1976

Programming Phase of Water Response Ecosystem Model: I. Introduction and Support Programs

P. W. Lommen

1975/76 PROGRESS REPORT

PROGRAMMING PHASE OF WATER RESPONSE ECOSYSTEM MODEL: I. INTRODUCTION AND SUPPORT PROGRAMS

P. W. Lommen*
Utah State University
(*now at HDR Ecosciences, Santa Barbara, California)

US/IBP DESERT BIOME
RESEARCH MEMORANDUM 76-36

in

REPORTS OF 1975 PROGRESS
Volume 1: Central Office, Modeling
Modeling Section, pp. 1-9

1975/76 Proposal No. 2.1.3

Printed 1976

Citation format: Author(s). 1976. Title.
US/IBP Desert Biome Res. Memo. 76-36.
Utah State Univ., Logan. 9 pp.

Ecology Center, Utah State University, Logan, Utah 84322

This report describes a portion of the Desert Biome Water Response Ecosystem Model. Five Research Memoranda comprise the full description: Introduction and support programs (RM 76-36); Abiotic submodels (RM 76-37); Animal submodel (RM 76-38); Perennial plant, nitrogen and decomposition submodels (RM 76-39); and Annual plant submodel (RM 76-40). The objectives of the Water Response Model, information on the arrangement of material distributed among the five Research Memoranda and descriptions of program MAIN and support programs F1, F3 and FTAVE are contained in Research Memorandum 76-36, **Programming phase of water response ecosystem model: I. Introduction and support programs.** The relationships between various sections of the model, their interactions and location in the report series are summarized in Table 1 of RM 76-36.

INTRODUCTION

As pointed out by Wilkin et al. (1975), the Water Response Ecosystem Model is a compromise between "general-purpose" and "question-oriented" models. It is an ecosystem-level model constrained in its design to answer the following a priori question:

> What is the **effect** on the annual, above-ground **phytomass** on the five validation sites of **increasing** or **decreasing** the annual **water input** above or below the **long-term** pattern now prevailing?

For further details, definitions and constraints implied by this question see Wilkin et al. (1975).

ARRANGEMENT OF MATERIAL

The full description of the Water Response Ecosystem Model comprises five Research Memoranda. This particular memorandum provides background information including model objective, arrangement of material in these five memoranda and naming convention for FORTRAN variables. It also briefly describes the MAIN bookkeeping program, and fully describes mathematical support programs F1, F3 and FTAVE. (These last three programs are described here because of their frequent use by various submodels.)

The other four Research Memoranda are: Abiotic Submodels (RM 76-37); the Animal Submodel (RM 76-38); the Perennial Vegetation, Nitrogen and Decomposition Submodels (RM 76-39); and the Annual Plant Submodel (RM 76-40). Each of these reports gives general and detailed descriptions of the submodel(s) included, as well as mathematical relationships and a program listing. Each general description shows the biological and/or abiotic assumptions, ideas and facts which make up the submodel. These general descriptions do not require a knowledge of computer languages. The detailed descriptions document the translation of assumptions, ideas and facts into computer code (FORTRAN) so that the reader can use and/or change the code if desired. Thus, details are given on the submodels themselves, as well as on subsections of them, and on the mathematical support programs for them. Key derivations are also given. Submodels, subsections and support programs and their interrelationships are found in Table 1.

A naming convention for FORTRAN variables, which is used in submodels of the Water Response Ecosystem Model, follows.

NAMING CONVENTION OF FORTRAN VARIABLES

In order to increase convenience and reduce confusion, a convention and hierarchy in the naming of variables in the submodels are used. All names of state, communication and driving variables follow the convention, as do most (but not all) of the names of the temporary variables and parameters.

Table 1. Submodels of Water Response Ecosystem Model showing associated subsections and mathematical support programs, with the relevant Research Memoranda numbers

Submodel and RM number	FORTRAN name	Subsections of submodel	Mathematical support programs (submodel and/or subsection using support program in parentheses)*
Annual plant biomass (76-40)	EXOTIC	RUUT TEAVG	F1 (EXOTIC) F3 (EXOTIC)
Perennial plant biomass (76-39)	VEG	VPHEN VGROW VTRANS VDETH	F1 (VEG) F3 (VEG) FTAVE (VEG, VPHEN) FWVP (VEG, VTRANS)
Animal dynamics (76-38)	ANML		
Nitrogen cycle (76-39)	N		F1 (N)
Decomposition (76-39)	DCMP		F1 (DCMP)
Weather generator (76-37)	PSWG	EVAP RINT	F1 (EVAP, RINT) IPROB (PSWG) RNOR (PSWG)
Soil temperature profile (76-37)	HEAT		FTAVE (HEAT) TDM (HEAT)
Soil water potential profile (76-37)	WATER		TDM (WATER) WBAL (WATER) WTIME (WATER)

*Since programs F1, F3 and FTAVE are widely used they are all described in detail in the introductory Research Memorandum (76-36). Each of the other six support programs listed in this column is described in the Research Memorandum which also describes the submodel for which it is a support program (RM numbers given in left-hand column of this table).

Table 2. Convention and hierarchy utilized to name variables

First Letter of Name	Type of Variable
X	State variable
C	Communication variable
T	Temporary (internal) variable
Z	Driving variable
P	Parameter

Table 3. Characteristic letters for submodels (second letter of name)

Submodel	FORTRAN Name	Characteristic Letter
Annual plant biomass	EXOTIC	E
Perennial plant biomass	VEG	V
Animal dynamics	ANML	A
Nitrogen cycle	N	N
Decomposition	DCMP	D
Soil temperature profile	HEAT	H
Soil water potential profile	WATER	W

The first letter of a variable name is either *X*, *C*, *T*, *Z* or *P*, whose meanings are shown in Table 2, along with the hierarchy used if a variable has more than one use. The second letter of a variable name (with the exception of driving variables) is the characteristic letter of the submodel of origin (Table 3). The remaining letters of the name are chosen to be a phonetic representation of the variable.

DESCRIPTION OF PROGRAM MAIN

MAIN is chiefly a bookkeeping program. It is large (about 1800 records) and was written by Jon D. Gustafson, Natural Resources Ecology Laboratory, Colorado State University, Ft. Collins, CO 80523. It will be only briefly described here.

MAIN performs four functions for the model. First, it performs necessary initializations by reading some data itself and by causing the initialization sections of the submodels to do their tasks. Second, MAIN causes each submodel to be executed once each time-step. The submodels determine the change during the time-step of every state variable. They then add these changes to the state variables. Third, by keeping track of time, MAIN determines current Julian date during a simulation and stops the simulation after a specified number of days have elapsed. Fourth, it prints debugging information at specified time intervals and produces graphs at the end of the simulation.

DESCRIPTION OF SUPPORT PROGRAMS

Mathematical Support Program F1

Function *F*1 is a simple linear interpolation program over one independent variable. It is supplied with two or more pairs of data points ($x_1, y_1; x_2, y_2; \ldots$). Then, given a value of x, say x^*, it finds a corresponding value for y^* by looking through the pairs of data points for two adjacent values of x, x_J and x_{J+1}, such that $x_J < x^* < x_{J+1}$, and then calculates y^* by Equation 1:

$$y^* = y_J + [(y_{J+1} - y_J)/(x_{J+1} - x_J)]\,(x^* - x_J) \qquad (1)$$

The values of x must be arranged $x_1 < x_2 < x_3 \ldots$ If there are N pairs of data points and $x^* \leqslant x_1$, then $y^* = y_1$; if $x^* > x_n$, then $y = y_n$.

The FORTRAN variable VALUE is called x^* above. Values of $x, x_2, \ldots, x_n$ are in array DTAPTS ($1 \rightarrow N$, 1); corresponding y values are in DTAPTS ($1 \rightarrow N$, 2).

```
IF (VALUE .GE. DTAPTS(NPTS,1)) GO TO 35        F1  14
IF (VALUE .LE. DTAPTS(1,1)) GO TO 40           F1  15
```

Check if VALUE is greater than or equal to largest x (if yes, proceed to statement 35), and check if VALUE is less than or equal to smallest x (if yes, go to statement 40).

```
      DO 30 J=1,NPTS-1
      IF ((VALUE .GE. DTAPTS(J,1)) .AND. (VALUE .LE.
     C DTAPTS(J+1,1))) GO TO 60
  030 CONTINUE
```

We reach here if VALUE is within normal range of x values. Go through this loop until J is found such that $x_J \leqslant$ VALUE $\leqslant x_{J+1}$. Then go to statement 60.

```
  035 F1=DTAPTS(NPTS,2)
      GO TO 70
```

Reach here only if VALUE $\geqslant x_n$. Set $F1 = y_n$ and return. $F1$ is called y^* above.

```
  040 F1=DTAPTS(1,2)
      GO TO 70
```

If VALUE $\leqslant x_1$, set $F1 = y_1$ and return.

```
  060 P=(DTAPTS(J+1,2)-DTAPTS(J,2))/(DTAPTS(J+1,1)-DTAPTS(J,1))
      B=DTAPTS(J,2)-P*DTAPTS(J,1)
      F1=P*VALUE+B
```

Interpolate between adjacent data points according to Equation 1, above, and return.

Mathematical Support Program F3

Function $F3$ is a simple linear interpolation program over two independent variables. It is supplied with a family of curves of z vs. x for two or more y values. Then, given values of x and y, say x^* and y^*, it interpolates between adjacent x values and between adjacent y curves to find z^*. All z vs. x curves must have the same set of x values and, as with $F1$, $x_1 < x_2 < \ldots < x_{nr-1}$. Also, $y_1 < y_2 < \ldots < y_{nc-1}$.

```
      FUNCTION F3(X, Y, DATA51, NR, NC)
```

X and Y are called x^* and y^* above. DATA51 is an NR by NC data array containing the family of $F3$ (called z above) vs. x curves for two or more y values. The first column contains a strange number in DATA51 (1,1) as a reminder that the position is not used, and a set of x values in DATA51 ($2 \rightarrow NR$, 1), the smallest value first and in increasing order. Column 2 contains the lowest y value in DATA51 (1, 2) and then a set of values for $F3$ which correspond to this y value and to the set of x values in column 1. Column 3 contains the next larger y value and the set of $F3$ values which correspond to the y value and to the x value in column 1. Column 4 contains the next larger value of y, etc., up through all $NC-1$ values of y.

```
      IF(X .GT. DATA51(2,1)) GO TO 50
      F3=DATA51(2,2)
      RETURN
```

If $X \leqslant x_1$, $F3 =$ value at x_1 on first y curve.

```
50 IF(X .LT. DATA51(NR,1)) GO TO 60
   F3=DATA51(NR,2)
   RETURN
```

If $X \geqslant x_{nr-1}$, $F3$ = value at x_{nr-1} on first y curve.

```
60 IF(Y .GT. DATA51(1,2)) GO TO 70
   YOUT = .TRUE.
   J=2
```

If $Y \leqslant y_1$, set logical variable YOUT equal to true (Y is outside normal range) and $J = 2$, which means value of $F3$ will be found below from values in second column of DATA51, which correspond to lowest value of y.

```
70 IF(Y .LT. DATA51(1,NC)) GO TO 100
   YOUT = .TRUE.
   J = NC
```

If $Y \geqslant y_{nc-1}$, set YOUT equal to true and $J = NC$, which means value of $F3$ will be found below from values in last column of DATA51, which correspond to largest value of y.

```
    DO 130 I=1,NR-1
130 DATA52(I) = DATA51(I+1, 1)
```

Set up use of function $F1$ by loading x values from DATA51 in first $NR-1$ places of array DATA52.

```
    IF(.NOT. YOUT) GO TO 190
    DO 160 I=NR, 2*(NR-1)
160 DATA52(I) = DATA51( I+2-NR, J)
    F3 = F1(X, DATA52, NR-1)
    RETURN
```

Execute these lines only if Y is outside normal range of values. Here the next $NR-1$ places in DATA52 are loaded with the curve from DATA51, which is in location DATA51 ($2 \rightarrow NR$, J). Then the value of $F3$ is found by function $F1$.

```
190 J=3
200 IF(Y .LE. DATA51(1,J)) GO TO 205
    J=J+1
    GO TO 200
```

We reach here only if $y_1 < Y < y_{nc-1}$. The value of J is sought such that $y_{J+2} < Y \leqslant y_{J-1}$ (remember that y_J is in column $J+1$ of DATA51).

```
205 DO 210 I=NR, 2*(NR-1)
210 DATA52(I) = DATA51(I+2-NR, J)
    F3H = F1(X, DATA52, NR-1)
```

If the proper J has been found, reload the second $NR-1$ places in DATA51 and calculate $F3H$ with $F1$. $F3H$ is the first of two quantities which will be used to finally determine $F3$.

```
      DO 240 I=NR, 2*(NR-1)
  240 DATA52(I) = DATA51(I+2-NR, J-1)
      F3L = F1(X, DATA52, NR-1)
```

F3 45
F3 46
F3 47

Reload DATA52 with the curve for the next lower value of y and calculate $F3L$ with $F1$.

```
      B = (Y - DATA51(1,J-1)) / (DATA51(1,J) - DATA51(1,J-1))
      F3 = F3L + (F3H - F3L) * B
```

F3 48
F3 49

Interpolate between $F3L$ and $F3H$ to find $F3$.

Mathematical Support Program FTAVE

Function FTAVE computes average air temperature over the previous NDAYS. The only complicating factor is that submodel PSWG supplies one average air temperature every PMDT days, and NDAYS may not be an integer multiple of PMDT.

This routine is used: 1) by HEAT to compute a 30-day average temperature for soil temperature at 60 cm; 2) by VEG to compute a 15-day average temperature for determining optimum net photosynthesis temperatures; and 3) by VPHEN to determine a 10-day average, which is then used to test for temperature limits and/or thresholds for switching phenophases.

```
      RPMDT=PMDT
      RNDAYS=NDAYS
      R=RNDAYS/RPMDT
      NR=R
```

FTAVE 17
FTAVE 18
FTAVE 19
FTAVE 21

Make real numbers of PMDT, NDAYS and their ratio. NR is largest whole number of time-steps in NDAYS.

```
      SUM=0.0
      DO 100 I=1, NR
  100 SUM = SUM + ZHAIRT(I)
```

FTAVE 22
FTAVE 24
FTAVE 25

Add together the first NR elements of the ZHAIRT array. Present and past time-step temperatures are stored in ZHAIRT, most recent first, as far back as 30 days.

```
      RNR=NR
      SUM=SUM + ZHAIRT(I)*(R-RNR)
```

FTAVE 28
FTAVE 30

If NDAYS is not an integer multiple of PMDT, then only a portion of the oldest time-step temperature is needed.

```
      FTAVE=SUM/R
```

FTAVE 32

This is the calculation of the average temperature.

COMPLETE PROGRAM LISTING

Function F1

```
      FUNCTION F1(VALUE,DTAPTS,NPTS)
C
C     MARCH 1976        PAUL W. LOMMEN
C
C  GIVEN SOME PAIRS OF DATA POINTS THE FUNCTION SIMPLY INTERPOLATES
C  BETWEEN THEM.  VALUES OF THE INDEPENDENT VARIABLE OUTSIDE THE RANGE
C  ARE SET EQUAL TO FIRST Y VALUE OR LAST Y VALUE, DEPENDING ON IF THE
C  INDEPENDENT VARIABLE  IS BELOW OR ABOVE THE RANGE OF X.    DATA
C  POINTS MUST BE IN ORDER OF INCREASING X.
C  MAY NOT HAVE TWO IDENTICAL X  VALUES IN DATA POINTS
C
      DIMENSION DTAPTS(NPTS,2)
C
      IF (VALUE .GE. DTAPTS(NPTS,1)) GO TO 35
      IF (VALUE .LE. DTAPTS(1,1)) GO TO 40
C
      DO 30 J=1,NPTS-1
      IF ((VALUE .GE. DTAPTS(J,1)) .AND. (VALUE .LE.
     C DTAPTS(J+1,1))) GO TO 60
  030 CONTINUE
C
  035 F1=DTAPTS(NPTS,2)
      GO TO 70
C
  040 F1=DTAPTS(1,2)
      GO TO 70
C
  060 P=(DTAPTS(J+1,2)-DTAPTS(J,2))/(DTAPTS(J+1,1)-DTAPTS(J,1))
      B=DTAPTS(J,2)-P*DTAPTS(J,1)
      F1=P*VALUE+B
   70 CONTINUE
C
      RETURN
      END
```

Function F3

```
      FUNCTION F3(X, Y, DATA51, NR, NC)
C  THIS IS A TWO DIMENSIONAL LINEAR INTERPOLATION FUNCTION
C  AUGUST 1976     PAUL W. LOMMEN
      DIMENSION DATA51(NR, NC), DATA52(30)
      LOGICAL YOUT
      YOUT=.FALSE.
C
C  CHECK IF VALUES WITHIN RANGE
      IF(X .GT. DATA51(2,1)) GO TO 50
      F3=DATA51(2,2)
      RETURN
   50 IF(X .LT. DATA51(NR,1)) GO TO 60
      F3=DATA51(NR,2)
      RETURN
   60 IF(Y .GT. DATA51(1,2)) GO TO 70
      YOUT = .TRUE.
      J=2
      GO TO 100
   70 IF(Y .LT. DATA51(1,NC)) GO TO 100
      YOUT = .TRUE.
      J = NC
  100 CONTINUE
C
C  IF WE REACH HERE WE WILL BE CALLING F1 AT LEAST ONCE
C  SET UP CALL TO F1 BY LOADING X VALUES IN DATA52
      DO 130 I=1,NR-1
  130 DATA52(I) = DATA51(I+1, 1)
C
      IF(.NOT. YOUT) GO TO 190
      DO 160 I=NR, 2*(NR-1)
  160 DATA52(I) = DATA51( I+2-NR, J)
      F3 = F1(X, DATA52, NR-1)
      RETURN
C
C
C  NOW START HUNTING FOR PROPER RANGE OF Y
  190 J=3
  200 IF(Y .LE. DATA51(1,J)) GO TO 205
      J=J+1
      GO TO 200
```

```
C
  205 DO 210 I=NR, 2*(NR-1)
  210 DATA52(I) = DATA51(I+2-NR, J)
      F3H = F1(X, DATA52, NR-1)
      DO 240 I=NR, 2*(NR-1)
  240 DATA52(I) = DATA51(I+2-NR, J-1)
      F3L = F1(X, DATA52, NR-1)
      B = (Y - DATA51(1,J-1)) / (DATA51(1,J) - DATA51(1,J-1))
      F3 = F3L + (F3H - F3L) * B
      RETURN
C
      END
```

Function FTAVE

```
      FUNCTION FTAVE(PMDT, NDAYS)
C  MARCH  1976     PAUL LOMMEN
C  THIS FUNCTION COMPUTES AVERAGE AIR TEMPERATURE OVER PREVIOUS NDAYS.
C
C  ZHAIRT IS ARRAY HOLDING THM PRESENT AND PAST TIME STEP AVERAGE AIR
C  TEMPERATURES.   THM, REMEMBER, IS THE LARGEST FULL INTEGER NUMBER OF
C  TIME STEPS IN 30 DAYS.   ZHAIRT(1) HOLDS PRESENT TIME STEP
C  TEMPERATURE (ZAIRT).   ZHAIRT(2) HOLDS PREVIOUS VALUE OF ZAIRT, ETC.
C
      INTEGER PMDT, NDAYS
C
C  FT IS IN COMMON WITH HEAT
      COMMON/FT/ZHAIRT(31)
C
C  THERE SEEMS TO BE A LOT OF CONVERTING BACK AND FORTH BETWEEN REAL AND
C  INTEGER VARIABLES IN THIS FUNCTION
      RPMDT=PMDT
      RNDAYS=NDAYS
      R=RNDAYS/RPMDT
C
      NR=R
      SUM=0.0
C
      DO 100 I=1, NR
  100 SUM = SUM + ZHAIRT(I)
C
C  PICK UP APPROPRIATE FRACTION OF OLDEST TIME STEP TEMPERATURE
      RNR=NR
C
      SUM=SUM + ZHAIRT(I)*(R-RNR)
C
      FTAVE=SUM/R
C
      RETURN
      END
```

LITERATURE CITED

D. C. Wilkin, P. W. Lommen, J. V. Robinson, and G. S. Innis. 1975. Conceptualization and early development of a Water Response Ecosystem Model. US/IBP Desert Biome Res. Memo. 75-49. Utah State Univ., Logan. 21 pp.