4(19X,1H+)/20X,1
     100(1HH))
  112 FORMAT (19X,1HH,100A1,1HH)
  113 FORMAT (5X,F12.1,3H +H,100A1,1HH)
  114 FORMAT (20X,100(1HH))
C
      END
```

```
      SUBROUTINE XOUT
C
C . . . . . . THIS SUBROUTINE (XOUT) IS CALLED BY THE MAIN PROGRAM
C          (RANGES) TO READ THE OUTPUT CONTROL COMMANDS.  EACH CARD
C          INPUT IS SCANNED FOR A COMMAND VERB IN THE FIRST FOUR
C          COLUMNS IN ALPHANUMERIC MODE AND LEFT JUSTIFIED.  UPON
C          RECOGNITION OF A COMMAND VERB THE REMAINING COLUMNS ARE
C          ACCEPTED IN INTEGER MODE, IN FIELDS OF FIVE DIGITS, AND
C          RIGHT JUSTIFIED.  ON THE PLOT CARD THE FIRST NUMERIC FIELD
C          IS TO CONTAIN THE NUMBER OF PLOTS (XNPLT) REQUESTED.
C
      INTEGER I,IBLNK,IDASH,IFYFE,INFO,IP,IQUAL,IVERB,I1,I2
      INTEGER J,JCHAR,K,KEY,L,LU,NUM(14),U1,U2,U3
      INTEGER XFR,XNF,XNLOC,XNPL,XNPLT,XNPR,ANST,XNVRS,XPL
      INTEGER XPR,XST
      REAL DT,TIME,TSTRT,X,XF,X1010,X1110
C
      COMMON /TX/ DT,TIME,TSTRT,X(99),XNF,X1010,X1110
      COMMON /XFTG/ KEY(5),IBLNK,IDASH,IFYFF,IQUAL,JCHAR(5)
      COMMON /XSYS/ XF(300),XFR(300),XNLOC(20,5),XNPL,XNPLT,XNPR,XNST,XN
     1VRS(20),XPL(99),XPR(99),XST(99)
      COMMON /XUNIT/ LU,U1,U2,U3
C
  101 READ (U1,123) IVERB,(NUM(I),I=1,14)
      DO 102 I=1,5
         IF (IVERB.EQ.KEY(I)) GO TO 103
  102 CONTINUE
      GO TO 119
C
C . . . . . . A COMMAND VERB HAS BEEN ENCOUNTERED.
C
  103 GO TO (104,105,109,112,119), I
C
C . . . . . . END.
C
  104 RETURN
C
C . . . . . . PRINT.
C          STORE THE INDICES OF THE STATE VARIABLES REQUESTED FOR
C          PRINTING (NUM(I)) IN THE PRINT STACK (XPR(ANPR)).
C
  105 DO 108 I=1,14
C
C . . . . . . INDICES .LE. 0 ARE ASSUMED A BLANK AND ARE IGNORED.
C          INDICES .GT. 99 PRODUCE A DIAGNOSTIC.  REPETITIOUS REQUESTS
C          ARE IGNORED.
C
         IF (NUM(I).LE.0) GO TO 108
         IF (NUM(I).GT.99) GO TO 120
         IF (ANPR.LE.0) GO TO 107
         DO 106 J=1,XNPR
            IF (NUM(I).EQ.XPR(J)) GO TO 108
  106    CONTINUE
  107    XNPR=XNPR+1
         XPR(ANPR)=NUM(I)
  108 CONTINUE
      GO TO 101
C
C . . . . . . FLOW.
C          SET THE FLOW PRINT FLAG FOR THE K-TH FLOW IN THE FLOW
C          REFERENCE TABLE (XFR(K)) FOR EACH PAIR OF INDICES IN THE
C          RANGE 1 THROUGH 99 (NUM(I) AND NUM (J)) FOR WHICH A
C          CORRESPONDING ENTRY EXISTS IN THE FLOW REFERENCE TABLE.
C
  109 DO 111 I=1,13,2
         J=I+1
         IF (NUM(I).LE.0.AND.NUM(J).LE.0) GO TO 111
         IF (NUM(I).LE.0.OR.NUM(I).GT.99) GO TO 120
         IF (NUM(J).LE.0.OR.NUM(J).GT.99) GO TO 120
         DO 110 K=1,XNF
            INFO=XFR(K)
            CALL XUNPAK (INFO,IP,I1,I2)
            IF (NUM(I).NE.I1.OR.NUM(J).NE.I2) GO TO 110
            IF (IP.EQ.1) GO TO 111
            XFR(K)=XFR(K)+10000
            GO TO 111
  110    CONTINUE
  111 CONTINUE
      GO TO 101
C
C . . . . . . PLOT.
C          THE NEXT (XNPLT) CARDS ARE READ, ONE CARD PER PLOT, WITH THE
C          FIRST FIVE NUMERIC FIELDS INTERPRETED AS THE INDICES OF UP TO
C          FIVE STATE VARIABLES TO APPEAR IN EACH PLOT.
C
  112 XNPLT=NUM(1)
      IF (XNPLT.LE.0.OR.XNPLT.GT.20) GO TO 121
      DO 118 I=1,XNPLT
         READ (U1,123) IVERB,(NUM(J),J=1,14)
         DO 113 J=1,3
            IF (IVERB.EQ.KEY(J)) GO TO 122
  113    CONTINUE
```

```
      XNVRS(I)=0
      DO 117 J=1,5
         IF (NUM(J).LE.0) GO TO 117
         IF (NUM(J).GT.99) GO TO 120
         XNVRS(I)=XNVRS(I)+1
         K=XNVRS(I)
         IF (XNPL.LE.0) GO TO 115
         DO 114 L=1,XNPL
            IF (NUM(J).EQ.XPL(L)) GO TO 116
 114     CONTINUE
 115     XNPL=XNPL+1
         XPL(XNPL)=NUM(J)
 116     XNLOC(I,K)=XNPL
 117  CONTINUE
 118 CONTINUE
     GO TO 101
C
C . . . . . . IF ERRORS OCCURED GENERATE A DIAGNOSTIC.
C
 119 WRITE (U2,124) IVERB,(NUM(I),I=1,14)
     WRITE (U2,125)
     STOP
 120 WRITE (U2,124) IVERB,(NUM(I),I=1,14)
     WRITE (U2,126)
     STOP
 121 WRITE (U2,124) IVERB,(NUM(I),I=1,14)
     WRITE (U2,127)
     STOP
 122 WRITE (U2,124) IVERB,(NUM(I),I=1,14)
     WRITE (U2,128)
     STOP
C
C . . . . . . FORMATS USED IN THIS SUBROUTINE.
C
 123 FORMAT (A4,6X,14I5)
 124 FORMAT (33H0*****ERROR IN DATA SECTION INPUT/10X,A4,6X,14I5)
 125 FORMAT (25H      ILLEGAL COMMAND VERB)
 126 FORMAT (43H      STATE VARIABLE INDEX .LE. 0 OR .GT. 99)
 127 FORMAT (45H      NO. OF PLOTS REQUESTED .LE. 0 OR .GT. 20)
 128 FORMAT (122H      COMMAND VERB ENCOUNTERED WHILE PROCESSING PLOT RE
    1QUEST, CHECK FOR NO. OF PLOTS REQUESTED .NE. NO. OF SUBSEQUENT CAR
    2DS)
C
     END
```

```
      SUBROUTINE XPLGEN
C
C . . . . . . THIS SUBROUTINE (XPLGEN) IS CALLED BY THE MAIN PROGRAM
C          (RANGES) TO GENERATE THE REQUESTED PLOTS UNDER CONTROL OF
C          THE PLOT GENERATION VARIABLES.  THE INFORMATION FOR PLOTTING
C          IS ON THE TEMPORARY MASS STORAGE DEVICE (U3).
C
      INTEGER I,II,ISTOP,LOC,LU,NVARS,U1,U2,U3,XFR,XNF,XNLOC
      INTEGER XNPL,XNPLT,XNPR,XNST,XNVRS,XPL,XPR,XST
      REAL DT,TIME,TSTRT,X,XF,XINC,XLINE(11),XMAX,XMIN,XIST
      REAL XIUTO,XIITO,YINC,YLINE(6),YMAX,YMIN,YIST,Z(100),ZZ
C
      COMMON /TX/ DT,TIME,TSTRT,X(99),XNF,XIUTO,XIITO
      COMMON /XSYS/ XF(300),XFR(300),XNLOC(20,5),XNPL,XNPLT,XNPR,XNST,XN
     1VRS(20),XPL(99),XPR(99),XST(99)
      COMMON /XUNT/ LU,U1,U2,U3
C
      DO 106 II=1,XNPLT
         XMIN=1.E30
         XMAX=-XMIN
         YMIN=1.E30
         YMAX=-YMIN
C
C . . . . . . SEARCH THE PLOT DATA, DETERMINING MAX AND MIN VALUES FOR
C          SCALE.
C
         REWIND U3
  101    READ (U3) ISTOP,TIME,(Z(I),I=1,XNPL)
         IF (ISTOP.GE.1) GO TO 103
         XMIN=AMIN1(XMIN,TIME)
         XMAX=AMAX1(XMAX,TIME)
         NVARS=XNVRS(II)
         DO 102 I=1,NVARS
            LOC=XNLOC(II,I)
            ZZ=Z(LOC)
            YMIN=AMIN1(YMIN,ZZ)
  102    YMAX=AMAX1(YMAX,ZZ)
         GO TO 101
C
C . . . . . . CALCULATE GRAPHICAL SCALING VALUES.
C
  103    CALL XRND (YMIN,YMAX,YMIN,YMAX)
         XLINE(1)=XMIN
         XLINE(11)=XMAX
         XINC=(XMAX-XMIN)/10.
         XIST=XMIN
         DO 104 I=2,10
            XIST=XIST+XINC
  104    XLINE(I)=XIST
         YLINE(1)=YMIN
         YLINE(6)=YMAX
         YINC=(YMAX-YMIN)/5.
         YIST=YMIN
         DO 105 I=2,5
            YIST=YIST+YINC
  105    YLINE(I)=YIST
C
C . . . . . . GENERATE THE PLOT.
C
         REWIND U3
         CALL XGRAPH (II,NVARS,XNLOC,XLINE,YLINE,Z,YMIN,YMAX,XNPL,XPL)
  106 CONTINUE
      RETURN
C
      END
```

```
      SUBROUTINE XPLOT (VAL,ISTOP)
C
C . . . . . . THIS SUBROUTINE (XPLOT) IS CALLED BY THE MAIN PROGRAM
C        (RANGES) TO GENERATE ONE RECORD OF PLOT VARIABLE VALUES ON
C        MASS STORAGE DEVISE (U3) AT EACH CALL.  THE FORMAT OF EACH
C        RECORD IS ISTOP (ISTOP=1 IS FINAL  ISTOP=0 IS OTHERWISE),
C        TIME, AND VAL(I).  THE NUMBER OF RECORDS IS XNPL+2.  STORAGE
C        OF THE STATE VARIABLES TO BE SAVED FOR PLOTTING ARE IN THE
C        ARRAY (VAL).
C
      INTEGER I,ISTOP,J,LU,U1,U2,U3,XFR,XNF,XNLOC,XNPL
      INTEGER XNPLT,XNPR,XNST,XNVRS,XPL,XPR,XST
      REAL DT,TIME,TSTRT,VAL(300),X,XF,X10T0,X11T0
C
      COMMON /TX/ DT,TIME,TSTRT,X(99),XNF,X10T0,X11T0
      COMMON /XSYS/ XF(300),XFR(300),XNLOC(20,5),XNPL,XNPLT,XNPR,XNST,XN
     1VRS(20),XPL(99),XPR(99),XST(99)
      COMMON /XUNT/ LU,U1,U2,U3
C
      DO 101 I=1,XNPL
         J=XPL(I)
  101 VAL(I)=X(J)
      WRITE (U3) ISTOP,TIME,(VAL(I),I=1,XNPL)
      RETURN
C
      END
```

```
      SUBROUTINE XPRINT (VAL,XPR,XNPR)
C
C . . . . . . THIS SUBROUTINE (XPRINT) IS CALLED BY THE MAIN PROGRAM
C        (RANGES) TO PRODUCE PRINTED OUTPUT OF STATE VARIABLE NAMES
C        AND VALUES, FOUR STATE VARIABLES PER LINE.  STORAGE OF THE
C        STATE VARIABLES TO BE PRINTED ARE IN THE ARRAY (VAL).
C
      INTEGER I,J,J1,J2,LU,NKNT,NLINE,U1,U2,U3,XNF,XNPR
      INTEGER XPR(99)
      REAL DT,TIME,TSTRT,VAL(300),X,X10T0,X11T0
C
      COMMON /TX/ DT,TIME,TSTRT,X(99),XNF,X10T0,X11T0
      COMMON /XUNT/ LU,U1,U2,U3
C
      DO 101 I=1,XNPR
         J=XPR(I)
  101 VAL(I)=X(J)
C
C . . . . . . CHECK FOR EQUALITY OF TIME AND TSTRT.
C
      IF (TIME.LE.TSTRT) GO TO 102
      WRITE (U2,110) TIME
  102 NLINE=XNPR/4+1
      NKNT=MOD(XNPR,4)
      IF (NKNT.NE.0) GO TO 103
      NLINE=NLINE-1
      NKNT=4
  103 J1=1
      DO 109 I=1,NLINE
         IF (I.EQ.NLINE) GO TO 104
         J2=J1+3
         WRITE (U2,111) (XPR(J),VAL(J),J=J1,J2)
         IF (LU.GT.0) WRITE (LU,115) (XPR(J),VAL(J),J=J1,J2)
         GO TO 109
  104    J2=J1+NKNT-1
         GO TO (105,106,107,108), NKNT
  105    WRITE (U2,112) (XPR(J),VAL(J),J=J1,J2)
         IF (LU.GT.0) WRITE (LU,115) (XPR(J),VAL(J),J=J1,J2)
         GO TO 109
  106    WRITE (U2,113) (XPR(J),VAL(J),J=J1,J2)
         IF (LU.GT.0) WRITE (LU,115) (XPR(J),VAL(J),J=J1,J2)
         GO TO 109
  107    WRITE (U2,114) (XPR(J),VAL(J),J=J1,J2)
         IF (LU.GT.0) WRITE (LU,115) (XPR(J),VAL(J),J=J1,J2)
         GO TO 109
  108    WRITE (U2,111) (XPR(J),VAL(J),J=J1,J2)
         IF (LU.GT.0) WRITE (LU,115) (XPR(J),VAL(J),J=J1,J2)
  109 J1=J1+4
      RETURN
C
C . . . . . . FORMATS USED IN THIS SUBROUTINE.
C
  110 FORMAT (8H0TIME = ,F12.1)
  111 FORMAT (10X,4(2HX(,I2,4H) = ,F12.2,5X))
  112 FORMAT (10X,2HX(,I2,4H) = ,F12.2,5X)
  113 FORMAT (10X,2(2HX(,I2,4H) = ,F12.2,5X))
  114 FORMAT (10X,3(2HX(,I2,4H) = ,F12.2,5X))
  115 FORMAT (1H0,I2,F12.5)
C
      END
```

```
      SUBROUTINE XRND (ZMIN,ZMAX,RNZMIN,RNZMAX)
C
C . . . . . . THIS SUBROUTINE (XRND) IS CALLED BY THE SUBROUTINE
C        (XPLGEN) TO DETERMINE THE APPROPRIATE SCALING FOR A PLOT
C        OF A FUNCTION WHOSE VALUES RANGE FROM ZMIN TO ZMAX.  RNZMIN
C        AND RNZMAX ARE THE EXTREME VALUES OF THE GRAPH.
C
      INTEGER I
      REAL RNVAR,RNZMAX,RNZMIN,VAR,Z,ZMAX,ZMIN,ZRND,ZZ
C
C . . . . . . THE CASE WHERE ZMIN = ZMAX IS TREATED SEPARATELY.
C
      IF ((ABS(ZMIN-ZMAX)).GT.1.0E-30) GO TO 107
      IF (ABS(ZMAX).GT.1.0) GO TO 101
      RNZMIN=-1.
      RNZMAX=1.
      RETURN
C
C . . . . . . SCALE Z UNTIL THE FIRST SIGNIFICANT DIGIT IS IN THE
C        THOUSANDS PLACE AND ROUND AT THE DECIMAL PLACE.
C
  101 Z=ZMAX
      I=0
  102 IF (Z.GE.1000.) GO TO 103
      Z=Z*10.
      I=I-1
      GO TO 102
  103 IF (Z.LT.10000.) GO TO 104
      Z=Z/10.
      I=I+1
      GO TO 103
  104 Z=INT(Z+.5)
C
C . . . . . . DETERMINE THE NUMBER OF SIGNIFICANT DIGITS IN Z, TRUNCATE
C        THE LAST ONE, AND USE THIS NUMBER AS A BASIS FOR SETTING THE
C        GRAPH VALUES.
C
      Z=Z/10.
      I=I+1
  105 ZRND=INT(Z)
C
C . . . . . . CHECK FOR THE EQUALITY OF ZRND AND Z.
C
      IF ((ABS(ZRND-Z)).GT.1.0E-30) GO TO 106
      Z=Z/10.
      I=I+1
      GO TO 105
  106 IF (Z.GE.0.) RNZMIN=ZRND-1.
      IF (Z.LT.0.) RNZMIN=ZRND-2.
      RNZMAX=RNZMIN+3.
C
C . . . . . . RESTORE THE NUMBERS TO THE ORIGINAL MAGNITUDE.
C
      RNZMIN=RNZMIN*10.**I
      RNZMAX=RNZMAX*10.**I
      RETURN
C
C . . . . . . IN THE GENERAL CASE THE DIFFERENCE, ZMAX-ZMIN, IS
C        TRUNCATED TO THE FIRST SIGNIFICANT DIGIT AND ENLARGED IF
C        NECESSARY TO ENCOMPASS THE ENTIRE RANGE, ZMIN-ZMAX.
C
  107 VAR=ZMAX-ZMIN
      I=0
  108 IF (VAR.GE.1.) GO TO 109
      VAR=VAR*10.
      I=I-1
      GO TO 108
  109 IF (VAR.LT.10.) GO TO 110
      VAR=VAR/10.
      I=I+1
      GO TO 109
  110 RNVAR=INT(VAR)
```

```
C
C . . . . . . CHECK FOR THE EQUALITY OF RNVAR AND VAR.
C
      IF ((ABS(RNVAR-VAR)).LE.1.0E-30) GO TO 111
      IF (VAR.GT.0.) RNVAR=RNVAR+1.
      IF (VAR.LT.0.) RNVAR=RNVAR-1.
C
C . . . . . . TRUNCATE ZMIN AT THE SAME DECIMAL PLACE AS THE
C            DIFFERENCE, ZMAX-ZMIN, WAS TRUNCATED AND LOWER THIS VALUE IF
C            NECESSARY TO INSURE THAT IT IS LESS THAN ZMIN.  THIS VALUE IS
C            USED FOR RNZMIN AND THE TRUNCATED DIFFERENCE, RNVAR, IS ADDED
C            TO OBTAIN RNZMAX (RNVAR IS ENLARGED IF NECESSARY TO INSURE
C            INCLUSION OF THE ENTIRE INTERVAL).
C
  111 Z=ZMIN*10.**(-I)
      ZZ=ZMAX*10.**(-I)
      ZRND=INT(Z)
      IF (VAR.LT.0.) GO TO 112
      IF (Z.GE.0.) RNZMIN=ZRND
      IF (Z.LT.0.) RNZMIN=ZRND-1.
      IF (RNZMIN+RNVAR.LT.ZZ) RNVAR=RNVAR+1.
      GO TO 113
  112 IF (Z.GT.0.) RNZMIN=ZRND+1.
      IF (Z.LE.0.) RNZMIN=ZRND
      IF (RNZMIN+RNVAR.GT.ZZ) RNVAR=RNVAR-1.
  113 RNZMAX=RNZMIN+RNVAR
C
C . . . . . . RESTORE THE NUMBERS TO THE ORIGINAL MAGNITUDE.
C
      RNZMIN=RNZMIN*10.**I
      RNZMAX=RNZMAX*10.**I
      RETURN
C
      END
```

```
      SUBROUTINE XUNPAK (IV,IP,IF,IT)
C
C . . . . . . THIS SUBROUTINE (XUNPAK) IS CALLED BY THE MAIN PROGRAM
C            (RANGES) AND THE SUBROUTINE (XOUT) TO UNPACK INFORMATION
C            STORED IN THE FLOW REFERENCE TABLES.
C                 SOME VARIABLE DEFINITIONS&
C            IV       - INTEGER VARIABLE CONTAINING THE WORD TO BE
C                       UNPACKED.
C            IF       - INTEGER VARIABLE CONTAINING THE INDEX OF SOURCE
C                       COMPARTMENT.
C            IT       - INTEGER VARIABLE CONTAINING THE INDEX OF
C                       DESTINATION COMPARTMENT.
C
      INTEGER I,IF,IP,IT,IV
C
      IP=IV/10000
      I=IV-IP*10000
      IF=I/100
      IT=I-IF*100
      RETURN
C
      END
```

APPENDIX #3

Example Subroutine Substitution

SUBROUTINE RUMEN will be used to explain the procedure for substituting new or revised subroutines for the original. An early version of RUMEN can be substituted into the deck for the current RUMEN routine. Notice that the substituted version of RUMEN has a shorter argument list than the current routine. By shortening the argument list in the calling routine the substitution is correctly made. RUMEN is called near the end of SUBROUTINE XFLWS. The switch is complete by dropping variables TIME, IN, MSZ, RDM, RFL, RDMD, MIPRO and VL75. Conversely, if more or different variables are needed from other routines the argument list of the call and the subroutine can be lengthened. The correct form of the calling statement and the substituted subroutine follow.

In SUBROUTINE XFLWS:

```
CALL RUMEN (GREEN, DRY, NGN, NFOR, LVWT, SUPN, SUPL, OMN, GAN, GNN)
```

The substituted RUMEN routine:

```
      SUBROUTINE RUMEN(GREEN,DRY,NGN,NFOR,LVWT,SUPN,SUPL,OMN,GAN,GNN)
C                                                                       RUMEN002
C     .....CALCULATES INTAKE AND GAIN FOR RUMINANT ANIMALS AS A         RUMEN003
C     FUNCTION OF CONTENT OF FORAGE                                     RUMEN004
C     .....SUBROUTINE SUITABLE FOR NON-REPRODUCTIVE GROWING ANIMALS,    RUMEN005
C     BUT IS NOT DESIGNED FOR MATURE AND PREGNANT ANIMALS.              RUMEN006
C                                                                       RUMEN007
      REAL INTAK,NFOR,NDIET,NGN,NSD,LVWT
C                                                                       RUMEN009
C     INPUTS                                                            RUMEN010
C                                                                       RECRD032
C     GREEN=AMOUNT OF LIVE FORAGE  GRAMS PER METER SQ                   RUMEN011
C     DRY  =AMOUNT OF STANDING DEAD FORAGE  GRAMS PER METER SQ          RUMEN012
C     NGN = NITROGEN CONTENT OF GREEN FORAGE (PERCENT)                  RUMEN013
C     NFOR = AVERAGE NITROGEN CONTENT OF FORAGE SELECTED (PERCENT)      RUMEN014
C          NFOR = DIET VALUES FROM FUNCTION DIET                        RUMEN015
C     LVWT= ANIMAL WEIGHT IN GRAMS
C     SUPL = PROPORTION OF LVWT GIVEN AS SUPPLEMENTAL FEED.
C                         DIMEMSION--  1/TIME                           RUMEN018
C     SUPN =   NITROGEN CONTENT OF SUPPLEMENTAL FEED(PERCENT)           RUMEN019
C
C
C     OUTPUTS                                                           RUMEN020
C
C     GNN =   INTAKE OF LIVE FORAGE(GRAMS)                              RUMEN021
C     OMN =   INTAKE OF STANDING DEAD FORAGE (GRAMS) UNCORRECTED FOR    RUMEN022
C          AMOUNT OF SUPPLEMENTAL FEED SUPPLIED                         RUMEN023
C     GAN  =  ANIMAL GAIN PER DAY(GRAMS)                                RUMEN024
C     NITROGEN CONTENT OF STANDING DEAD HERBAGE ASSUMED TO BE 1.0 PCT   RUMEN025
C     DATA FOR SUBROUTINE FROM RICE 1972                                RUMEN026
C     COMMENTS AND VERIFICATION BY COOK 1972                            RUMEN027
C                                                                       RUMEN028
      IF(LVWT.LE.0.0) GO TO 105
C                                                                       RUMEN030
C     AVAIL TERM ACCORDING TO BEMENT 1969                               RUMEN031
C     CORRECTION OF INTAKE FOR FORAGE AVAILABILITY                      RUMEN032
C                                                                       RUMEN033
      AVAIL=1.0-EXP(-00.167*(GREEN+DRY  ))+SUPL/0.02
      IF (AVAIL.GT.1.0) AVAIL=1.0                                       RUMEN035
      NSD=1.0                                                           RUMEN036
      IF (NFOR.LE.0.4) NFOR=0.4                                         RUMEN037
      NDIET=NFOR                                                        RUMEN038
      FOREF=0.02                                                        RUMEN039
      O=0.0                                                             RUMEN040
C                                                                       RUMEN041
C     .....BW75 IS THE STANDARD BODY WT IN KILOGRAMS TO THE 3/4 POWER   RUMEN042
C                                                                       RUMEN043
      BW75=(0.001*LVWT)**0.75
      K=1                                                               RUMEN045
C                                                                       RUMEN046
C     COMPUTE INTAKE AS A FUNCTION OF NITROGEN CONTENT OF DIET          RUMEN047
C     .....CALCULATE INTAKE PER KILOGRAM OF METABOLIC BODY WEIGHT       RUMEN048
C                                                                       RUMEN049
  101 BWINT=(50.0*NDIET-20.0)*AVAIL                                     RUMEN050
      INTAK=BWINT*BW75                                                  RUMEN051
      IF (SUPL.LE.0.0) GO TO 102                                        RUMEN052
C                                                                       RUMEN053
C     .....IF SUPPLEMENTAL FEED, CALCULATE INTAKE                       RUMEN054
C                                                                       RUMEN055
      FORIT=INTAK/LVWT-SUPL
      IF ((FORIT+SUPL).LE.0.0) GO TO 102                                RUMEN057
C                                                                       RUMEN058
C     ADJUST NITROGEN OF DIET BY SUPPLEMENT                             RUMEN059
C     .....CALCULATE NITROGEN IN DIET                                   RUMEN060
C                                                                       RUMEN061
      NDIET=(NFOR*FORIT+SUPN*SUPL)/(FORIT+SUPL)                         RUMEN062
```

```
C
C     .....COMPARE INTAKE TO REFERENCE
C
      TEST=FOREF/FORIT
      IF (TEST.GT.0.98.AND.TEST.LT.1.02) GO TO 102
      K=K+1
C
C     .....AT MOST 20 ITERATIONS ARE ALLOWED
C
      IF (K.GT.20) GO TO 102
C
C     .....IF ERROR IS GREATER THAN 2 PERCENT, ESTABLISH NEW REFERENCE
C        AND RECOMPUTE INTAKE
C
      FOREF=FORIT
      GO TO 101
  102 INTAK=INTAK-SUPL*LVWT
      IF (NGN.GT.NSD.AND.NGN.GE.NFOR.AND.NFOR.GT.NSD) GO TO 103
      IF(GREEN+ DRY ) 98,98,99
   98 INTAK=0.
      GO TO 104
   99 O=DRY/(GREEN+DRY)
      GO TO 104
  103 CONTINUE
C
C     PROPORTION OF LIVE (G) AND DEAD (O) FORAGE COMPUTED BY EQUATIONS
C          O = 1.0 - G
C           O(NSD) + G(NGN) = NFOR
C     THEREFORE
C
      O=(NGN-NFOR)/(NGN-NSD)
  104 G=1.0-O
C
C     .....CALCULATE DEAD FORAGE INTAKE
C
      OMN=O*INTAK
C
C     .....CALCULATE LIVE FORAGE INTAKE
C
      GNN=G*INTAK
C
C     THE CORRECT FORM OF THE EQUATION FOR GAIN WILL RESEMBLE
C     GAN = (0.35 * BWINT - 14.0) * BW75
C     THE FOLLOWING EQUATION IS A DUMMY PENDING DEBUG OF THE ABOVE
C
      GAN=0.005*LVWT*G + 0.25*SUPL
  105 CONTINUE
      RETURN
C
      END
```

APPENDIX #4

PROGRAM RANGES (INPUT, OUTPUT, PLTSV, TAPE5 = INPUT, TAPE6 = OUTPUT, TAPE3 = PLTSV, TAPE4, TAPE7)

We call this card a "program" card. It specifies the input, output and scratch tape devices. There is a one to one correspondence between the first three parameters and the following three equivalences. The total meaning is that TAPE5 corresponds to the system input device, TAPE6 corresponds to the system output device and TAPE3 is a user declared storage file, in this case the information to be plotted on the line printer is stored on file PLTSV. TAPE4 and TAPE7 are scratch files used for saving any information that may be desired. Specific WRITE statements on these devices are accounted for in SUBROUTINE XPRINT, where LU is defined to be 4 or 7 in the block data subprogram, then the state variables specified in the data section will be written on one of these respective devices. The information can subsequently be used in any manner, i.e. printed, punched, etc.

The meaning of the tape numbers should become clear when considered in conjunction with READ and WRITE statements. For example, READ (U1, 105) means read from TAPE5 which in this program corresponds to the system input device. Similarly, WRITE (U2, 106) will write on the system output device declared TAPE6 on the program card.

Variables U1, U2 and U3 are initialized in a DATA statement at about line 66 of the block data subprogram. Then when the program says READ (U1, 105) a read from TAPE5 = INPUT is implied because U1 was initialized to 5. Also, note these values can be readily changed in the DATA statement to any unit number that is convenient for your installation. For example, if your card reader is designated as unit 1 then just present DATA U1/1/ and all subsequent READS will be from the card reader.

APPENDIX #5

Stochastic Precipitation Generator

A stochastic precipitation algorithm requiring only three parameters for generating a daily precipitation regime has been developed by Todorovic and Woolhiser (1971) and is used in the RANGES model. The three parameters required by the algorithm are (i) probability of a dry day preceded by a dry day, P(D/D), (ii) probability of a dry day, P(D), and (iii) the average storm size, SS. The data values used for these parameters are average weekly values for the year and can be represented succinctly using a Fourier series for each parameter for the year interval. Five Fourier coefficients for each parameter are read by FUNCTION PPT and then expanded by SUBROUTINE SER into daily values for the year. This apparent complexity has been incorporated to allow RANGES to be used at most grassland sites in the western United States. The algorithm used to determine whether a storm occurs on any given day of the year uses the probability of a wet day preceded by a dry day, P(W/D), and the probability of a wet day preceded by a wet day, P(W/W). These can be obtained from the input, P(D/D) and P(D), by the methods of Heermann et al. (1971). A uniformly distributed random variable is used to determine if the conditional probability is high enough to generate a storm, based on whether a storm occurred the previous day. If a storm occurs, another uniform random variable is used in conjunction with the storm size parameter to generate the event size from an exponential distribution.

Algorithm for generating daily precipitation events.

(i) Read in Fourier coefficients for the three precipitation parameters, P(D/D), P(D) and SS.

(ii) Expand into a series representation of the daily values for the year.

$$P(D) = a_0 + a_1 \cos(\pi \cdot \text{day}) + b_1 \sin(\pi \cdot \text{day})$$
$$+ a_2 \cos(2\pi \cdot \text{day}) + b_2 \sin(2\pi \cdot \text{day})$$

where a_0 is the mean response of the parameter over the year and a_1, b_1, a_2, and b_2 are the remaining Fourier regression coefficients.

(iii) Convert P(D/D) and P(D) to P(W/D) and P(W/W). Let t = the current day and t-1 = the preceding day.

$$P(W_t/D_{t-1}) = 1 - P(D_t/D_{t-1})$$

$$P(D_t/W_{t-1}) = P(D_t) - P(D_{t-1}) \cdot P(D_t/D_{t-1})$$

$$P(W_t/W_{t-1}) = 1 - P(D_t/W_{t-1})$$

(iv) Initialize simulation to begin with no storm.

(v) Generate a uniformly distributed random variable.

(vi) Test to see if the random number (r_1) is less than the transition probability (P).

(vii) If $r_1 \leq P$, then generate a storm, otherwise set precipitation for the day to zero and set the transition probability to be tested to P(W/D).

(viii) If a storm occurs, $r_1 > P$, generate another uniform random number and compute the storm size from an exponential distribution.

$$SS = -\log r_2/\lambda$$

where

r_2 = uniform random variable

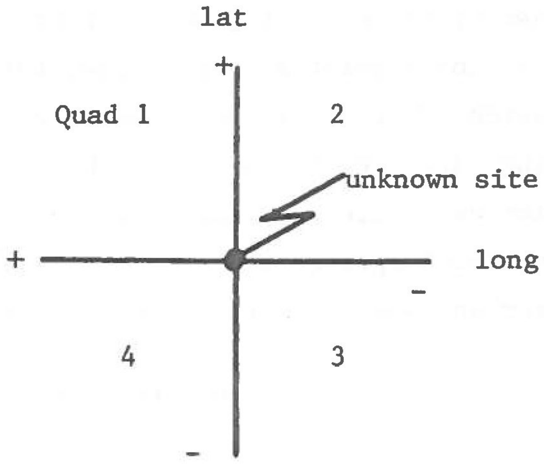
λ = 1/mean weekly storm size.

The Fourier coefficients required by this algorithm can be generated for most grassland sites in the western United States. A computer program, PREGEN, has been written which uses as input latitude, longitude and elevation to generate the necessary Fourier coefficients for input into RANGES. PREGEN checks an unknown site to see if it lies in the area of representativeness of a known weather station as reported by Heermann et al. (1971). If it does, then the Fourier coefficients are generated from the data for the known weather station. Otherwise, a linear interpolation is performed on surrounding known stations to determine a weighted average of their data, from which Fourier coefficients are generated. The average is weighted because elevation is more highly correlated with precipitation than proximity. Therefore, twice as much influence was ascribed to elevation similarity than to horizontal proximity.

Algorithm for interpolating between known weather stations for an unknown site.

(i) Check to see if unknown site lies in area of representativeness of any known stations. If it does use the parameters for the known station unless the user specifies an interpolation is to be performed.

(ii) If a site does not lie in the area of representativeness of any known station then set a conceptual coordinate system on the unknown site with the vertex at the site.



(iii) Now find the minimum distance station within each quadrant within a radius of 120 miles.

(iv) Calculate the weighting factors for, at most, four stations surrounding the unknown station. Note - there may be fewer than four stations.

a) Find the similarity index (SID) for station distances.

$$SID_i = 2w/(a+b_i)$$

where i = quadrants 1, 2, 3, 4;

a = the minimum distance for the four quadrants;

b_i = the distance of the station in quadrant i; and

w = a. In general, w is the minimum of a and b_i.

b) Normalize the similarity indexes (SID).

c) Find similarity index for station elevations (SIE).

d) Normalize SIE.

e) Compute weighting factor

$$WAI = (2 \cdot SIE + SID)/3$$

f) Normalize WAI.

(v) Generate Fourier series for the year for the three parameters P(D/D), P(D) and SS for, at most, four stations.

(vi) Multiply the resulting daily values by the weighting factor.

(vii) Sum the weighted daily values for each parameter to obtain the weighted average for the unknown site.

(viii) Find new Fourier coefficients for the result and print them for use in RANGES.

(ix) If the user specifies rain generation from PREGEN, perform the rain generation algorithm as described for RANGES.

PREGEN users guide. The total PREGEN package consists of a FORTRAN deck and a data deck. The data deck has a card with user specified input parameters and a weather station card file supplied with the program. The user input card allows various PREGEN options to be chosen. All variables on the first data card are integer and should be right justified in the defined field.

Columns	Variable Name	Description
1-5	ILAT	Latitude of site to be interpolated
6-10	ILON	Longitude of site to be interpolated
11-15	ILEV	Elevation of site to be interpolated
16	IS	IS=1, print out name, latitude, longitude and elevation of stations searched
19	IP	IP=1, generate precipitation regime for interpolated site
22-24	IY	IY=n, number of years precipitation to be generated where $n \leq 999$

25	IO	IO=1, daily precipitation amounts printed, otherwise yearly
28	IZ	IZ=1, interpolation is desired even though the unknown may lie in the area of representativeness of a known station.
31	IT	IT=1, ignore interpolation routine, instead read Fourier coefficients from input and generate precipitation amounts with IY and IO used as output criteria. This option implies that the weather station data file should not be read and should be replaced by seven cards with the following format:
1st card		TITLE FORMAT (2A4,A2) TITLE = name of the site for which precipitation is being generated
2nd card		AA(4) FORMAT (3X,F10.0) AA(4) mean response of P(D/D) parameter
3rd card		AA(5),BB(5),AA(6),BB(6) FORMAT (4F10.0) AA(I), BB(I), I=5,6 are Fourier regression coefficients for P(D/D)
4th card		AA(1) FORMAT (3X,F10.0) AA(1) mean response of P(D) parameter
5th card		AA(2),BB(2),AA(3),BB(3) FORMAT (4F10.0) AA(I),BB(I), I=2,3 are Fourier regression coefficients for P(D)
6th card		AA(7) FORMAT(3X,F10.0) AA(7) mean response of SS parameter
7th card		AA(8),BB(8),AA(9),BB(9) FORMAT (4F10.0) AA(I),BB(I), I=8,9 are Fourier regression coefficients for SS

The second part of the data deck consists of 253 weather station records with the following information punched on two cards per station. A blank card will be the terminator for this section.

Columns	Variable Name	Description
1st card		
1-10	NAME	Weather station name
13-18	no variable	Average yearly rainfall
20-22	NN	Representative miles north for station
24-26	NS	Representative miles south for station
28-30	NE	Representative miles east for station
32-34	NW	Representative miles west for station
37-40	LEV	Weather station elevation, feet
43-47	LAT	Weather station latitude, degrees
50-54	LON	Weather station longitude, degrees
2nd card		
1-5	SWDT(1)	Mean response of parameter P(D)
6-10	SWDT(2)	Fourier coefficients for P(D)
11-15	SWDT(3)	Fourier coefficients for P(D)
16-20	SWDT(4)	Fourier coefficients for P(D)
21-25	SWDT(5)	Fourier coefficients for P(D)
26-30	SWDT(6)	Mean response of parameter P(D/D)
•	•	Fourier coefficients
•	•	Fourier coefficients
•	•	Fourier coefficients
46-50	SWDT(10)	Fourier coefficients
51-55	SWDT(11)	Mean response of parameter SS
•	•	Fourier coefficients
•	•	Fourier coefficients
•	•	Fourier coefficients
71-75	SWDT(15)	Fourier coefficients

PREGEN requires the following resources on a CDC 6400 computer; about 35k, 7 seconds compilation time and 7 seconds execution time for an interpolation run with 10 years simulated daily rainfall with yearly summaries only.

```
      PROGRAM PREGEN
     1(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C                                                                              C
C      XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    C
C       XX  XX      XX  XX      XX  XX      XX  XX      XX  XX      XX  XX     C
C        XXXX        XXXX        XXXX        XXXX        XXXX        XXXX      C
C     XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX   C
C     XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX   C
C        XXXX        XXXX        XXXX        XXXX        XXXX        XXXX      C
C       XX  XX      XX  XX      XX  XX      XX  XX      XX  XX      XX  XX     C
C      XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    C
C                                                                              C
C                                                                              C
C     XXXXXXXXX   XXXXXXXXX   XXXXXXXXXX   XXXXXXXX   XXXXXXXXXX  XX      XX   C
C     XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXX     XX   C
C     XX      XX  XX      XX  XX          XX      XX  XX          XXX     XX   C
C     XX      XX  XX      XX  XX          XX          XX          XX X    XX   C
C     XX      XX  XX      XX  XX          XX          XX          XX XX   XX   C
C     XXXXXXXXXX  XXXXXXXXXX  XXXXXX      XX   XXXX   XXXXXX      XX  X   XX   C
C     XXXXXXXXX   XXXXXXXXX   XXXXXX      XX   XXXXX  XXXXXX      XX  X   XX   C
C     XX          XX  XX      XX          XX      XX  XX          XX   X  XX   C
C     XX          XX   XX     XX          XX      XX  XX          XX   XX XX   C
C     XX          XX    XX    XX          XX      XX  XX          XX    XXXX   C
C     XX          XX     XX   XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XX    XXXX   C
C     XX          XX      XX  XXXXXXXXXX   XXXXXXXX   XXXXXXXXXX  XX     XXX   C
C                                                                              C
C                                                                              C
C      XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    C
C       XX  XX      XX  XX      XX  XX      XX  XX      XX  XX      XX  XX     C
C        XXXX        XXXX        XXXX        XXXX        XXXX        XXXX      C
C     XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX   C
C     XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXXXXXX   C
C        XXXX        XXXX        XXXX        XXXX        XXXX        XXXX      C
C       XX  XX      XX  XX      XX  XX      XX  XX      XX  XX      XX  XX     C
C      XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    XX    C
C                                                                              C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C             THIS PROGRAM WAS WRITTEN BY BRAD J. GILBERT FOR USE              C
C        WITH THE COMPUTER PROGRAM (RANGES).                                   C
C                                                                              C
C                   *****************************************                  C
C                                                                              C
C             PURPOSE-   TO DETERMINE THE FOURIER COEFFICIENTS BY              C
C        MERELY INPUTING THE LATITUDE, LONGITUDE, AND ELEVATION OF             C
C        ANY SITE IN THE WESTERN UNITED STATES FOR INPUT INTO THE              C
C        PROGRAM (RANGES).                                                     C
C                                                                              C
C                   *****************************************                  C
C                                                                              C
C             PROGRAM UNITS-    (MAIN PROGRAM) - PREGEN                        C
C        (SUBPROGRAMS) - BLOCK DATA                                            C
C        (SUBROUTINES) - APPLI, BEADAY, FORTI, MINUM, SEARCH, SER              C
C                  WAVG1, WAVG2                                                C
C        (USER DEFINED FUNCTIONS) - ICON                                       C
C        (CONTROL DATA UTILITY FUNCTIONS) - RANF                               C
C        (EXTERNAL ANSI FUNCTIONS) - ALOG, COS, SIN, SQRT                      C
C        (INTRINSIC ANSI FUNCTIONS) - ABS, AMIN1, FLOAT, IABS                  C
C             ABS CALCULATES THE ABSOLUTE VALUE FOR A REAL ARGUMENT.           C
C             ALOG FINDS THE NATURAL LOGARITHM OF A REAL VARIABLE.             C
C             AMIN1 CHOOSES THE SMALLEST VALUE AMONG A LIST OF REAL            C
C        ARGUMENTS.  AS A RESULT THE NUMBER OF ARGUMENTS MUST BE TWO           C
C        OR GREATER.                                                           C
C             COS IS THE TRIGONOMETRIC COSINE WITH THE ARGUMENT IN             C
C        RADIANS AND TYPE REAL.                                                C
C             FLOAT IS A CONVERSION PROCESS GIVEN AN INTEGER ARGUMENT          C
C        TO A REAL VARIABLE.                                                   C
C             IABS FINDS THE ABSOLUTE VALUE FOR AN INTEGER ARGUMENT.           C
C             RANF IS A RANDOM NUMBER GENERATOR WHICH RETURNS VALUES           C
C        UNIFORMLY DISTRIBUTED OVER THE INTERVAL BETWEEN ZERO AND ONE,         C
C        INCLUDING ZERO BUT NOT INCLUDING ONE.  THE ARGUMENT IS JUST           C
C        A DUMMY WHICH ENDS UP BEING IGNORED.                                  C
C             SIN IS THE TRIGONOMETRIC SINE WITH THE ARGUMENT IN               C
C        RADIANS AND TYPE REAL.                                                C
C             SQRT CALCULATES THE SQUARE ROOT OF ITS REAL ARGUMENT.            C
C                                                                              C
C                   *****************************************                  C
C                                                                              C
C             VARIABLE DEFINITIONS-   SOME VARIABLE DEFINITIONS APPEAR         C
C        BELOW WHILE OTHERS ARE DEFINED IN THE USERS GUIDE TO THE              C
C        RANGES GRASSLAND SIMULATION MODEL (APPENDIX 5).  EXCEPT FOR           C
C        SOME SUBSCRIPTS ALL THE VARIABLES ARE DEFINED.                        C
C        A       - REAL ARRAY CONTAINING THE COSINE COEFFICIENTS               C
C                  FOR THE FOURIER SERIES.                                     C
C        A       - REAL VARIABLE CONTAINING A TESTING VALUE ON                 C
C                  ELEVATION SIMILARITY FOR TWO STATIONS EQUIDISTANT           C
C                  IN THE SUBROUTINE (MINUM).                                  C
C        AT      - REAL VARIABLE CONTAINING THE SUMMATION VALUE                C
C                  USED TO NORMALIZE THE ARRAY (AV).                           C
```

```
C         AV        - REAL ARRAY CONTAINING THE AVERAGE WEIGHTING FACTORS.    C
C         AI        - REAL VARIABLE CONTAINING THE WEATHER GENERATION         C
C                     PARAMETER FOR A QUADRANT.                               C
C         B         - REAL ARRAY CONTAINING THE SINE COEFFICIENTS FOR         C
C                     THE FOURIER SERIES.                                     C
C         D         - REAL VARIABLE CONTAINING A TESTING VALUE ON             C
C                     ELEVATION SIMILARITY FOR TWO STATIONS EQUIDISTANT       C
C                     IN THE SUBROUTINE (MINUM).                              C
C         DI        - REAL VARIABLE CONTAINING THE SUMMATION VALUE            C
C                     USED TO NORMALIZE THE ARRAY (QRSQ).                     C
C         ELON      - INTEGER VARIABLE CONTAINING THE EAST LONGITUDE          C
C                     OF AREA OF APPLICABILITY.                               C
C         EOF       - INTEGER VARIABLE CONTAINING BLANK CHARACTERS            C
C                     FOR A END OF FILE CHECK.                                C
C         ET        - REAL VARIABLE CONTAINING THE SUMMATION VALUE            C
C                     USED TO NORMALIZE THE ARRAY (V).                        C
C         IER       - INTEGER VARIABLE CONTAINING AN ERROR CODE FROM          C
C                     THE SUBROUTINE (FORIT).                                 C
C         IT        - INTEGER VARIABLE CONTAINING THE WHOLE NUMBER            C
C                     OF THE VARIABLE (IVAR).                                 C
C         IV        - INTEGER VARIABLE CONTAINING THE FINAL FRACTIONAL        C
C                     PART OF THE VARIABLE (IVAR).                            C
C         IVAR      - INTEGER VARIABLE CONTAINING THE MINUTES OF DEGREES.     C
C         K         - INTEGER VARIABLE CONTAINING THE NUMBER OF STATIONS      C
C                     WITHIN A TWO DEGREE RADIUS IN EACH QUADRANT IN THE      C
C                     SUBROUTINE (MINUM).                                     C
C         KK        - INTEGER ARRAY CONTAINING THE NUMBER OF STATIONS         C
C                     WITHIN A TWO DEGREE RADIUS IN EACH QUADRANT.            C
C         LLAT      - INTEGER VARIABLE CONTAINING THE LATITUDE FACTOR         C
C                     FOR CALCULATING THE VARIABLE (RSQ).                     C
C         LT        - INTEGER ARRAY CONTAINING THE NUMBER OF STORMS           C
C                     IN ONE YEAR STORED BY WEEKS.                            C
C         M         - INTEGER VARIABLE CONTAINING THE MAXIMUM ORDER           C
C                     OF HARMONICS TO BE FITTED.                              C
C         N         - INTEGER VARIABLE CONTAINING THE SIZE OF THE             C
C                     INTERVAL USED TO DETERMINE THE FOURIER                  C
C                     COEFFICIENTS.                                           C
C         NLAT      - INTEGER VARIABLE CONTAINING THE NORTH LATITUDE          C
C                     OF AREA OF APPLICABILITY.                               C
C         NT        - INTEGER VARIABLE CONTAINING A CONSTANT FACTOR           C
C                     IN GENERATING THE FOURIER SERIES OF DAILY VALUES.       C
C         NR        - INTEGER VARIABLE CONTAINING AN INDICATOR OF             C
C                     RAIN ON A CERTAIN DAY OR NO RAIN.                       C
C         P         - REAL VARIABLE CONTAINING A CERTAIN DAYS PROBABILITY     C
C                     FOR DRY DAY PRECEDED BY A DRY DAY.                      C
C         PR        - REAL VARIABLE CONTAINING THE CUMULATIVE YEARLY          C
C                     PRECIPITATION.                                          C
C         QLAT      - REAL ARRAY CONTAINING THE MINIMUM LATITUDES             C
C                     FOR EACH QUADRANT.                                      C
C         QLEV      - REAL ARRAY CONTAINING THE MINIMUM ELEVATIONS            C
C                     FOR EACH QUADRANT.                                      C
C         QLON      - REAL ARRAY CONTAINING THE MINIMUM LONGITUDES            C
C                     FOR EACH QUADRANT.                                      C
C         QMIND     - REAL VARIABLE CONTAINING THE MINIMUM DISTANCE           C
C                     FROM THE UNKNOWN STATION TO A KNOWN STATION IN          C
C                     ANY OF THE QUADRANTS.                                   C
C         QMINL     - REAL VARIABLE CONTAINING THE MINIMUM ELEVATION          C
C                     FROM THE UNKNOWN STATION TO A KNOWN STATION IN          C
C                     ANY OF THE QUADRANTS.                                   C
C         QNAM      - INTEGER MATRIX CONTAINING THE MINIMUM STATION           C
C                     NAMES FOR EACH QUADRANT.                                C
C         QP        - REAL VARIABLE CONTAINING THE YEARLY AVERAGE FOR         C
C                     PRECIPITATION.                                          C
C         QRSQ      - REAL ARRAY CONTAINING THE STATION DISTANCES FOR         C
C                     EACH QUADRANT.                                          C
C         QSWDI     - REAL ARRAY CONTAINING THE WEATHER GENERATION            C
C                     PARAMETERS FOR EACH QUADRANT.                           C
C         QT        - REAL ARRAY CONTAINING THE DAILY PROBABILITIES           C
C                     OF A DRY DAY PRECEDED BY A WET DAY.                     C
C         QUAD      - REAL ARRAY CONTAINING FOURIER PARAMETERS FOR ONE        C
C                     QUADRANT IN THE SUBROUTINE (SFR).                       C
C         Q0        - REAL ARRAY CONTAINING THE DAILY PROBABILITIES           C
C                     OF A DRY DAY PRECEDED BY A DRY DAY.                     C
C         Q1        - REAL ARRAY CONTAINING THE DAILY PROBABILITIES           C
C                     OF A DRY DAY - UNCONDITIONAL.                           C
C         R         - REAL VARIABLE CONTAINING A RANDOM NUMBER FROM           C
C                     THE RANDOM NUMBER GENERATOR USED TO DETERMINE IF A      C
C                     CONDITIONAL PROBABILITY IS HIGH ENOUGH OR USED          C
C                     TO GENERATE A EVENT SIZE.                               C
C         RSQ       - REAL VARIABLE CONTAINING THE STATION DISTANCE           C
C                     FROM A RADIUS OF INTEREST.                              C
C         S         - REAL MATRIX CONTAINING THE FOURIER SERIES EXPANSION     C
C                     FOR PARAMETERS OF FOUR QUADRANTS.                       C
C         SLAT      - INTEGER VARIABLE CONTAINING THE SOUTH LATITUDE          C
C                     OF AREA OF APPLICABILITY.                               C
C         STZ       - REAL ARRAY CONTAINING THE SIZE OF WEEKLY STORMS.        C
C         SUM       - REAL VARIABLE CONTAINING A SUMMATION VALUE USED         C
C                     IN THE EXPANSION OF FOURIER SERIES.                     C
C         T         - INTEGER ARRAY CONTAINING A HEADING FOR THE              C
C                     FOURIER COEFFICIENTS FOR P(D/D), P(D), AND              C
C                     STORM SIZE.                                             C
C         TMP       - REAL VARIABLE CONTAINING THE FRACTIONAL PART OF         C
C                     THE VARIABLE (IVAR).                                    C
C         TS        - REAL ARRAY CONTAINING THE WEIGHTED MEAN FOURIER         C
C                     SERIES EXPANSION.                                       C
C         V         - REAL ARRAY CONTAINING THE STATION ELEVATIONS            C
C                     FOR EACH QUADRANT.                                      C
```

```
C         WLON     - INTEGER VARIABLE CONTAINING THE WEST LONGITUDE             C
C                    OF AREA OF APPLICABILITY.                                  C
C         WM       - REAL ARRAY CONTAINING THE AMOUNT OF PRECIPITATION          C
C                    WEEKLY IN A YEARS TIME.                                    C
C         XI       - REAL VARIABLE CONTAINING THE REAL VALUE OF THE             C
C                    VARIABLE SUBSCRIPT (I) USED IN THE EXPANSION OF            C
C                    THE FOURIER SERIES.                                        C
C         XLAM     - REAL ARRAY CONTAINING THE AVERAGE WEEKLY STORM             C
C                    SIZE REPRESENTED BY THE RECIPROCAL OF THE MEAN             C
C                    DAILY PRECIPITATION ON RAINY DAYS.                         C
C         XLAT     - REAL VARIABLE CONTAINING THE LATITUDE OF THE               C
C                    MINIMUM STATION IN THE SUBROUTINE (MINUM).                 C
C         XLEV     - REAL VARIABLE CONTAINING THE ELEVATION OF THE              C
C                    MINIMUM STATION IN THE SUBROUTINE (MINUM).                 C
C         XLON     - REAL VARIABLE CONTAINING THE LONGITUDE OF THE              C
C                    MINIMUM STATION IN THE SUBROUTINE (MINUM).                 C
C         XN       - REAL VARIABLE CONTAINING THE REAL VALUE OF THE             C
C                    VARIABLE SUBSCRIPT (N) USED IN THE EXPANSION OF            C
C                    THE FOURIER SERIES.                                        C
C         XNAM1    - INTEGER VARIABLE CONTAINING THE FIRST FOUR                 C
C                    CHARACTERS FOR THE NAME OF THE MINIMUM STATION             C
C                    IN THE SUBROUTINE (MINUM).                                 C
C         XNAM2    - INTEGER VARIABLE CONTAINING THE SECOND FOUR                C
C                    CHARACTERS FOR THE NAME OF THE MINIMUM STATION             C
C                    IN THE SUBROUTINE (MINUM).                                 C
C         XNAM3    - INTEGER VARIABLE CONTAINING THE LAST TWO                   C
C                    CHARACTERS FOR THE NAME OF THE MINIMUM STATION             C
C                    IN THE SUBROUTINE (MINUM).                                 C
C         XRSQ     - REAL VARIABLE CONTAINING THE MINIMUM STATION               C
C                    DISTANCE FROM A RADIUS OF INTEREST.                        C
C         ZI       - REAL VARIABLE CONTAINING THE PRECIPITATION FOR             C
C                    A JULIAN DAY.                                              C
C                                                                               C
C                     ****************************************                  C
C                                                                               C
C              PROGRAM FILES-    TAPE5 REFERENCED BY THE VARIABLE (READU)       C
C         IS EQUIVALENCED TO THE INPUT FILE. IN THIS CASE THE CARD              C
C         READER.  TAPE6 REFERENCED BY THE VARIABLE (WRITEU) IS                 C
C         EQUIVALENCED TO THE OUTPUT FILE. IN THIS CASE THE LINE                C
C         PRINTER.  THESE TWO VARIABLES ARE INITIALIZED IN THE BLOCK            C
C         DATA SUBPROGRAM.                                                      C
C                                                                               C
C                     ****************************************                  C
C                                                                               C
C              FORMAT OF DATA ENTRIES ARE ALSO DESCRIBED IN THE USERS           C
C         GUIDE TO THE RANGES GRASSLAND SIMULATION MODEL (APPENDIX 5).          C
C                                                                               C
C                     ****************************************                  C
C                                                                               C
C              REMARKS-   ALL CODING WITHIN THIS PROGRAM HAS BEEN               C
C         CONFORMED TO STANDARD ANSI FORTRAN EXCEPT THE FOLLOWING               C
C         ITEMS, THE PROGRAM CARD WHICH IS A PART OF SCOPE 3.3 OPERATING        C
C         SYSTEM, SOME MIXED MODE OPERATIONS, AND THE CDC RANDOM NUMBER         C
C         UTILITY FUNCTION (RANF) REFERENCED IN THE SUBROUTINE (READAY)         C
C         LINES 97 AND 113.  NOTE THAT USE OF THE FUNCTION (RANF)               C
C         GIVES DIFFERENT OUTPUT FROM THE SUBROUTINE (READAY) FOR               C
C         DIFFERENT SYSTEMS AND DIFFERENT RUNS.                                 C
C              SUBROUTINE (FORIT) WAS EXTRACTED FROM THE IBM SCIENTIFIC         C
C         SUBROUTINE PACKAGE WITH THE ADDITION OF TWO WRITE STATEMENTS.         C
C                                                                               C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      INTEGER I,IJ,ILAT,ILEV,ILON,IO,IP,IS,IT,IY,IZ,I1,I2               PRGFN001
      INTEGER I3,I4,J,K,KK,M,MJ,M60,NF,NN,NS,NW,QNAM                    PRGFN002
      INTEGER READU,WRITEU                                              PRGFN003
      REAL A,AA,AV,B,BB,QLAT,QLEV,QLON,QRSQ,QSWQT,RSQ,S,TS              PRGFN004
C                                                                       PRGFN005
      COMMON /FILES/ READU,WRITEU                                       PRGFN006
      COMMON /VIPS/ A(3),AA(9),AV(5),B(3),BB(9),ILAT,ILEV,ILON,IO,IP,IS,PRGFN007
     1IT,IY,IZ,KK(5),NF,NN,NS,NW,QLAT(5),QLEV(5),QLON(5),QNAM(5,3),QRSQ(PRGFN008
     225),QSWQT(75),RSQ,S(365,4),TS(365)                                PRGFN009
C                                                                       PRGFN010
      READ (READU,111) ILAT,ILON,ILEV,IS,IP,IY,IO,IZ,IT                 PRGFN011
      IF (IT.NE.1) GO TO 101                                            PRGFN012
C                                                                       PRGFN013
C . . . . . . FOURIER COEFFICIENTS READ AS INPUT AND PRECIPITATION      PRGFN014
C        AMOUNTS GENERATED.                                             PRGFN015
C                                                                       PRGFN016
      CALL READAY                                                       PRGFN017
      GO TO 110                                                         PRGFN018
C                                                                       PRGFN019
C . . . . . . INTERPOLATION PROCESS FOR THE FOURIER COEFFICIENTS.       PRGFN020
C                                                                       PRGFN021
  101 DO 102 I=1,5                                                      PRGFN022
          QLAT(I)=0.                                                    PRGFN023
          QLON(I)=0.                                                    PRGFN024
          QLEV(I)=0.                                                    PRGFN025
  102     QRSQ(I)=0.                                                    PRGFN026
          DO 103 J=1,75                                                 PRGFN027
          QSWQT(J)=0.                                                   PRGFN028
  103 CONTINUE                                                          PRGFN029
C                                                                       PRGFN030
C . . . . . . CONVERT MINUTES OF DEGREES TO FRACTIONS OF DEGREES.       PRGFN031
C                                                                       PRGFN032
      ILAT=ICON(ILAT)                                                   PRGFN033
      ILON=ICON(ILON)                                                   PRGFN034
      WRITE (WRITEU,112)                                                PRGFN035
      WRITE (WRITEU,113) ILAT,ILON,ILEV                                 PRGFN036
```

```
C
C . . . . . . SEARCH TAPE AND FIND AT MOST 4 STATIONS IN QUADRANTS
C         SURROUNDING USER LOCATION.
C
      CALL SEARCH
C
C . . . . . . IF UNKNOWN LIES IN AREA OF APPLICABILITY OF A KNOWN
C         STATION WRITE MESSAGE AND CHECK FOR COEFFICIENT USAGE.
C
      IF (KK(5)) 108,108,104
  104 WRITE (WRITEU,114) (QNAM(5,J),J=1,3),QLAT(5),QLON(5),QLEV(5)
      WRITE (WRITEU,122)
      DO 105 I=61,71,5
         I1=I+1
         I2=I+2
         I3=I+3
         I4=I+4
         WRITE (WRITEU,123) QSWDT(I),QSWDT(I1),QSWDT(I3),QSWDT(I2),QSWDT
     I   (I4)
  105 CONTINUE
      IF (IZ.EQ.1) GO TO 108
      K=0
      DO 107 I=1,7,3
         M=I+K
         M60=M+60
         AA(I)=QSWDT(M60)
         BB(I)=0.
         DO 106 J=1,2
            IJ=I+J
            MJ=M+J+60
            AA(IJ)=QSWDT(MJ)
            MJ=MJ+2
            BB(IJ)=QSWDT(MJ)
  106    CONTINUE
         K=K+5
  107 CONTINUE
      CALL READAY
      STOP
  108 WRITE (WRITEU,115)
      WRITE (WRITEU,116)
      DO 109 I=1,4
         WRITE (WRITEU,117) I,(QNAM(I,J),J=1,3),QLAT(I),QLON(I),QLEV(I)
  109 CONTINUE
      WRITE (WRITEU,118)
      WRITE (WRITEU,119) (KK(I),I=1,4)
C
C . . . . . . COMPUTE WEIGHTING FACTORS.
C
      CALL WAVG1
      WRITE (WRITEU,120)
      WRITE (WRITEU,121) (AV(I),I=1,4)
C
C . . . . . . COMPUTE NEW FOURIER COEFFICIENTS TO PASS TO RANGES
C        SIMULATION.
C
      CALL WAVG2
      IF (IP.EQ.1) CALL READAY
  110 STOP
C
C . . . . . . FORMATS USED IN THIS PROGRAM.
C
  111 FORMAT (3I5,6I3)
  112 FORMAT (1H1,20X,39HCOORDINATES OF POINT TO BE INTERPOLATED,52H ALL
     1 LAT AND LON IN DEGREES AND HUNDRETHS OF DEGREES)
  113 FORMAT (1H0,20X,6HLAT = ,I10,5X,6HLON = ,I10,5X,7HELEV = ,I10/)
  114 FORMAT (1H0,49HUNKNOWN LIES IN AREA OF APPLICABILITY OF STATION ,2
     1A4,A2,8H  LAT = ,F5.0,8H  LON = ,F6.0,9H  ELEV = ,F5.0)
  115 FORMAT (1H0,10X,33HMINIMUM STATIONS IN EACH QUADRANT)
  116 FORMAT (1H0,12X,4HQUAD,12X,4HNAME,17X,3HLAT,17X,3HLON,16X,4HELEV)
  117 FORMAT (14X,I1,10X,2A4,A2,3(10X,F10.4))
  118 FORMAT (1H0,10X,76HNUMBER OF STATIONS WITHIN 2 DEGREE RADIUS (APPR
     1OX. 120 MI.) IN EACH QUADRANT)
  119 FORMAT (1H0,2HKK,4(5X,I10))
  120 FORMAT (1H ///18H WEIGHTING FACTORS)
  121 FORMAT (1H0,4(10X,F10.4))
  122 FORMAT (//74H FOURIER COEFFICIENTS FOR CLIMATIC PARAMETERS,P(D/D),
     1 P(D), AND STORM SIZE/17X,2HA0,18X,2HA1,18X,2HB1,18X,2HA2,18X,2HB
     22)
  123 FORMAT (1H0,5(10X,F10.4))
C
      END
```

```
      SUBROUTINE APPLI (LAT,LON,LEV,NAME,SWDT)                          APPLI001
C                                                                       APPLI002
C . . . . . . THIS SUBROUTINE (APPLI) IS CALLED BY THE SUBROUTINE       APPLI003
C       (SEARCH) TO DETERMINE IF AN UNKNOWN STATION LIES WITHIN AN      APPLI004
C       AREA OF APPLICABILITY (ACCORDING TO HEERMAN ET. AL. AND         APPLI005
C       GIFFORD ET AL.).  IF IT LIES IN MORE THAN ONE REGION THE        APPLI006
C       CLOSEST STATION IS CHOSEN.                                      APPLI007
C                                                                       APPLI008
      INTEGER ELON,ILAT,ILEV,ILON,IO,IP,IS,IT,IY,IZ,KK,LAT              APPLI009
      INTEGER LEV,LON,NAME(3),NE,NLAT,NN,NS,NW,QNAM,SLAT                APPLI010
      INTEGER WLON                                                      APPLI011
      REAL A,AA,AV,B,BB,QLAT,QLEV,QLON,QRSQ,QSWDT,RSQ                   APPLI012
      REAL S,SWDT(15),TS                                                APPLI013
C                                                                       APPLI014
      COMMON /VIPS/ A(3),AA(9),AV(5),B(3),BB(9),ILAT,ILEV,ILON,IO,IP,IS,APPLI015
     1IT,IY,IZ,KK(5),NE,NN,NS,NW,QLAT(5),QLEV(5),QLON(5),QNAM(5,3),QRSQ(APPLI016
     225),QSWDT(75),RSQ,S(365,4),TS(365)                                APPLI017
C                                                                       APPLI018
C . . . . . . DETERMINE LATITUDE AND LONGITUDE OF AREA OF APPLICABILITY.APPLI019
C                                                                       APPLI020
      ELON=LON-NE/59.*100                                               APPLI021
      WLON=LON+NW/59.*100                                               APPLI022
      NLAT=LAT+NN/78.*100                                               APPLI023
      SLAT=LAT-NS/78.*100                                               APPLI024
C                                                                       APPLI025
C . . . . . . SEE IF UNKNOWN STATION IN AREA OF APPLICABILITY.          APPLI026
C                                                                       APPLI027
      IF (ILON.LT.ELON.OR.ILON.GT.WLON) GO TO 101                       APPLI028
      IF (ILAT.LT.SLAT.OR.ILAT.GT.NLAT) GO TO 101                       APPLI029
      CALL MINUM (KK(5),LAT,LON,LEV,ILEV,QLAT(5),QLON(5),QLEV(5),QSWDT(6APPLI030
     11),SWDT,RSQ,QRSQ(5),NAME,QNAM(5,1),QNAM(5,2),QNAM(5,3))           APPLI031
  101 RETURN                                                            APPLI032
C                                                                       APPLI033
      END                                                               APPLI034
```

```
      SUBROUTINE READAY
C
C . . . . . . THIS SUBROUTINE (READAY) IS CALLED BY THE MAIN PROGRAM
C        (PREGEN) AND THE SUBROUTINE (SEARCH) TO SIMULATE DAILY
C        PRECIPITATION.
C
      INTEGER I.ILAT.ILEV.ILON.IO.IP.IS.IT.IW.IY.IZ.J.K
      INTEGER KK.L.LT(52).M.NF.NN.NR.NS.NW.READU.T(9)
      INTEGER TITLE(3).WRITEU
      REAL A.AA.AV.B.BB.P.PR.QLAT.QLEV.QLON.QNAM.QP.QRSQ
      REAL QSWQT.QI(365).QO(365).QT(365).P.RSQ.S.SIZ(52).TS
      REAL WM(52).XLAM(365).Z1
C
      COMMON /FILES/ READU.WRITEU
      COMMON /VIPS/ A(3).AA(9).AV(5).B(3).BB(9).ILAT.ILEV.ILON.IO.IP.IS.
     1IT.IY.IZ.KK(5).NF.NN.NS.NW.QLAT(5).QLEV(5).QLON(5).QNAM(5.3).QRSQ(
     225).QSWQT(75).PSQ.S(365.4).TS(365)
C
      DATA T(1)/4HP(D)/.T(2)/4H    /.T(3)/2H  /
      DATA T(4)/4HP(D//.T(5)/4HD)  /.T(6)/2H  /
      DATA T(7)/4HSTOR/.T(8)/4HM SI/.T(9)/2HZE/
C
      K=0
      IF (IT.NE.1) GO TO 102
      READ (READU.115) (TITLE(I).I=1.3)
      WRITE (WRITEU.116) (TITLE(I).I=1.3)
C
C . . . . . . READ FOURIER COEFFICIENTS FOR QO.QI.SS.  PARAMETERS
C        READ IN FOLLOWING ORDER. P(D/D).P(D).SS.
C
      DO 101 L=1.7.3
         K=K+1
         K1=K+1
         K2=K+2
C
C . . . . . . REORDER AA AND BB TO BE COMPATIBLE WITH RANGES AND PREGEN.
C
         I=7
         IF (L.EQ.1) I=4
         IF (L.EQ.4) I=1
         READ (READU.117) AA(I)
         WRITE (WRITEU.118) T(K).T(K1).T(K2).AA(I)
         J=1
         IJ=I+J
         IJ1=IJ+1
         READ (READU.119) AA(IJ).BB(IJ).AA(IJ1).BB(IJ1)
         WRITE (WRITEU.120) AA(IJ).BB(IJ).AA(IJ1).BB(IJ1)
  101 CONTINUE
  102 CALL SER (2.AA(1).AA(2).BB(2).QI)
      CALL SER (2.AA(4).BB(5).BB(5).QO)
      CALL SER (2.AA(7).AA(8).BB(8).XLAM)
      QT(1)=(QI(1)-QI(365)*QO(1))/(1.-QI(365))
C
C . . . . . . CALCULATE P(D/W) FROM P(D) AND P(D/D) WHICH WERE DERIVED
C        FROM DATA.
C
      DO 103 K=2.365
  103 QT(K)=(QI(K)-QI(K-1)*QO(K))/(1.-QI(K-1))
C
C . . . . . . DATA DRY/DRY SO CONVERT TO WET/DRY.
C
      DO 104 K=1.365
         QO(K)=1.-QO(K)
C
C . . . . . . DATA DRY/WET CONVERT TO WET/WET.
C
         QI(K)=1.-QT(K)
  104 CONTINUE
C
C . . . . . . INITIALIZE.
C
      DO 105 M=1.365
C
C . . . . . . XLAM IS IN FACT THE RECIPROCAL OF THE MEAN DAILY
C        PRECIPITATION ON RAINY DAYS.
C
         XLAM(M)=1./XLAM(M)
  105 CONTINUE
      P=QO(1)
      J=1
      DO 106 M=1.52
         LT(M)=0
  106 WM(M)=0.
      QP=0.
C
C . . . . . . YEAR LOOP.
C
      DO 112 J=1.IY
         PR=0.
C
C . . . . . . DAY LOOP.
C
         DO 111 I=1.365
C
C . . . . . . GENERATE UNIFORM RANDOM VARIABLE.
```

```
C                                                                       READY096
            R=RANF(1.0)                                                 READY097
            IF (R.LE.P) GO TO 107                                       READY098
C                                                                       READY099
C . . . . . . NR=1, THEN NO RAIN ON DAY I.                              READY100
C                                                                       READY101
            NR=1                                                        READY102
            ZI=0.                                                       READY103
            GO TO 108                                                   READY104
C                                                                       READY105
C . . . . . . NR=2, THEN RAIN OCCURRED.                                 READY106
C                                                                       READY107
 107        NR=2                                                        READY108
C                                                                       READY109
C . . . . . . TRANSFORM UNIFORM RANDOM VARIABLE (R), TO EXPONENTIAL     READY110
C        DISTRIBUTION WITH PARAMETER LAMBDA.                            READY111
C                                                                       READY112
            R=RANF(1.0)                                                 READY113
            ZI=-ALOG(R)/XLAM(I)                                         READY114
            IW=(I-1)/7+1                                                READY115
            IF (IW.GE.53) GO TO 108                                     READY116
C                                                                       READY117
C . . . . . . COMPUTE FREQUENCY OF STORMS AND AMOUNTS OF                READY118
C        PRECIPITATION.                                                 READY119
C                                                                       READY120
            LT(IW)=LT(IW)+1                                             READY121
            WM(IW)=WM(IW)+ZI                                            READY122
C                                                                       READY123
C . . . . . . CUMULATIVE YEARLY RAINFALL.                               READY124
C                                                                       READY125
 108        IF (IO.EQ.1) WRITE (WRITEU,121) I,ZI                        READY126
            PR=PR+ZI                                                    READY127
            IF (NR-1) 109,109,110                                       READY128
 109        P=Q0(I)                                                     READY129
            GO TO 111                                                   READY130
 110        P=Q1(I)                                                     READY131
 111     CONTINUE                                                       READY132
         WRITE (WRITEU,122) J,PR                                        READY133
         QP=QP+PR                                                       READY134
 112 CONTINUE                                                           READY135
      QP=QP/IY                                                          READY136
      WRITE (WRITEU,123) IY,QP                                          READY137
      WRITE (WRITEU,124)                                                READY138
      DO 114 L=1,52                                                     READY139
         IF (LT(L).NE.0) GO TO 113                                      READY140
         STZ(L)=0.                                                      READY141
         GO TO 114                                                      READY142
 113     STZ(L)=WM(L)/LT(L)                                             READY143
 114 WM(L)=WM(L)/IY                                                     READY144
      WRITE (WRITEU,125) (L,LT(L),WM(L),STZ(L),L=1,52)                  READY145
      RETURN                                                            READY146
C                                                                       READY147
C . . . . . . FORMATS USED IN THIS SUBROUTINE.                          READY148
C                                                                       READY149
 115 FORMAT (2A4,A2)                                                    READY150
 116 FORMAT (1H1,10X,2A4,A2)                                            READY151
 117 FORMAT (3X,F10.0)                                                  READY152
 118 FORMAT (1H0,10X,18HFOURIER COFF FOR  ,2A4,A2,10X,5HA1 = ,F10.4)    READY153
 119 FORMAT (4F10.0)                                                    READY154
 120 FORMAT (12(4X,F6.4))                                               READY155
 121 FORMAT (10X,11HJULIAN DAY ,I3,2X,9HPRECIP = ,F10.4)                READY156
 122 FORMAT (1H0,10X,20HCUM YEARLY PRECIP YR,I3,3H = ,F10.4)            READY157
 123 FORMAT (1H0,10X,19HYEARLY AVERAGE FOR ,I3,1X,8HYEARS = ,F10.4)     READY158
 124 FORMAT (1H1,10X,5HWK NO,10X,9HNO STORMS,10X,16HMEAN WKLY PRECIP,10 READY159
     1X,8HSTORM SZ)                                                     READY160
 125 FORMAT (11X,I5,12X,I5,12X,F16.4,10X,F8.4)                          READY161
C                                                                       READY162
      END                                                               READY163

      BLOCK DATA                                                        BLKDT001
C                                                                       BLKDT002
C . . . . . . THIS SUBPROGRAM (BLOCK DATA) ENTERS DATA VALUES INTO      BLKDT003
C        LABELED COMMON BLOCKS PRIOR TO PROGRAM EXECUTION.              BLKDT004
C                                                                       BLKDT005
      INTEGER ILAT,ILEV,ILON,IO,IP,IS,IT,IY,IZ,KK,NE,NN                 BLKDT006
      INTEGER NS,NW,QNAM,READU,WRITEU                                   BLKDT007
      REAL A,AA,AV,B,BB,QLAT,QLEV,QLON,QRSQ,QSWDT,RSQ,S,TS              BLKDT008
C                                                                       BLKDT009
      COMMON /FILES/ READU,WRITEU                                       BLKDT010
      COMMON /VIPS/ A(3),AA(9),AV(5),B(3),BB(9),ILAT,ILEV,ILON,IO,IP,IS,BLKDT011
     1IT,IY,IZ,KK(5),NE,NN,NS,NW,QLAT(5),QLEV(5),QLON(5),QNAM(5,3),QRSQ(BLKDT012
     25),QSWDT(75),RSQ,S(365,4),TS(365)                                 BLKDT013
C                                                                       BLKDT014
      DATA QNAM(1,1)/4H    /,QNAM(1,2)/4H    /,QNAM(1,3)/2H  /          BLKDT015
      DATA QNAM(2,1)/4H    /,QNAM(2,2)/4H    /,QNAM(2,3)/2H  /          BLKDT016
      DATA QNAM(3,1)/4H    /,QNAM(3,2)/4H    /,QNAM(3,3)/2H  /          BLKDT017
      DATA QNAM(4,1)/4H    /,QNAM(4,2)/4H    /,QNAM(4,3)/2H  /          BLKDT018
      DATA QNAM(5,1)/4H    /,QNAM(5,2)/4H    /,QNAM(5,3)/2H  /          BLKDT019
      DATA READU/5/,WRITEU/6/                                           BLKDT020
C                                                                       BLKDT021
      END                                                               BLKDT022
```

```
      SUBROUTINE FORIT (FNT,N,M,A,B,IER)
C
C . . . . . . THIS SUBROUTINE (FORIT) IS CALLED BY THE SUBROUTINE
C        (WAVE2).  IT WAS TAKEN DIRECTLY FROM THE IBM SCIENTIFIC
C        SUBROUTINE PACKAGE.
C
C     SUBROUTINE FORIT
C
C     PURPOSE
C        FOURIER ANALYSIS OF A PERIODICALLY TABULATED FUNCTION.
C        COMPUTES THE COEFFICIENTS OF THE DESIRED NUMBER OF TERMS
C        IN THE FOURIER SERIES F(X)=A(0)+SUM(A(K)COS KX+B(K)SIN KX)
C        WHERE K=1,2,...,M TO APPROXIMATE A GIVEN SET OF
C        PERIODICALLY TABULATED VALUES OF A FUNCTION.
C
C     USAGE
C     CALL FORIT(FNT,N,M,A,B,IER)
C
C     DESCRIPTION OF PARAMETERS
C        FNT-VECTOR OF TABULATED FUNCTION VALUES OF LENGTH 2N+1
C        N  -DEFINES THE INTERVAL SUCH THAT 2N+1 POINTS ARE TAKEN
C            OVER THE INTERVAL (0,2PI). THE SPACING IS THUS 2PI/2N+1
C        M  -MAXIMUM ORDER OF HARMONICS TO BE FITTED
C        A  -RESULTANT VECTOR OF FOURIER COSINE COEFFICIENTS OF
C            LENGTH M+1
C            A SUB 0, A SUB 1,..., A SUB M
C        B  -RESULTANT VECTOR OF FOURIER SINE COEFFICIENTS OF
C            LENGTH M+1
C            B SUB 0, B SUB 1,..., B SUB M
C        IER-RESULTANT ERROR CODE WHERE
C           IER=0 NO ERROR
C           IER=1 N NOT GREATER OR EQUAL TO M
C           IER=2 M LESS THAN 0
C
C     REMARKS
C        M MUST BE GREATER THAN OR EQUAL TO ZERO
C        N MUST BE GREATER THAN OR EQUAL TO M
C        THE FIRST ELEMENT OF VECTOR B IS ZERO IN ALL CASES
C
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C     NONE
C
C     METHOD
C        USES RECURSIVE TECHNIQUE DESCRIBED IN A. RALSTON, H. WILF,
C        'MATHEMATICAL METHODS FOR DIGITAL COMPUTERS', JOHN WILEY
C        AND SONS, NEW YORK, 1960, CHAPTER 24. THE METHOD OF INDEXING
C        THROUGH THE PROCEDURE HAS BEEN MODIFIED TO SIMPLIFY THE
C        COMPUTATION.
C
C
      INTEGER READU,WRITEU
      DIMENSION A(3), B(3), FNT(365)
      COMMON /FILES/ READU,WRITEU
```

```
C                                                                       FORIT054
C . . . . . . CHECK FOR ERROR PARAMETER ERRORS.                         FORIT055
C                                                                       FORIT056
      IER=0                                                             FORIT057
      IF (M) 101,102,102                                                FORIT058
  101 IER=2                                                             FORIT059
      WRITE (WRITEU,110)                                                FORIT060
      RETURN                                                            FORIT061
  102 IF (M-N) 104,104,103                                              FORIT062
  103 IER=1                                                             FORIT063
      WRITE (WRITEU,111)                                                FORIT064
      RETURN                                                            FORIT065
C                                                                       FORIT066
C . . . . . . COMPUTE AND PRESET CONSTANTS.                             FORIT067
C                                                                       FORIT068
  104 AN=N                                                              FORIT069
      COEF=2.0/(2.0*AN+1.0)                                             FORIT070
      CONST=3.141593*COEF                                               FORIT071
      S1=SIN(CONST)                                                     FORIT072
      C1=COS(CONST)                                                     FORIT073
      C=1.0                                                             FORIT074
      S=0.0                                                             FORIT075
      J=1                                                               FORIT076
      FNTZ=FNT(1)                                                       FORIT077
  105 U2=0.0                                                            FORIT078
      U1=0.0                                                            FORIT079
      I=2*N+1                                                           FORIT080
C                                                                       FORIT081
C . . . . . . FORM FOURIER COEFFICIENTS RECURSIVELY.                    FORIT082
C                                                                       FORIT083
  106 U0=FNT(I)+2.0*C*U1-U2                                             FORIT084
      U2=U1                                                             FORIT085
      U1=U0                                                             FORIT086
      I=I-1                                                             FORIT087
      IF (I-1) 107,107,106                                              FORIT088
  107 A(J)=COEF*(FNTZ+C*U1-U2)                                          FORIT089
      B(J)=COEF*S*U1                                                    FORIT090
      IF (J-(M+1)) 108,109,109                                          FORIT091
  108 U=C1*C-S1*S                                                       FORIT092
      S=C1*S+S1*C                                                       FORIT093
      C=U                                                               FORIT094
      J=J+1                                                             FORIT095
      GO TO 105                                                         FORIT096
  109 A(1)=A(1)*0.5                                                     FORIT097
      RETURN                                                            FORIT098
C                                                                       FORIT099
C . . . . . . FORMATS USED IN THIS SUBROUTINE.                          FORIT100
C                                                                       FORIT101
  110 FORMAT (90H0IN FORIT - M MUST BE GREATER THAN OR EQUAL TO ZERO. PLFORIT102
     1EASE CHECK SITUATION NOT SATISFIED.)                              FORIT103
  111 FORMAT (87H0IN FORIT - N MUST BE GREATER THAN OR EQUAL TO M. PLEASFORIT104
     1E CHECK SITUATION NOT SATISFIED.)                                 FORIT105
C                                                                       FORIT106
      END                                                               FORIT107
```

```
      FUNCTION ICON(IVAR)                                               ICON 001
C                                                                       ICON 002
C . . . . . . . THIS FUNCTION (ICON) IS CALLED BY THE MAIN PROGRAM      ICON 003
C        (PREGEN) AND THE SUBROUTINE (SEARCH) TO CONVERT MINUTES OF     ICON 004
C        DEGREES TO FRACTIONS OF DEGREES.                               ICON 005
C                                                                       ICON 006
      INTEGER IT,IV,IVAR                                                ICON 007
      REAL TMP                                                          ICON 008
C                                                                       ICON 009
      IT=IVAR/100                                                       ICON 010
      IT=IT*100                                                         ICON 011
      TMP=IVAR-IT                                                       ICON 012
      IV=(TMP/60.)*100.                                                 ICON 013
      ICON=IT+IV                                                        ICON 014
      RETURN                                                            ICON 015
C                                                                       ICON 016
      END                                                               ICON 017
```

```
      SUBROUTINE MINUM (K,LAT,LON,LEV,ILEV,XLAT,XLON,XLEV,QSWDT,SWDT,RSQ
     1,XRSQ,NAME,XNAM1,XNAM2,XNAM3)
C
C . . . . . . THIS SUBROUTINE (MINUM) IS CALLED BY THE SUBROUTINES
C          (SEARCH AND APPL1) TO FIND THE CLOSEST STATION TO THE POINT
C          OF INTEREST.  IT IS CALLED FOR EACH QUADRANT.
C
      INTEGER ILEV,K,LAT,LEV,LON,M,M1,M2,M3,M4,NAME(3)
      INTEGER XNAM1,XNAM2,XNAM3
      REAL A,B,QSWDT(15),RSQ,SWDT(15),XLAT,XLEV,XLON,XRSQ
C
      IF (K.EQ.0) GO TO 102
      IF (RSQ-XRSQ) 102,101,104
C
C . . . . . . IF TWO STATIONS ARE EQUIDISTANT THEN THE CLOSEST DEPENDS
C          ON ELEVATION SIMILARITY.
C
  101 A=ABS(FLOAT(ILEV)-XLEV)
      B=IABS(ILEV-LEV)
      IF (A-B) 104,104,102
C
C . . . . . . SAVE THE LATITUDE, LONGITUDE, ELEVATION AND FOURIER
C          PARAMETERS OF THE MINIMUM STATION.
C
  102 XLAT=LAT
      XLON=LON
      XLEV=LEV
      XRSQ=RSQ
      XNAM1=NAME(1)
      XNAM2=NAME(2)
      XNAM3=NAME(3)
      K=K+1
C
C . . . . . . REORDER THE ARRAY (QSWDT) TO SATISFY THE REQUIREMENTS
C          FOR THE SUBROUTINE (SER).
C
      DO 103 M=1,11,5
         M1=M+1
         M2=M+2
         M3=M+3
         M4=M+4
         QSWDT(M)=SWDT(M)*.001
         QSWDT(M1)=SWDT(M1)*.001
         QSWDT(M2)=SWDT(M3)*.001
         QSWDT(M3)=SWDT(M2)*.001
         QSWDT(M4)=SWDT(M4)*.001
  103 CONTINUE
  104 RETURN
C
      END
```

```
      SUBROUTINE SEARCH
C
C . . . . . . THIS SUBROUTINE (SEARCH) IS CALLED BY THE MAIN PROGRAM
C         (PREGEN) TO SEARCH THE DATA FILE TO SEE IF AN UNKNOWN STATION
C        LIES IN THE AREA OF APPLICABILITY OF ANY KNOWN STATIONS AND
C        WHICH STATIONS ARE THE CLOSEST IN EACH OF FOUR QUADRANTS.
C
      INTEGER EOF,I,IJ,ILAT,ILEV,ILON,IO,IP,IS,IT,IY,IZ
      INTEGER I1,I2,I3,I4,J,K,KK,LAT,LEV,LLAT,LON,M,MJ
      INTEGER NAME(3),NE,NN,NS,NW,ONAM,READU,WRITEU
      REAL A,AA,AV,B,BB,OLAT,OLEV,OLON,ORSQ,OSWDT,RSQ
      REAL S,SWDT(15),TS
C
      COMMON /FILES/ READU,WRITEU
      COMMON /VIPS/ A(3),AA(9),AV(5),B(3),BB(9),ILAT,ILEV,ILON,IO,IP,IS,
     1IT,IY,IZ,KK(5),NE,NN,NS,NW,OLAT(5),OLEV(5),OLON(5),ONAM(5,3),ORSQ(
     25),OSWDT(75),RSQ,S(365,4),TS(365)
C
      DATA EOF/4H    /
C
      DO 101 I=1,5
  101 KK(I)=0
      IF (IS.EQ.1) WRITE (WRITEU,121)
C
C . . . . . . READ LATITUDE, LONGITUDE, ELEVATION AND WEATHER
C        PARAMETERS FOR ONE STATION.
C
  102 READ (READU,122) (NAME(I),I=1,3),NN,NS,NE,NW,LEV,LAT,LON
      IF (NAME(1).EQ.EOF) GO TO 112
      READ (READU,123) (SWDT(J),J=1,15)
      LAT=ICON(LAT)
      LON=ICON(LON)
      IF (LAT.EQ.ILAT.AND.LON.EQ.ILON) GO TO 113
      IF (IS.NE.1) GO TO 103
      WRITE (WRITEU,124) (NAME(I),I=1,3),LAT,LON,LEV
C
C . . . . . . DETERMINE IF THE STATION LIES WITHIN THE RADIUS OF
C        INTEREST.  ON THE AVERAGE 59 MILES = 1 DEGREE LONGITUDE AND
C        1 LATITUDE DEGREE = 1.32 AVERAGE LONGITUDE DEGREES.
C
  103 LLAT=(LAT-ILAT)*1.32
      RSQ=LLAT**2+(LON-ILON)**2
      RSQ=SQRT(RSQ)
C
C . . . . . . DETERMINE IF STATION IS WITHIN 2 DEGREES (118 MI.)
C        RADIUS.  LAT, LON READ IN I5 FORMAT.  THUS THEY ARE TOO BIG
C        BY A FACTOR OF 100.
C
      IF (RSQ-200.) 104,104,102
C
C . . . . . . DETERMINE IF UNKNOWN IN AREA OF APPLICABILITY OF KNOWN
C        STATION.
C
  104 IF (SWDT(1).NE.0..AND.SWDT(6).NE.0..AND.SWDT(11).NE.0.) GO TO 105
      WRITE (WRITEU,120) (NAME(I),I=1,3)
      GO TO 102
  105 CALL APPLI (LAT,LON,LEV,NAME,SWDT)
C
C . . . . . . DETERMINE WHICH QUADRANT IT IS IN AND IF IT IS THE
C        MINIMUM DISTANCE STATION IN THAT QUADRANT SO FAR.
C
      IF (LAT-ILAT) 106,106,108
  106 IF (LON-ILON) 107,107,111
  107 CALL MINUM (KK(3),LAT,LON,LEV,ILEV,OLAT(3),OLON(3),OLEV(3),OSWDT(3
     11),SWDT,RSQ,ORSQ(3),NAME,ONAM(3,1),ONAM(3,2),ONAM(3,3))
      GO TO 102
  108 IF (LON-ILON) 109,109,110
  109 CALL MINUM (KK(2),LAT,LON,LEV,ILEV,OLAT(2),OLON(2),OLEV(2),OSWDT(1
     16),SWDT,RSQ,ORSQ(2),NAME,ONAM(2,1),ONAM(2,2),ONAM(2,3))
      GO TO 102
  110 CALL MINUM (KK(1),LAT,LON,LEV,ILEV,OLAT(1),OLON(1),OLEV(1),OSWDT(1
     11),SWDT,RSQ,ORSQ(1),NAME,ONAM(1,1),ONAM(1,2),ONAM(1,3))
      GO TO 102
  111 CALL MINUM (KK(4),LAT,LON,LEV,ILEV,OLAT(4),OLON(4),OLEV(4),OSWDT(4
     16),SWDT,RSQ,ORSQ(4),NAME,ONAM(4,1),ONAM(4,2),ONAM(4,3))
      GO TO 102
  112 RETURN
  113 WRITE (WRITEU,125) (NAME(I),I=1,3)
      IF (SWDT(1).NE.0..AND.SWDT(6).NE.0..AND.SWDT(11).NE.0.) GO TO 114
      WRITE (WRITEU,119)
      GO TO 102
  114 WRITE (WRITEU,126)
      DO 115 I=1,15
  115 SWDT(I)=SWDT(I)*.001
      DO 116 I=1,11,5
          I1=I+1
          I2=I+2
          I3=I+3
          I4=I+4
          WRITE (WRITEU,127) SWDT(I),SWDT(I1),SWDT(I2),SWDT(I3),SWDT(I4)
  116 CONTINUE
      IF (IP.NE.1) RETURN
      K=0
      DO 118 I=1,7,3
          M=I+K
          AA(I)=SWDT(M)
```

```
         BB(1)=0.
         DO 117 J=1,2
            IJ=I+J
            MJ=M+J
            AA(IJ)=SWDT(MJ)
            MJ=MJ+1
            BB(IJ)=SWDT(MJ)
            M=M+1
  117    CONTINUE
         K=K+5
  118 CONTINUE
      CALL BEADAY
      RETURN
C
C . . . . . . FORMATS USED IN THIS SUBROUTINE.
C
  119 FORMAT (1H0,111HTHIS STATION DOES NOT HAVE A COMPLETE DATA SET - T
     1HE PROGRAM WILL ATTEMPT TO USE ANOTHER STATION OR INTERPOLATE)
  120 FORMAT (10X,9HSTATION  ,2A4,A2,86HIS WITHIN TWO DEGREE RADIUS, BUT
     1 DOES NOT HAVE A COMPLETE DATA SET - THEREFORE IGNORED)
  121 FORMAT (1H0,20X,17HSTATIONS SEARCHED)
  122 FORMAT (2A4,A2,9X,4(I3,1X),1X,I4,2(2X,I5))
  123 FORMAT (15F5.0)
  124 FORMAT (5X,2A4,A2,5X,6HLAT = ,I10,5X,6HLON = ,I10,5X,7HELEV = ,I10
     1)
  125 FORMAT (1H0,2A4,A2,47H IS A WEATHER STATION AT THIS LAT, LON AND E
     1LEV)
  126 FORMAT (//75H FOURIER COEFFICIENTS FOR CLIMATIC PARAMETERS, P(D/D)
     1, D(D), AND STORM SIZE/17X,2HA0,18X,2HA1,18X,2HB1,18X,2HA2,18X,2HB
     22)
  127 FORMAT (1H0,5(10X,F10.4))
C
      END


      SUBROUTINE SER (NT,A1,A,B,QUAD)
C
C . . . . . . THIS SUBROUTINE (SER) IS CALLED BY THE SUBROUTINES
C         (WAVG2 AND BEADAY) TO GENERATE A FOURIER SERIES REPRESENTATION
C         OF THE DAILY VALUES FOR A GIVEN PARAMETER OVER A YEARS TIME.
C
      INTEGER I,N,NT
      REAL A(3),A1,B(3),QUAD(365),SUM,XI,XN
C
      DO 102 I=1,365
         XI=FLOAT(I)
         SUM=0.
         DO 101 N=1,NT
            XN=FLOAT(N)
  101    SUM=SUM+A(N)*COS(.01725*XI*XN)+B(N)*SIN(.01725*XI*XN)
  102 QUAD(I)=A1+SUM
      RETURN
C
      END
```

```
      SUBROUTINE WAVG1
C
C . . . . . . . THIS SUBROUTINE (WAVG1) IS CALLED BY THE MAIN PROGRAM
C         (PREGED) TO COMPUTE THE WEIGHTING FACTORS FOR INTERPOLATION.
C
      INTEGER I,ILAT,ILEV,ILON,IO,IP,IS,IT,IY,IZ,KK,NF,NN
      INTEGER NS,NW,QNAM,READU,WRITEU
      REAL A,AA,AT,AV,B,BB,DT,ET,QLAT,QLEV,QLON,QMIND
      REAL QMINL,QRSQ,QSWDT,RSQ,S,TS,V(4)
C
      COMMON /FILES/ READU,WRITEU
      COMMON /VIRS/ A(3),AA(9),AV(5),B(3),BB(9),ILAT,ILEV,ILON,IO,IP,IS,
     1IT,IY,IZ,KK(5),NF,NN,NS,NW,QLAT(5),QLEV(5),QLON(5),QNAM(5,3),QRSQ(
     275),QSWDT(75),RSQ,S(365,4),TS(365)
C
      DT=0.
      AT=0.
      ET=0.
C
C . . . . . . . SUM DISTANCES.
C
      QMIND=0.
      DO 101 I=1,4
          IF (QRSQ(I).EQ.0.) GO TO 101
          IF (QMIND.EQ.0.) QMIND=QRSQ(I)
          QMIND=AMIN1(QMIND,QRSQ(I))
  101 CONTINUE
      DO 103 I=1,4
          IF (QRSQ(I).EQ.0.) GO TO 102
C
C . . . . . . . COMPUTE A SIMILARITY INDEX (DT) USING THE CLOSEST
C         STATION FROM ALL FOUR QUADRANTS AS THE BASIS.  SIMILARITY
C         INDEX = 2W/(A+B) WHERE W IS THE MINIMUM OF A AND B. IN THIS
C         CASE, THE MINIMUM DISTANCE STATION FROM ALL FOUR QUADRANTS.
C
          QRSQ(I)=2.*QMIND/(QMIND+QRSQ(I))
          DT=DT+QRSQ(I)
  102     V(I)=0.
  103 CONTINUE
      IF (DT) 104,104,105
  104 WRITE (WRITEU,115)
      RETURN
C
C . . . . . . . FIND RELATIVE DISTANCES.
C
  105 DO 106 I=1,4
  106 QRSQ(I)=QRSQ(I)/DT
C
C . . . . . . . SUM ELEVATIONS.  NOTE - A ZERO VALUE FOR ELEVATION
C         INDICATES NO STATION PRESENT IN THAT QUADRANT, NOT AN
C         ELEVATION OF ZERO.
C
      QMINL=0.
      DO 109 I=1,4
          IF (QLEV(I).EQ.0.) GO TO 109
          V(I)=ABS(QLEV(I)-FLOAT(ILEV))
          IF (V(I)) 107,107,108
  107     V(I)=1.
          QMINL=1.
          GO TO 109
  108     IF (QMINL.EQ.0.) QMINL=V(I)
          QMINL=AMIN1(QMINL,V(I))
  109 CONTINUE
      DO 110 I=1,4
C
C . . . . . . . COMPUTE SIMILARITY INDEX FOR ELEVATIONS (ET).
C
          IF (V(I).EQ.0.) GO TO 110
          V(I)=2.*QMINL/(QMINL+V(I))
          ET=ET+V(I)
  110 CONTINUE
C
C . . . . . . . FIND RELATIVE SIMILARITY IN ELEVATIONS.
C
      DO 111 I=1,4
  111 V(I)=V(I)/ET
C
C . . . . . . . FIND AVERAGE WEIGHTING FACTOR (AV) - AVERAGE OF TWO TIMES
C         ELEVATION FACTOR AND THE DISTANCE FACTOR (QSRQ).
C
      DO 112 I=1,4
  112 AV(I)=(2.*V(I)+QRSQ(I))/3.
C
C . . . . . . . NORMALIZE AVERAGE WEIGHTING FACTOR (AV).
C
      DO 113 I=1,4
  113 AT=AT+AV(I)
      DO 114 I=1,4
  114 AV(I)=AV(I)/AT
      RETURN
C
C . . . . . . . FORMATS USED IN THIS SUBROUTINE.
C
  115 FORMAT (1H0,47HNO STATIONS WITHIN TWO DEGREE RADIUS OF UNKNOWN)
C
      END
```

```
      SUBROUTINE WAVG2                                                  WAVG2001
C                                                                       WAVG2002
C . . . . . . THIS SUBROUTINE (WAVG2) IS CALLED BY THE MAIN PROGRAM     WAVG2003
C        (PREGEN) TO GENERATE DAILY VALUES FOR P(D/D), P(D), AND SS     WAVG2004
C        AS WELL AS FINDING THE WEIGHTED AVERAGE AND FINALLY RECOMPUTES WAVG2005
C        FOURIER COEFFICIENTS.                                          WAVG2006
C                                                                       WAVG2007
      INTEGER IA,IB,IER,IJ,ILAT,ILEV,ILON,IO,IP,IS,IT,IY                WAVG2008
      INTEGER IZ,J,J1,K,KK,L,LJ,M,NE,NN,NS,NT,NW,QNAM                   WAVG2009
      INTEGER READU,WRITEU                                              WAVG2010
      REAL A,AA,AV,A1,B,BB,QLAT,QLEV,QLON,QRSQ,QSWDT,RSQ                WAVG2011
      REAL S,TS                                                         WAVG2012
C                                                                       WAVG2013
      COMMON /FILES/ READU,WRITEU                                       WAVG2014
      COMMON /VIPS/ A(3),AA(9),AV(5),B(3),BB(9),ILAT,ILEV,ILON,IO,IP,IS,WAVG2015
     1IT,IY,IZ,KK(5),NE,NN,NS,NW,QLAT(5),QLEV(5),QLON(5),QNAM(5,3),QRSQ(WAVG2016
     25),QSWDT(75),RSQ,S(365,4),TS(365)                                 WAVG2017
C                                                                       WAVG2018
      L=1                                                               WAVG2019
      NT=2                                                              WAVG2020
C                                                                       WAVG2021
C . . . . . . LOOP FOR GENERATING THE THREE PARAMETERS, P(D/D), P(D),   WAVG2022
C        AND SS.                                                        WAVG2023
C                                                                       WAVG2024
      DO 105 IJ=1,11,5                                                  WAVG2025
         DO 101 K=1,365                                                 WAVG2026
  101    TS(K)=0.                                                       WAVG2027
C                                                                       WAVG2028
C . . . . . . LOOP FOR THE FOUR QUADRANTS.                              WAVG2029
C                                                                       WAVG2030
         DO 103 J=1,4                                                   WAVG2031
         IJJ=(J-1)*15+IJ                                                WAVG2032
         A1=QSWDT(IJJ)                                                  WAVG2033
            IA=15*(J-1)+IJ+1                                            WAVG2034
            IB=IA+2                                                     WAVG2035
C                                                                       WAVG2036
C . . . . . . GENERATE PARAMETER SERIES FOR EACH QUADRANT.              WAVG2037
C                                                                       WAVG2038
            CALL SER (NT,A1,QSWDT(IA),QSWDT(IB),S(1,J))                 WAVG2039
C                                                                       WAVG2040
C . . . . . . SUM WEIGHTED DAILY PARAMETER VALUES OVER THE FOUR         WAVG2041
C        QUADRANTS.                                                     WAVG2042
C                                                                       WAVG2043
            DO 102 K=1,365                                              WAVG2044
               TS(K)=TS(K)+S(K,J)*AV(J)                                 WAVG2045
  102       CONTINUE                                                    WAVG2046
  103    CONTINUE                                                       WAVG2047
C                                                                       WAVG2048
C . . . . . . GENERATE NEW FOURIER COEFFICIENT FOR ITH PARAMETER IN     WAVG2049
C        SUBROUTINE (FORIT).                                            WAVG2050
C                                                                       WAVG2051
         N=182                                                          WAVG2052
         M=2                                                            WAVG2053
         CALL FORIT (TS,N,M,A,B,IER)                                    WAVG2054
C                                                                       WAVG2055
C . . . . . . WRITE TO OUTPUT.                                          WAVG2056
C                                                                       WAVG2057
         IF (IJ.EQ.1) WRITE (WRITEU,106)                                WAVG2058
         WRITE (WRITEU,107) A(1),(A(J),B(J),J=2,3)                      WAVG2059
         AA(L)=A(1)                                                     WAVG2060
         BB(L)=0.                                                       WAVG2061
         DO 104 J=1,2                                                   WAVG2062
            LJ=L+J                                                      WAVG2063
            J1=J+1                                                      WAVG2064
            AA(LJ)=A(J1)                                                WAVG2065
  104    BB(LJ)=B(J1)                                                   WAVG2066
         L=L+3                                                          WAVG2067
  105 CONTINUE                                                          WAVG2068
      RETURN                                                            WAVG2069
C                                                                       WAVG2070
C . . . . . . FORMATS USED IN THIS SUBROUTINE.                          WAVG2071
C                                                                       WAVG2072
  106 FORMAT (//75H FOURIER COEFFICIENTS FOR CLIMATIC PARAMETERS, P(D/D)WAVG2073
     1, P(D), AND STORM SIZE/17X,2HA0,18X,2HA1,18X,2HB1,18X,2HA2,18X,2HBWAVG2074
     22)                                                                WAVG2075
  107 FORMAT (1H0,5(10X,F10.4))                                         WAVG2076
C                                                                       WAVG2077
      END                                                               WAVG2078
```

APPENDIX #6

Coefficient Determination for Sine Function Weather Generation

An algorithm follows which enables a user to determine the input parameters for the RANGES model which will determine the coefficients for either the temperature or precipitation sine functions.

(i) Compute average monthly values for either temperature or precipitation (Fig. 1).

(ii) Plot the monthly values for a year. The result will probably give data that has the form of a sine function. If not, then this method may not be applicable.

(iii) Find the difference between the high and low average monthly values (e.g. 2.8 - .2 = 2.6). This is twice the amplitude of the sine wave. Divide the difference by two to find the amplitude H1(H1 = 2.6/2 = 1.3).

(iv) Add the low average monthly value to the amplitude and construct a horizontal line (L) at this height (2.6/2 + .2 - 1.5).

(v) Find the point where the sine wave crosses line L going up from left to right, and determine the number (H2) of the day of the year (Julian day) where this occurs (H2 = 110 days).

(vi) Set H3 equal to the low average monthly value (H3 = .2).

(vii) These three parameters will be read into the model and take their corresponding places in the following equation:

```
Y = H1 * (SIN ((DDAY - H2) * .0172) + 1.) + H3
```

where in the example the expanded equation is:

```
PPT = 1.3 * (SIN ((DDAY - 110) * .0172) + 1.) + .2
```

The monthly values predicted by this equation are represented by o in Fig. 1.

If the predicted monthly values are summed, they may over or underestimate the average yearly precipitation. Therefore, it may be necessary to increase or decrease the amplitude of the sine wave to obtain a more realistic yearly average. In our example, precipitation is overestimated, so the following procedure is used.

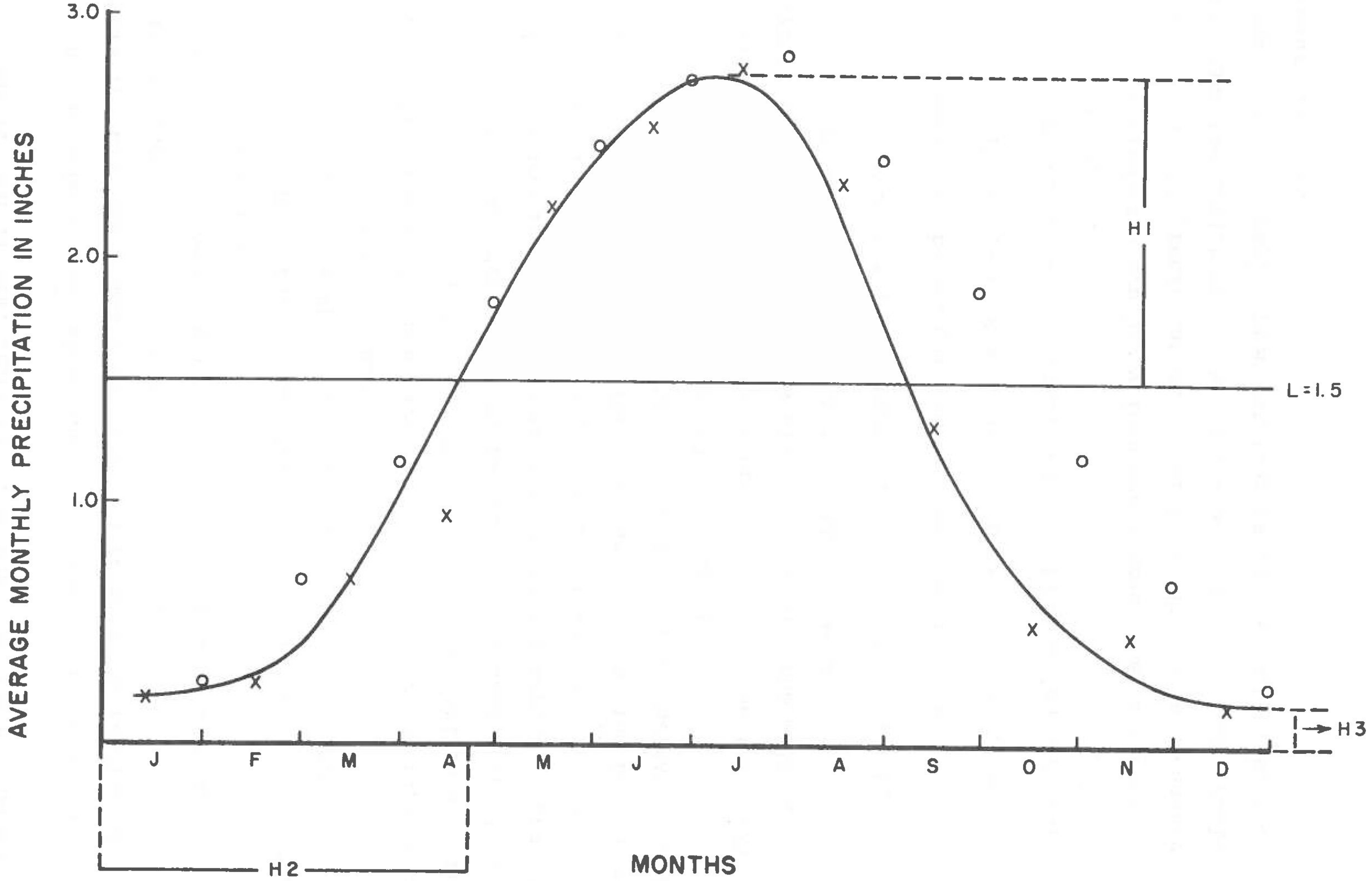


Fig. 1. Comparison of sine function prediction (o) and actual (x) monthly precipitation amounts for the Pawnee Site.

The steps to improve sine function estimate of average yearly precipitation are:

(i) Determine a new axis based on a lower maximum value (L = (2.6 - .2) /2 + .2 = 1.4).

(ii) Find the new amplitude (H1 = 2.4/2 = 1.2).

(iii) Substitute the new H1 into the sine function and leave H2, H3 as before.

(iv) Check the new estimate (Fig. 2).

Again, the yearly estimate is too high. Therefore, another iteration is required to converge on an adequate function. Therefore, determine a new axis (L = (2.2 - .2)/2 + .2 = 1.2). Find a new H1, H1 = 2.0/2 = 1.0), and recompute the estimates. Now, the yearly average of the sine function is 14.6" compared with 14.9" computed from the data (Fig. 3). It is assumed close enough, although there is some underestimation in the summer and overestimation in the fall.

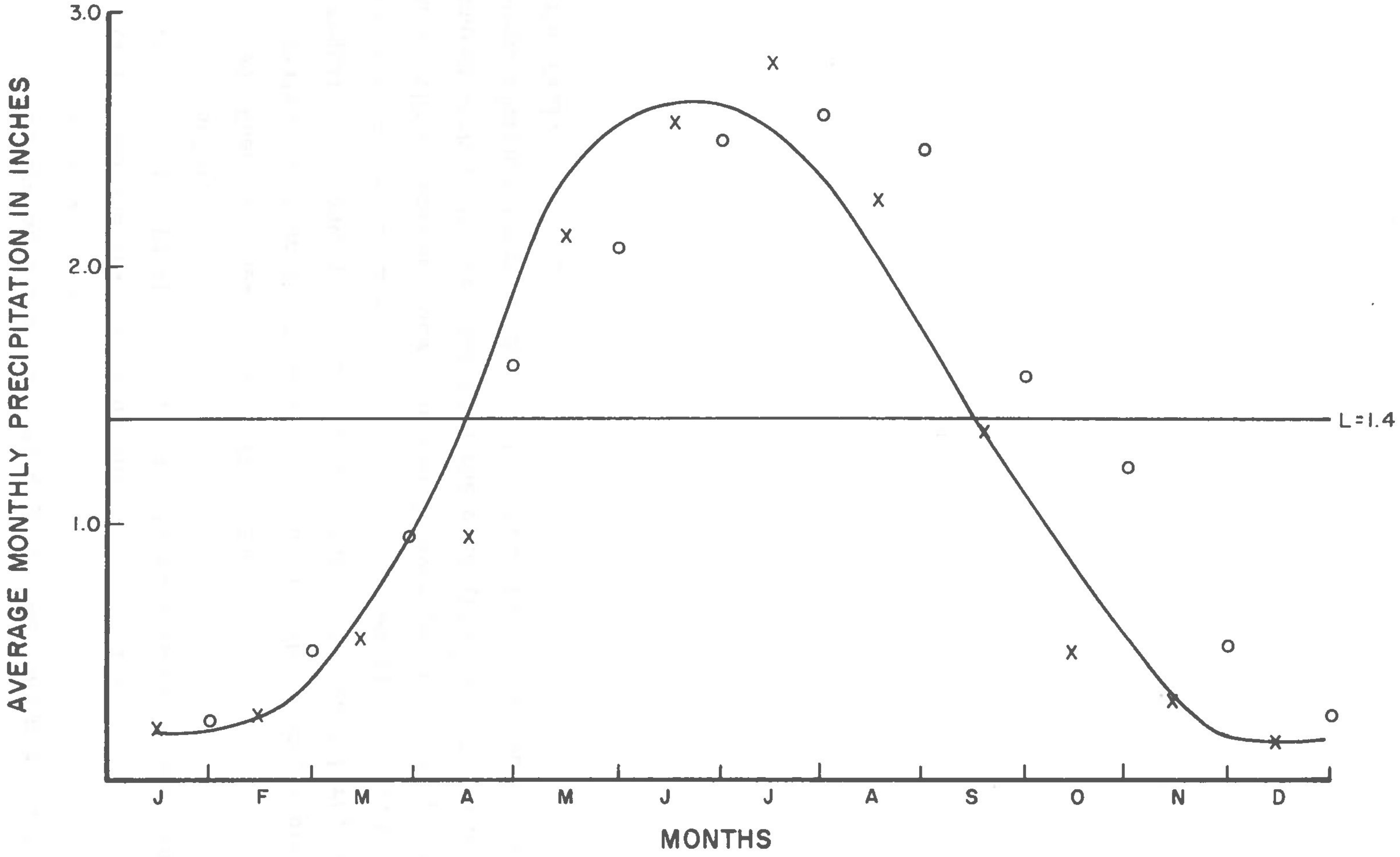


Fig. 2. Comparison of sine function prediction (o) and actual (x) monthly precipitation amounts for the Pawnee Site.

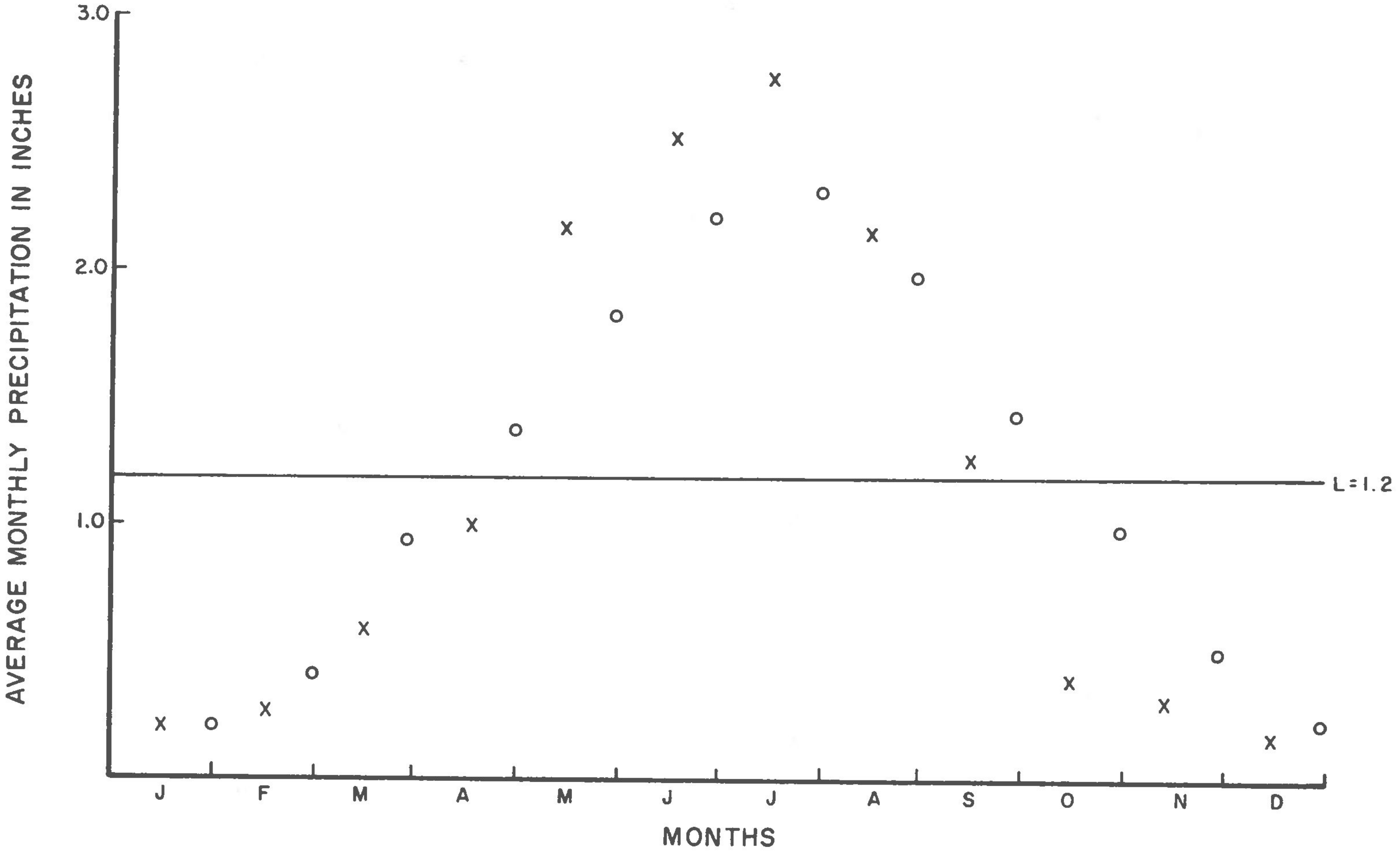


Fig. 3. Comparison of sine function prediction (o) and actual (x) monthly precipitation amounts for the Pawnee Site.