2-1972

Annuals

Not Listed

MODELLING REPORT SERIES NUMBER 14

ANNUALS

VERSION 1

DESERT BIOME
UTAH STATE UNIVERSITY
LOGAN, UTAH 84321
FEBRUARY, 1972

THE PREPARATION OF THIS MODEL WAS WHOLLY SUPPORTED THROUGH THE US/IBP DESERT BIOME PROGRAM, UNDER GRANT # GB 15886 FROM THE NATIONAL SCIENCE FOUNDATION.

I N T R O D U C T I O N

Reports in this series are intended for internal use by Desert Biome collaborators. They are not to be quoted or referred to in formal publications. These reports have been produced by the Desert Biome Modelling Group, with the assistance of participants in the Desert Biome and other researchers.

The main function of the models, at this stage of their development, is to provide guidance in the research efforts of the Biome. Therefore, it will be noted that most of the information which they contain is fragmentary evidence, best available estimates, arbitrary assumptions or non-Biome supported research. The collection and incorporation of more accurate data will come after these models have been prepared in this form. Validation of the models will also come later.

Any use of the models must recognize the limitations imposed by their development at this early stage of research.

(1) Biological interpretations must be performed with extreme caution. Output, for example, should be viewed in relation to system behavior (stability, general time relationships, relative magnitude of the variables, general responses to parameter modifications, etc.). These properties should be related to the processes incorporated in the model structure. No particular significance should be attached to the specific numbers given as output.

(2) Data included in these models must not be used without explicit approval of the investigators who have supplied them to us. Please contact the Desert Biome Central Office for details.

(3) The material contained in the models does not constitute publication. It is subject to revision. The modeling group requests that this material not be cited without their expressed permission.

As particular models are revised we will be re-issuing them in new versions. The versions will be numbered according to the general scheme:

Version 1. Models which have been developed by the modeling group in isolation from subject area specialists who have provided the question which has been modeled.

Version 2. Models revised to incorporate subject-areas specialist's criticisms.

Version 3. Models revised to incorporate finds of biome-sponsored research.

```
1       ANNUALS: PROC OPTIONS(MAIN);
        /*************************************************************************/
        /*                                                                       */
        /*          'ANNUALS' WAS BUILT IN RESPONSE TO FOUR RELATED              */
        /*      QUESTIONS (SEE MODELLING REPORT SERIES NUMBER 11,                */
        /*      'QUESTIONS, VERSION 1'):                                         */
        /*                                                                       */
        /*      AL0201  HOW DO THE SMALL MAMMALS DIVIDE UP THE FOOD              */
        /*              RESOURCES AVAILABLE?                                     */
        /*                                                                       */
        /*      AL0202  WHAT ARE THE COMPARATIVE ROLES OF THE VEGETATION         */
        /*              VS. SEEDS OF ANNUALS FOR USE BY RODENTS?                 */
        /*                                                                       */
        /*      AL0830  WHAT IS THE EFFECT OF VARYING BIOMASS GROWTH OF          */
        /*              ANNUALS ON THEIR SEED PRODUCTION?                        */
        /*                                                                       */
        /*      AL0901  WHAT IS THE EFFECT OF THE TIMING AND QUALITY OF          */
        /*              ANNUALS ON RODENT REPRODUCTION?                          */
        /*                                                                       */
        /*      'ANNUALS' IS THE FIRST HALF OF THE ANNUALS-RODENT MODEL          */
        /*      NEEDED TO ANSWER THESE QUESTIONS.  THE ANNUALS GERMINATE,        */
        /*      GROW, AND SET SEED IN RESPONSE TO SEASON CHANGES,                */
        /*      TEMPERATURE, RAINFALL (FOR GERMINATION), AND SOIL MOISTURE       */
        /*      (FOR GROWTH).  THE EFFECTS OF HERBIVORY AND GRANIVORY ON         */
        /*      THE POPULATION DYNAMICS OF THE ANNUALS WILL BE SIMULATED         */
        /*      WHEN A RODENT MODEL HAS BEEN BUILT AND INTERFACED WITH           */
        /*      THIS MODEL.                                                      */
        /*                                                                       */
        /*************************************************************************/
        /*                                                                       */
```

```
/*                                                                    */
/**********************************************************************/
/*                                                                    */
/*          THIS PROGRAM SIMULATES THE YEARLY GROWTH AND SEED         */
/*      PRODUCTION OF TWO SETS OF DESERT ANNUALS (SUMMER ANNUALS      */
/*      AND WINTER ANNUALS), WHICH HAVE THE FOLLOWING BASIC LIFE      */
/*      CYCLES:                                                       */
/*                                                                    */
/*          SUMMER ANNUALS GERMINATE IN THE LATE SPRING OR            */
/*          SUMMER (DEPENDING ON THE SITE LOCATION), USUALLY          */
/*          GROW FOR 2-3 MONTHS, AND THEN SET SEED AND DIE IN         */
/*          LATE SUMMER OR AUTUMN.                                    */
/*                                                                    */
/*          WINTER ANNUALS MAY GERMINATE EITHER IN AUTUMN OR          */
/*          EARLY SPRING, DEPENDING ON LOCATION AND THE SEASONAL      */
/*          RAINFALL PATTERN.  IF AUTUMN GERMINATION OCCURS,          */
/*          THE PLANTS OVER-WINTER AS SMALL ROSETTES.  ACTIVE         */
/*          GROWTH OCCURS IN THE SPRING, WITH SEED SET AND DEATH      */
/*          OCCURING IN LATE SPRING OR EARLY SUMMER.                  */
/*                                                                    */
/*          THE PROGRAM HAS A STANDARD WEEKLY 'DO' LOOP NESTED        */
/*      INSIDE A 'RAINFALL EVENT' LOOP.  AT THE START OF EACH YEAR    */
/*      DURING THE SIMULATION, THE WEEKS WHICH CONSTITUTE THE         */
/*      BEGINNING OF 'SPRING', 'SUMMER', 'AUTUMN', AND 'WINTER',      */
/*      RESPECTIVELY, ARE READ IN FOR THAT YEAR.  FOR A GIVEN YEAR,   */
/*      THE WEEK AND AMOUNT OF THE FIRST RAINFALL EVENT ARE READ IN.  */
/*      THIS DETERMINES THE VALUE OF THE WEEK_OF_NEXT_RAIN.  THE      */
/*      'DO' LOOP THEN ADVANCES UNTIL THAT RAINFALL EVENT OCCURS      */
/*      AND ITS AFFECT ON THE ANNUALS IS SIMULATED.  THAT WEEK THEN   */
/*      BECOMES THE WEEK_OF_LAST_RAIN.  THEN THE WEEK AND THE         */
/*      AMOUNT OF THE NEXT RAINFALL EVENT ARE READ IN, A NEW          */
/*      WEEK_OF_NEXT_RAIN IS COMPUTED, AND THE 'DO' LOOP THEN         */
/*      ADVANCES FROM WEEK_OF_LAST_RAIN TO THE WEEK_OF_NEXT_RAIN.     */
/*      THIS PROCESS OF READING IN THE WEEK AND THE AMOUNT OF THE     */
/*      NEXT RAINFALL EVENT CONTINUES THROUGH THE REMAINDER OF        */
/*      THAT YEAR.                                                    */
/*                                                                    */
/**********************************************************************/
/*                                                                    */
```

```
/*                                                                    */
/**********************************************************************/
/*                                                                    */
/*          THE ASSUMPTIONS USED IN THIS PROGRAM ARE AS FOLLOWS       */
/*      AND WERE MADE WITH TUCSON, ARIZONA IN MIND:                   */
/*                                                                    */
/*      1.  WINTER RAINS ARE FRONTAL AND OF LOWER INTENSITY THAN      */
/*          THE SUMMER RAINS, WHICH ARE CONVECTIONAL.  ASSUME THAT    */
/*          THE FRACTION OF RAINFALL WHICH INFILTRATES IN THE SUMMER  */
/*          IS 0.4; IF IT'S NOT SUMMER, ASSUME THAT 0.8 INFILTRATES.  */
/*                                                                    */
/*      2.  POTENTIAL EVAPOTRANSPIRATION (PET) IS CALCULATED WITH     */
/*          THE BLAINEY-CRIDDLE FORMULE ('EMPIRICAL METHODS OF        */
/*          PREDICTING EVAPOTRANSPIRATION USING AIR TEMPERATURE AS    */
/*          THE PRIMARY VARIABLE', IN ASAE CONFERENCE PROCEEDINGS:    */
/*          EVAPOTRANSPIRATION AND ITS ROLE IN WATER RESOURCE         */
/*          MANAGEMENT, CHICAGO, ILL., DEC. 5 & 6, 1966).  THE        */
/*          COVER COEFFICIENT USED IN THIS FORMULA IS BROKEN INTO      */
/*          TWO PARTS:  1) PERENNIAL_COVER_COEFFICIENT = 0.25, AND    */
/*          2) ANNUAL_COVER_COEFFICIENT WHICH RANGES BETWEEN ZERO     */
/*          AND 0.05 AND IS A FUNCTION OF THE BIOMASS OF THE ANNUALS. */
/*          (NOTE:  THE BLAINEY-CRIDDLE FORMULA OVERESTIMATES PET     */
/*          IN SUMMER AND UNDERESTIMATES IT IN WINTER.)               */
/*                                                                    */
/*      3.  ACTUAL EVAPOTRANSPIRATION (AET) IS OBTAINED BY REDUCING   */
/*          PET BY THE DIMENSIONLESS RATIO: (SOIL MOISTURE/FIELD      */
/*          CAPACITY).  THIS IS A COMMON EMPIRICAL PROCEDURE AND      */
/*          CAUSES THE SOIL TO DRY OUT AT A DECREASING RATE FROM      */
/*          THE TIME OF WETTING.                                      */
/*                                                                    */
/*      4.  SOIL MOISTURE IS REPRESENTED BY THE VARIABLE 'SOIL_WATER' */
/*          AND IS THE % MOISTURE BY VOLUME IN THE TOP LAYER OF       */
/*          SOIL OF THICKNESS 'SOIL_DEPTH'.  FIELD CAPACITY IS        */
/*          ASSUMED TO BE 18% AND WILT POINTS FOR THE ANNUALS ARE     */
/*          WINTER_WILT_POINT AND SUMMER_WILT_POINT, RESPECTIVELY.    */
/*                                                                    */
/*      5.  GERMINATION BIOMASS OF THE ANNUALS IS ASSUMED TO BE A     */
/*          FUNCTION OF THREE FACTORS:                                */
/*               A. SEASON OF THE YEAR                                */
/*               B. RAINFALL IN A GIVEN WEEK                          */
/*               C. SEED RESERVES IN THE SOIL                         */
/*                                                                    */
/*          A MAXIMUM FRACTION OF THE SEEDS ARE ALLOWED TO GERMINATE  */
/*          IN ANY PARTICULAR WEEK, AS DETERMINED BY THE PARAMETERS   */
/*          'MAX_FRACTION_SUMMER_GERMINATION' AND                     */
/*          'MAX_FRACTION_WINTER_GERMINATION'.                        */
/*                                                                    */
/**********************************************************************/
/*                                                                    */
```

```
/*                                                                    */
/**********************************************************************/
/*                                                                    */
/*      6. WEEKLY BIOMASS INCREMENT IS ASSUMED TO BE A PRODUCT OF:    */
/*           A. BIOMASS ALREADY PRESENT                               */
/*           B. A GAUSSIAN FUNCTION OF MEAN AIR TEMPERATURE           */
/*           C. A SOIL MOISTURE FACTOR                                */
/*           D. A 'MAXIMUM GROWTH RATE', REPRESENTING THE LARGEST     */
/*              POSSIBLE FRACTIONAL INCREASE IN A WEEK WHEN ALL       */
/*              CONDITIONS ARE OPTIMAL.                               */
/*                                                                    */
/*      7.  THE SIZE OF THE SEED CROP IS A FUNCTION OF TOTAL          */
/*          BIOMASS AT THE TIME OF SEED SET.  SEED RESERVES IN THE    */
/*          SOIL ARE DECREMENTED BY THE AMOUNT LOST IN GERMINATION.   */
/*          OTHER LOSSES (RODENTS, ANTS, FUNGUS, ETC.) ARE NOT        */
/*          INCLUDED.  A RODENT MODEL WILL BE BUILT TO INTERFACE      */
/*          WITH THIS MODEL.                                          */
/*                                                                    */
/*      8.  THE WEEKLY MEAN TEMPERATURE HAS BEEN GIVEN A SINUSOIDAL   */
/*          SHAPE WHICH REFLECTS THE LONG-TERM MEAN VALUES.  THE      */
/*          PRESENT STATE OF THE PROGRAM DOES NOT WARRANT USING       */
/*          REAL TEMPERATURE DATA WHICH FLUXUATES FROM YEAR TO YEAR.  */
/*                                                                    */
/*      9.  THE UNITS USED ARE:                                       */
/*              RAINFALL IS IN MILLIMETERS/WEEK.                      */
/*              EVAPORATION VARIABLES ARE IN MILLIMETERS/WEEK.        */
/*              SOIL MOISTURE VARIABLES ARE IN UNITS OF % MOISTURE    */
/*                BY VOLUME.                                          */
/*              TEMPERATURES ARE IN DEGREES CELSIUS.                  */
/*              ALL BIOMASS VARIABLES (E.G., 'GREEN_VEG',             */
/*                'WEEKLY_PRODUCTION','ANNUAL_SEED') ARE IN           */
/*                UNITS OF GRAMS/SQ. METER.                           */
/*                                                                    */
/**********************************************************************/
/*                                                                    */
```

```
/*                                                                    */
/**********************************************************************/
/*                                                                    */
/*      TAKING JAN. 1 AS THE FIRST DAY OF WEEK 1, THE FOLLOWING TABLE*/
/*           INDICATES THE DATE CORRESPONDING TO THE FIRST DAY OF     */
/*           EACH SUCCEEDING WEEK:                                    */
/*                                                                    */
/*  WEEK      DATE          WEEK      DATE          WEEK      DATE   */
/*    1       JAN  1          19      MAY  7          37     SEPT 10  */
/*    2            8          20           14          38          17  */
/*    3           15          21           21          39          24  */
/*    4           22          22           28          40     OCT  1  */
/*    5           29          23      JUNE  4          41           7  */
/*    6       FEB  5          24           11          42          15  */
/*    7           12          25           18          43          22  */
/*    8           19          26           25          44          29  */
/*    9           26          27      JULY  2          45     NOV  5  */
/*   10      MARCH  5          28            9          46          12  */
/*   11           12          29           16          47          19  */
/*   12           19          30           23          48          26  */
/*   13           26          31           30          49     DEC  3  */
/*   14      APRIL  2          32      AUG  6          50          10  */
/*   15           9          33           13          51          17  */
/*   16          16          34           20          52          24  */
/*   17          23          35           27                          */
/*   18          30          36      SEPT  3                          */
/*                                                                    */
/**********************************************************************/
/*                                                                    */
/*      AVERAGE MONTHLY VALUES FOR TUCSON, ARIZONA                    */
/*      (SOURCE:  JAEGER, 'NORTH AMERICAN DESERTS', CHAPTER 2)        */
/*                                                                    */
/*         J    F    M    A    M    J    J    A    S    O    N    D  */
/*                                                                    */
/* RAIN                                                               */
/* (IN.)  .72 1.01  .61  .31  .21  .16 1.98 2.17 1.46  .63  .45  .82 */
/* (MM.) 18.3 25.7 15.5  7.9  5.3  4.1 50.3 55.1 37.1 16.0 11.4 20.8 */
/*                                                                    */
/* DAYS                                                               */
/*  OF                                                                */
/* RAIN   4    4    4    2    1    1    9    9    5    4    2    4   */
/*                                                                    */
/* MEAN                                                               */
/* TEMP                                                               */
/* (F)    50   54   58   66   74   82   87   84   81   70   59   52  */
/* (C)    10   12   14   19   23   28   30   29   27   21   15   11  */
/*                                                                    */
/**********************************************************************/
/*                                                                    */
```

```
                /*                                                              */
                /****************************************************************/
                /*                                                              */
                /*      DECLARATIONS AND SOME INITIALIZATIONS (THESE VALUES ARE  */
                /*           APPROPRIATE FOR TUCSON, ARIZONA)                   */
                /*                                                              */
                /****************************************************************/
                /*                                                              */
 2              DCL #_YEARS_TO_RUN FIXED BIN;
 3              DCL AUTUMN BIT (1) INIT('0'B);
 4              DCL AUTUMN_MOISTURE_FACTOR_AXIS (4) INIT(0,0,1.0,1.0);
 5              DCL BOUNDS (12) FLOAT DEC INIT(40,0,25,0,.50,0,10,0,50,0,5,0);
 6              DCL COEFFICIENT_AXIS (2) INIT(0,0.05);
 7              DCL CURRENT_WEEK FIXED BIN INIT(1);
 8              DCL CURRENT_WEEK_OF_RUN FIXED BIN;
 9              DCL DEBUG_RUN BIT(1);
10              DCL FALSE BIT (1) INIT('0'B);
11              DCL FUNCTIONS_DESIRED BIT(1);
12              DCL INFILTRATION FLOAT DEC INIT (0);  /* MILLIMETERS OF WATER      */
13              DCL MAX_FRACTION_SUMMER_GERMINATION FLOAT;
14              DCL MAX_FRACTION_WINTER_GERMINATION FLOAT;
15              DCL IT_HAS_RAINED BIT (1) INIT('0'B);
16              DCL MAX_SUMMER_GROWTH_RATE FLOAT DEC;
17              DCL MAX_WINTER_GROWTH_RATE FLOAT DEC;
18              DCL OUTPUT_DATA (60) CHAR(130) VAR;
19              DCL OUTPUT_LINE FIXED BIN INIT(4);
20              DCL RAIN_AXIS (4) INIT(0,10,20,50);
21              DCL RAIN_THIS_WEEK INIT(0);
22              DCL RUNOFF INIT (0);                  /* MILLIMETERS OF WATER      */
23              DCL SEASON CHAR(6);
24              DCL SOIL_WATER_AXIS (5) INIT(0,4,8,12,16);
25              DCL SPRING BIT (1) INIT('0'B);
26              DCL SPRING_MOISTURE_FACTOR_AXIS (4) INIT(0,0,1.0,1.0);
27              DCL SUMMER BIT (1) INIT('0'B);
28              DCL SUMMER_GERMINATION_BIOMASS INIT(0);
29              DCL SUMMER_GREEN_VEG_AXIS (3) INIT(0,20,100);
30              DCL SUMMER_INFILTRATING_FRACTION INIT(.4);  /* FRACTION INFILTRATING */
31              DCL SUMMER_MOISTURE_FACTOR_AXIS (4) INIT(0,0,1.0,1.0);
32              DCL SUMMER_SEED_CROP INIT(0);
33              DCL SUMMER_SEED_CROP_AXIS (3) INIT(0,2,3);
34              DCL SUMMER_SOIL_WATER_AXIS (5) INIT(0,0.5,1.0,1.0,1.0);
35              DCL SUMMER_WEEKLY_PRODUCTION INIT(0);
36              DCL THIS_WEEK FIXED INIT(0);
37              DCL TRUE BIT (1) INIT('1'B);
38              DCL VEG_AXIS (2) INIT(0,50);
39              DCL WEEKLY_FRACTION_ANNUAL_DAY_HRS(52);
40              DCL WEEKLY_MEAN_TEMP(52);
```

```
41              DCL WEEK_OF_LAST_RAIN FIXED BIN INIT(1);
42              DCL WEEK_OF_NEXT_RAIN FIXED BIN INIT(0);
43              DCL WINTER BIT (1) INIT('1'B);
44              DCL WINTER_GERMINATION_BIOMASS INIT(0);
45              DCL WINTER_GREEN_VEG_AXIS (3) INIT(0,20,100);
46              DCL WINTER_INFILTRATING_FRACTION INIT(.8);  /* FRACTION INFILTRATING */
47              DCL WINTER_SEED_CROP INIT(0);
48              DCL WINTER_SEED_CROP_AXIS (3) INIT(0,2,3);
49              DCL WINTER_SOIL_WATER_AXIS (5) INIT(0,0.33,0.67,1.0,1.0);
50              DCL WINTER_WEEKLY_PRODUCTION INIT(0);
51              DCL YEAR_OF_NEXT_RAIN FIXED BIN INIT(1);
52              DCL YEAR_OF_RUN FIXED BIN INIT(1);
53              OPEN FILE(SYSPRINT) LINESIZE(130);
                /*                                                              */
                /****************************************************************/
                /*                                                              */
                /*      INITIALIZATION OF PARAMETERS (THESE VALUES ARE APPROPRIATE  */
                /*           FOR TUCSON, ARIZONA)                               */
                /*                                                              */
                /****************************************************************/
                /*                                                              */
54              FIELD_CAPACITY=18;                  /* % MOISTURE BY VOLUME     */
55              MAX_FRACTION_SUMMER_GERMINATION=.10; /* MAXIMUM PER WEEK        */
56              MAX_FRACTION_WINTER_GERMINATION=.10; /* MAXIMUM PER WEEK        */
57              MAX_SUMMER_GROWTH_RATE=.80;         /* FRACTIONAL INCREASE/WEEK */
58              MAX_WINTER_GROWTH_RATE=.40;         /* FRACTIONAL INCREASE/WEEK */
59              OPTIMUM_SUMMER_ANNUAL_TEMP= 28;     /* OPTIMUM GROWTH TEMPERATURE */
60              OPTIMUM_WINTER_ANNUAL_TEMP= 17;     /* OPTIMUM GROWTH TEMPERATURE */
61              PERENNIAL_COVER_COEFFICIENT=0.25;   /* DIMENSIONLESS            */
62              RAIN= 0;                            /* MILLIMETERS              */
63              SOIL_DEPTH=300;                     /* MILLIMETERS              */
64              SOIL_WATER=15;                      /* % MOISTURE BY VOLUME     */
65              SUMMER_ANNUAL_SEED=1.0;             /* GRAMS/SQ. METER          */
66              SUMMER_GREEN_VEG= 0;                /* GRAMS/SQ. METER          */
67              SUMMER_GROWTH_TEMP_WIDTH=5.0;       /* DEGREES CELSIUS          */
68              WINTER_ANNUAL_SEED=1.0;             /* GRAMS/SQ. METER          */
69              WINTER_GREEN_VEG= 3;                /* GRAMS/SQ. METER          */
70              WINTER_GROWTH_TEMP_WIDTH=3.0;       /* DEGREES CELSIUS          */
```

```
/*                                                                    */
/**********************************************************************/
/*                                                                    */
/*                    ***** OUTPUT OPTIONS *****                      */
/*                                                                    */
/*    1. #_YEARS_TO_RUN IS THE NUMBER OF YEARS THAT THE PROGRAM       */
/*          WILL SIMULATE                                             */
/*                                                                    */
/*       DEFAULT VALUE IS #_YEARS_TO_RUN=1                            */
/*                                                                    */
/*    2. FUNCTIONS_DESIRED='TRUE' GIVES GRAPHS OF ALL THE FUNCTIONS   */
/*          THAT ARE USED IN THE PROGRAM                              */
/*                                                                    */
/*       DEFAULT VALUE IS FUNCTIONS_DESIRED = 'FALSE'                 */
/*                                                                    */
/*    3. REPORT=1 GIVES A GRAPH OF WEEKLY_MEAN_TEMP, SOIL_WATER,      */
/*          GERMINATION_BIOMASS, WEEKLY PRODUCTION, GREEN VEGETATION*/
/*          AND WINTER_ANNUAL_SEED  AS A FUNCTION OF CURRENT_WEEK   */
/*                                                                    */
/*       REPORT=2 GIVES A TABLE OF SEASON, WEEKLY_MEAN_TEMP,          */
/*          RAIN_THIS_WEEK, POTENTIAL_EVAPOTRANSPIRATION,             */
/*          ACTUAL_EVAPOTRANSPIRATION, SOIL_WATER, GERMINATION        */
/*          BIOMASS, WEEKLY PRODUCTION, GREEN VEGETATION,             */
/*          WINTER_ANNUAL_SEED AND SUMMER_ANNUAL_SEED AS A FUNCTION */
/*          OF CURRENT_WEEK.                                          */
/*          THIS OPTION SHOULD BE USED FOR DEBUG RUNS.                */
/*                                                                    */
/*       DEFAULT VALUE IS REPORT=1.  THE GRAPH TAKES ABOUT TWICE      */
/*          AS MUCH TIME TO EXECUTE AS DOES THE TABLE, BUT IS         */
/*          EASIER TO READ                                            */
/*                                                                    */
/*    4. DEBUG_RUN = 'TRUE' WILL PRINT VALUES OF ALL THE IMPORTANT    */
/*          VARIABLES EACH WEEK OF THE RUN FOR DEBUGING PURPOSES      */
/*                                                                    */
/*       DEFAULT VALUE IS DEBUG_RUN = 'FALSE'                         */
/*                                                                    */
/**********************************************************************/
/*                                                                    */
71          #_YEARS_TO_RUN=1;                  /* DEFAULT IS A 1-YEAR RUN     */
72          DEBUG_RUN = FALSE;                 /* DEFAULT VALUE IS FALSE      */
73          FUNCTIONS_DESIRED = FALSE;         /* DEFAULT VALUE IS FALSE      */
74          REPORT=1;                          /* DEFAULT IS GRAPH OUTPUT     */
```

```
/*                                                                    */
/**********************************************************************/
/*                                                                    */
/*      THE FOLLOWING 'GET LIST' IS FOR READING IN VALUES FOR         */
/*           WEEKLY_FRACTION_ANNUAL_DAY_HRS AND WEEKLY_MEAN_TEMP      */
/*                                                                    */
/*      HERE ARE SAMPLE DATA CARDS (APPROPRIATE FOR TUCSON)           */
/*                                                                    */
/*      WEEKLY_FRACTION_ANNUAL_DAY_HRS:                               */
/*                                                                    */
/*      .0164 .0166 .0167 .0169 .0171 .0174 .0177 .0180               */
/*      .0183 .0187 .0191 .0194 .0198 .0202 .0205 .0209               */
/*      .0212 .0215 .0217 .0220 .0222 .0223 .0224 .0225               */
/*      .0225 .0225 .0224 .0223 .0222 .0220 .0217 .0215               */
/*      .0212 .0209 .0205 .0202 .0198 .0194 .0191 .0187               */
/*      .0183 .0180 .0177 .0174 .0171 .0169 .0167 .0166               */
/*      .0164 .0164 .0164 .0164                                       */
/*                                                                    */
/*      WEEKLY_MEAN_TEMP:                                             */
/*                                                                    */
/*      10.2 10.0 10.0 10.0 10.2 10.6 11.1 11.7 12.5 13.3             */
/*      14.3 15.3 16.4 17.6 18.7 20.0 21.2 22.3 23.5 24.6             */
/*      25.6 26.6 27.4 28.2 28.8 29.3 29.7 29.9 30.0 29.9             */
/*      29.7 29.3 28.8 28.2 27.4 26.6 25.6 24.6 23.5 22.3             */
/*      21.2 19.9 18.7 17.6 16.4 15.3 14.3 13.3 12.5 11.7             */
/*      11.1 10.6                                                     */
/*                                                                    */
/**********************************************************************/
/*                                                                    */
75      GET LIST (WEEKLY_FRACTION_ANNUAL_DAY_HRS, WEEKLY_MEAN_TEMP);
/*                                                                    */
/**********************************************************************/
/*                                                                    */
/*      THE FOLLOWING 'GET DATA' IS FOR READING IN VALUES TO          */
/*           OVERRIDE ANY OF THE DEFAULT INITIALIZATIONS.  A SAMPLE   */
/*           DATA CARD WOULD BE:                                      */
/*                                                                    */
/*    #_YEARS_TO_RUN=5,WINTER_GREEN_VEG=30,FIELD_CAPACITY=15;         */
/*                                                                    */
/**********************************************************************/
/*                                                                    */
76      GET DATA;
```

```
            /*                                                              */
            /******************************************************************/
            /*                                                              */
            /*      HEADINGS FOR OUTPUT TABLE                               */
            /*                                                              */
            /******************************************************************/
            /*                                                              */
77          IF REPORT=2 THEN DO;
79              PUT STRING (OUTPUT_DATA(1)) EDIT('YEAR','WEEK','SEASON','TEMP',
                    'RAIN','PET','AET','SOIL','GERMINATION','WEEKLY','GREEN',
                    'WINTER','SUMMER')
                    (X(3),A,X(3),A,X(3),A,X(3),A,X(4),A,X(3),A,X(4),A,X(5),A,X(4),
                    A,X(5),A,X(7),A,X(6),A,X(3),A);
80              PUT STRING (OUTPUT_DATA(2)) EDIT('(C)','(MM)','(MM)','(MM)',
                    'WATER','BIOMASS','PRODUCTION','VEGETATION','SEED','SEED')
                    (X(26),A,X(5),A,X(3),A,X(3),A,X(4),A,X(5),A,X(5),A,X(3),A,
                    X(4),A,X(5),A);
81              PUT STRING (OUTPUT_DATA(3)) EDIT('(%)','(G/SQ.M)','(G/SQ.M)',
                    '(G/SQ.M)','(G/SQ.M)','(G/SQ.M)')
                    (X(57),A,X(6),A,X(5),A,X(5),A,X(3),A,X(1),A);
82              END;
            /*                                                              */
            /******************************************************************/
            /*                                                              */
            /*      PRINT GRAPHS OF ALL THE FUNCTIONS USED IN THE PROGRAM   */
            /*                                                              */
            /******************************************************************/
            /*                                                              */
83          IF FUNCTIONS_DESIRED THEN DO;
85              CALL PUT_CURVE('FIG. 1, ANNUAL VEG(X) VS COVER COEF(Y)',
                     VEG_AXIS,COEFFICIENT_AXIS);
86              CALL PUT_CURVE('FIG. 2, RAINFALL(X) VS SUMMER GERMINATION'||
                    ' FACTOR(Y)',RAIN_AXIS,SUMMER_MOISTURE_FACTOR_AXIS);
87              PUT PAGE;
88              CALL PUT_CURVE('FIG. 3, SOIL WATER(X) VS SUMMER GROWTH FACTOR(Y)',
                    SOIL_WATER_AXIS,SUMMER_SOIL_WATER_AXIS);
89              CALL PUT_CURVE('FIG. 4, SUMMER GREEN VEG(X) VS SEED CROP(Y)',
                    SUMMER_GREEN_VEG_AXIS,SUMMER_SEED_CROP_AXIS);
90              PUT PAGE;
91              CALL PUT_CURVE('FIG. 5, RAINFALL(X) VS SPRING GERMINATION'||
                    ' FACTOR(Y)',RAIN_AXIS,SPRING_MOISTURE_FACTOR_AXIS);
92              CALL PUT_CURVE('FIG. 6, RAINFALL(X) VS AUTUMN GERMINATION'||
                    ' FACTOR(Y)',RAIN_AXIS,AUTUMN_MOISTURE_FACTOR_AXIS);
93              PUT PAGE;
94              CALL PUT_CURVE('FIG. 7, SOIL WATER(X) VS WINTER GROWTH FACTOR(Y)',
                    SOIL_WATER_AXIS,WINTER_SOIL_WATER_AXIS);
95              CALL PUT_CURVE('FIG. 8, WINTER GREEN VEG(X) VS SEED CROP(Y)',
                    WINTER_GREEN_VEG_AXIS,WINTER_SEED_CROP_AXIS);
96              PUT PAGE;
97              END;
```

```
                /*                                                              */
                /****************************************************************/
                /*                                                              */
                /*      STATEMENT LABEL 'NEW_DATE:' IS THE START OF THE EVENT LOOP.  */
                /*                                                              */
                /*      THE FOLLOWING 'GET DATA' IS FOR READING IN THE WEEK AND     */
                /*           THE AMOUNT OF THE NEXT RAINFALL EVENT. TYPICAL ENTRIES  */
                /*           ON A DATA CARD WOULD BE:                           */
                /*                                                              */
                /*                NEXT_WEEK= 41, RAIN= 35.0;                    */
                /*                                                              */
                /*           (THIS INDICATES THAT IT RAINED 35 MILLIMETERS DURING    */
                /*           THE SECOND WEEK OF OCTOBER)                        */
                /*                                                              */
                /****************************************************************/
                /*                                                              */
 98             NEW_DATE:
                    GET DATA;
 99             IF DEBUG_RUN THEN
100                 PUT SKIP(2) DATA (NEXT_WEEK,RAIN);
                /*                                                              */
                /****************************************************************/
                /*                                                              */
                /*      THE FOLLOWING SECTION IS FOR KEEPING TRACK OF THE CORRECT    */
                /*           YEAR AND WEEK DURING THE SIMULATION.  IGNORE THIS      */
                /*           PART AND GO TO THE NEXT PAGE.                      */
                /*                                                              */
                /****************************************************************/
                /*                                                              */
101                 IF NEXT_WEEK < THIS_WEEK THEN
102                     YEAR_OF_NEXT_RAIN=YEAR_OF_NEXT_RAIN+1;
103                 THIS_WEEK=NEXT_WEEK;
104                 WEEK_OF_NEXT_RAIN=NEXT_WEEK+52*(YEAR_OF_NEXT_RAIN-1);
```

```
                /*                                                              */
                /****************************************************************/
                /*                                                              */
                /*      START OF THE WEEKLY TIMING LOOP                         */
                /*                                                              */
                /*      THE TIME INTERVAL IS ONE WEEK.  TIME ADVANCES FROM THE WEEK  */
                /*           OF THE LAST RAINFALL EVENT TO THE WEEK OF THE NEXT  */
                /*           RAINFALL EVENT.  THE ONLY TIME-VARIABLE THAT IS    */
                /*           SIGNIFICANT WITHIN THIS WEEK LOOP IS 'CURRENT_WEEK',    */
                /*           WHICH ADVANCES FROM 1 TO 52 DURING A GIVEN YEAR OF  */
                /*           THE SIMULATION.                                    */
                /*                                                              */
                /****************************************************************/
                /*                                                              */
105             DO CURRENT_WEEK_OF_RUN = WEEK_OF_LAST_RAIN TO WEEK_OF_NEXT_RAIN;
106                 YEAR_OF_RUN=((CURRENT_WEEK_OF_RUN-1)/52)+1;
107                 CURRENT_WEEK=CURRENT_WEEK_OF_RUN-52*(YEAR_OF_RUN-1);
108                 IF YEAR_OF_RUN > #_YEARS_TO_RUN THEN GO TO END_PROGRAM;
110                 IF DEBUG_RUN THEN
111                     PUT SKIP(2) DATA (CURRENT_WEEK);
                /*                                                              */
                /****************************************************************/
                /*                                                              */
                /*           AT THE BEGINNING OF EACH YEAR, READ IN (VIA THE    */
                /*           'GET LIST') THE WEEKS OF THE SEASON CHANGES.       */
                /*           (SAMPLE DATA:  10 22 38 50)                        */
                /*                                                              */
                /****************************************************************/
                /*                                                              */
112                 IF CURRENT_WEEK=1 THEN DO;
114                     GET LIST (WEEK_SPRING_BEGINS,WEEK_SUMMER_BEGINS,
                                  WEEK_AUTUMN_BEGINS,WEEK_WINTER_BEGINS);
115                     IF DEBUG_RUN THEN
116                         PUT DATA(WEEK_SPRING_BEGINS,WEEK_SUMMER_BEGINS,
                                     WEEK_AUTUMN_BEGINS,WEEK_WINTER_BEGINS);
117                     END;
                /*                                                              */
                /****************************************************************/
                /*                                                              */
                /*      SEE IF IT RAINED THIS WEEK AND SET 'RAIN_THIS_WEEK'     */
                /*           ACCORDINGLY. (IT_HAS_RAINED IS A VARIABLE WHICH IS */
                /*           EITHER 'TRUE' OR 'FALSE')                          */
                /*                                                              */
                /****************************************************************/
                /*                                                              */
118             IT_HAS_RAINED =(CURRENT_WEEK_OF_RUN = WEEK_OF_NEXT_RAIN);
119             IF IT_HAS_RAINED THEN RAIN_THIS_WEEK=RAIN;
121                 ELSE RAIN_THIS_WEEK=0;
```

```
                /*                                                              */
                /******************************************************************/
                /*                                                              */
                /*      DETERMINE WHAT THE SEASON IS. (WINTER, SPRING, SUMMER,  */
                /*           AND AUTUMN ARE 'TRUE' OR 'FALSE' VARIABLES)        */
                /*                                                              */
                /******************************************************************/
                /*                                                              */
122             WINTER = (CURRENT_WEEK>=WEEK_WINTER_BEGINS
                         | CURRENT_WEEK<WEEK_SPRING_BEGINS);
123             SPRING = (CURRENT_WEEK>=WEEK_SPRING_BEGINS
                         & CURRENT_WEEK<WEEK_SUMMER_BEGINS);
124             SUMMER = (CURRENT_WEEK>=WEEK_SUMMER_BEGINS
                         & CURRENT_WEEK<WEEK_AUTUMN_BEGINS);
125             AUTUMN = (CURRENT_WEEK>=WEEK_AUTUMN_BEGINS
                         & CURRENT_WEEK<WEEK_WINTER_BEGINS);
                /*                                                              */
                /******************************************************************/
                /*                                                              */
                /*      IF IT HAS RAINED THIS WEEK, CALCULATE RUNOFF, INFILTRATION,  */
                /*           AND UPDATE SOIL WATER, KEEPING IT LESS THAN OR EQUAL    */
                /*           TO FIELD CAPACITY (RUNOFF, INFILTRATION, AND            */
                /*           DEEP-PERCOLATION ARE IN MM OF WATER; SOIL_WATER IS IN   */
                /*           % MOISTURE BY VOLUME IN THE TOP LAYER OF SOIL OF        */
                /*           THICKNESS 'SOIL_DEPTH')                                 */
                /*                                                              */
                /******************************************************************/
                /*                                                              */
126             IF IT_HAS_RAINED THEN DO;
128                IF SUMMER THEN INFILTRATION = RAIN_THIS_WEEK
                                               *SUMMER_INFILTRATING_FRACTION;
130                   ELSE INFILTRATION = RAIN_THIS_WEEK
                                          *WINTER_INFILTRATING_FRACTION;
131                RUNOFF=RAIN_THIS_WEEK-INFILTRATION;
132                SOIL_WATER=SOIL_WATER+(INFILTRATION/SOIL_DEPTH)*100;
133                IF DEBUG_RUN THEN
134                     PUT SKIP(2) DATA(RAIN_THIS_WEEK,INFILTRATION,RUNOFF,
                                SOIL_WATER);
135                IF SOIL_WATER > FIELD_CAPACITY THEN DO;
137                     DEEP_PERCOLATION=SOIL_DEPTH*(SOIL_WATER-FIELD_CAPACITY)/100;
138                     SOIL_WATER=FIELD_CAPACITY;
139                     IF DEBUG_RUN THEN
140                          PUT SKIP(2) DATA(DEEP_PERCOLATION,SOIL_WATER);
141                     END;
142                END;
```

```
                /*                                                              */
                /*                                                              */
                /******************************************************************/
                /*                                                              */
                /*      CALCULATE POTENTIAL EVAPORATRANSPIRATION, ACTUAL        */
                /*           EVAPORTANSPIRATION, AND UPDATE SOIL WATER.  PET AND */
                /*           AET ARE IN MM OF WATER/WEEK.                       */
                /*                                                              */
                /******************************************************************/
                /*                                                              */
143             IF SUMMER THEN VEG=SUMMER_GREEN_VEG;
145                ELSE VEG=WINTER_GREEN_VEG;
                                    /* SEE FIGURE 1 FOR THE FOLLOWING FUNCTION  */
146             ANNUAL_COVER_COEFFICIENT=CURVE(VEG,VEG_AXIS,COEFFICIENT_AXIS);
147             COVER_COEFFICIENT=PERENNIAL_COVER_COEFFICIENT
                                 +ANNUAL_COVER_COEFFICIENT;
148             POTENTIAL_EVAPOTRANSPIRATION = 25.4*COVER_COEFFICIENT
                                *(1.8*WEEKLY_MEAN_TEMP(CURRENT_WEEK)+32)
                                *WEEKLY_FRACTION_ANNUAL_DAY_HRS(CURRENT_WEEK);
149             ACTUAL_EVAPOTRANSPIRATION=POTENTIAL_EVAPOTRANSPIRATION
                                *(SOIL_WATER/FIELD_CAPACITY);
150             SOIL_WATER = SOIL_WATER
                           - (ACTUAL_EVAPOTRANSPIRATION/SOIL_DEPTH)*100;
151             IF DEBUG_RUN THEN
152                 PUT SKIP(2) DATA (POTENTIAL_EVAPOTRANSPIRATION,
                      ACTUAL_EVAPOTRANSPIRATION,SOIL_WATER);
                /*                                                              */
                /******************************************************************/
                /*                                                              */
                /*     GERMINATION OF SUMMER ANNUALS                            */
                /*                                                              */
                /******************************************************************/
153             IF SUMMER & IT_HAS_RAINED THEN DO;
                                    /* SEE FIGURE 2 FOR THE FOLLOWING FUNCTION  */
155                  SUMMER_MOISTURE_FACTOR=CURVE(RAIN_THIS_WEEK,
                                    RAIN_AXIS,SUMMER_MOISTURE_FACTOR_AXIS);
156                  SUMMER_GERMINATION_BIOMASS=MAX_FRACTION_SUMMER_GERMINATION
                                *SUMMER_MOISTURE_FACTOR*SUMMER_ANNUAL_SEED;
157                  SUMMER_ANNUAL_SEED=SUMMER_ANNUAL_SEED
                                -SUMMER_GERMINATION_BIOMASS;
158                  SUMMER_GREEN_VEG=SUMMER_GREEN_VEG+SUMMER_GERMINATION_BIOMASS;
159                  IF DEBUG_RUN THEN
160                     PUT SKIP(2) DATA (SUMMER_ANNUAL_SEED,SUMMER_MOISTURE_FACTOR,
                                SUMMER_GERMINATION_BIOMASS,SUMMER_GREEN_VEG);
161                  END;
```

```
                /*                                                              */
                /****************************************************************/
                /*                                                              */
                /*      GROWTH OF SUMMER ANNUALS                                */
                /*                                                              */
                /****************************************************************/
                /*                                                              */
162             IF SUMMER_GREEN_VEG ¬= 0 THEN DO;
164                 SUMMER_TEMP_FACTOR=EXP(-.5*((OPTIMUM_SUMMER_ANNUAL_TEMP
                            -WEEKLY_MEAN_TEMP(CURRENT_WEEK))/
                             SUMMER_GROWTH_TEMP_WIDTH)**2);
                                    /* SEE FIGURE 3 FOR THE FOLLOWING FUNCTION  */
165                 SUMMER_SOIL_WATER_FACTOR=CURVE(SOIL_WATER,SOIL_WATER_AXIS,
                        SUMMER_SOIL_WATER_AXIS);
166                 SUMMER_GROWTH_RATE=MAX_SUMMER_GROWTH_RATE*SUMMER_TEMP_FACTOR
                                      *SUMMER_SOIL_WATER_FACTOR;
167                 SUMMER_WEEKLY_PRODUCTION=SUMMER_GROWTH_RATE*SUMMER_GREEN_VEG;
168                 SUMMER_GREEN_VEG=SUMMER_GREEN_VEG+SUMMER_WEEKLY_PRODUCTION;
169                 IF DEBUG_RUN THEN
170                    PUT SKIP(2) DATA (SUMMER_TEMP_FACTOR,SUMMER_SOIL_WATER_FACTOR,
                           SUMMER_GROWTH_RATE,SUMMER_WEEKLY_PRODUCTION,
                           SUMMER_GREEN_VEG,WEEKLY_MEAN_TEMP(CURRENT_WEEK));
171                 END;
                /*                                                              */
                /****************************************************************/
                /*                                                              */
                /*      SEED SET FOR SUMMER ANNUALS                             */
                /*                                                              */
                /****************************************************************/
                /*                                                              */
172             IF CURRENT_WEEK = WEEK_AUTUMN_BEGINS THEN DO;
                                    /* SEE FIGURE 4 FOR THE FOLLOWING FUNCTION  */
174                 SUMMER_SEED_CROP = CURVE(SUMMER_GREEN_VEG,
                        SUMMER_GREEN_VEG_AXIS,SUMMER_SEED_CROP_AXIS);
175                 SUMMER_ANNUAL_SEED=SUMMER_ANNUAL_SEED+SUMMER_SEED_CROP;
176                 SUMMER_GREEN_VEG=0;
177                 SUMMER_WEEKLY_PRODUCTION=0;
178                 IF DEBUG_RUN THEN
179                    PUT SKIP(2) DATA (SUMMER_SEED_CROP,SUMMER_ANNUAL_SEED,
                          SUMMER_GREEN_VEG);
180                 END;
```

```
                /*                                                              */
                /****************************************************************/
                /*                                                              */
                /*      SPRING GERMINATION OF WINTER ANNUALS                    */
                /*                                                              */
                /****************************************************************/
181             IF SPRING & IT_HAS_RAINED THEN DO;
                                /* SEE FIGURE 5 FOR THE FOLLOWING FUNCTION  */
183                     SPRING_MOISTURE_FACTOR=CURVE(RAIN_THIS_WEEK,
                                RAIN_AXIS,SPRING_MOISTURE_FACTOR_AXIS);
184                     WINTER_GERMINATION_BIOMASS=MAX_FRACTION_WINTER_GERMINATION
                                *SPRING_MOISTURE_FACTOR*WINTER_ANNUAL_SEED;
185                     WINTER_ANNUAL_SEED=WINTER_ANNUAL_SEED
                                -WINTER_GERMINATION_BIOMASS;
186                     WINTER_GREEN_VEG=WINTER_GREEN_VEG+WINTER_GERMINATION_BIOMASS;
187                     IF DEBUG_RUN THEN
188                        PUT SKIP(2) DATA (WINTER_ANNUAL_SEED,SPRING_MOISTURE_FACTOR,
                                WINTER_GERMINATION_BIOMASS,WINTER_GREEN_VEG);
189                     END;
                /*                                                              */
                /****************************************************************/
                /*                                                              */
                /*      AUTUMN GERMINATION OF WINTER ANNUALS                    */
                /*                                                              */
                /****************************************************************/
                /*                                                              */
190             IF AUTUMN & IT_HAS_RAINED THEN DO;
                                /* SEE FIGURE 6 FOR THE FOLLOWING FUNCTION  */
192                     AUTUMN_MOISTURE_FACTOR=CURVE(RAIN_THIS_WEEK,
                                RAIN_AXIS,AUTUMN_MOISTURE_FACTOR_AXIS);
193                     WINTER_GERMINATION_BIOMASS=MAX_FRACTION_WINTER_GERMINATION
                                *AUTUMN_MOISTURE_FACTOR*WINTER_ANNUAL_SEED;
194                     WINTER_ANNUAL_SEED=WINTER_ANNUAL_SEED
                                -WINTER_GERMINATION_BIOMASS;
195                     WINTER_GREEN_VEG=WINTER_GREEN_VEG+WINTER_GERMINATION_BIOMASS;
196                     IF DEBUG_RUN THEN
197                        PUT SKIP(2) DATA (WINTER_ANNUAL_SEED,AUTUMN_MOISTURE_FACTOR,
                                WINTER_GERMINATION_BIOMASS,WINTER_GREEN_VEG);
198                     END;
```

```
                /*                                                              */
                /******************************************************************/
                /*                                                              */
                /*      GROWTH OF WINTER ANNUALS (GROWTH CAN TAKE PLACE IN AUTUMN,  */
                /*           WINTER, OR SPRING -- ALL THAT'S REQUIRED IS THAT SOME  */
                /*           GERMINATION HAS OCCURRED)                          */
                /*                                                              */
                /******************************************************************/
                /*                                                              */
199             IF WINTER_GREEN_VEG ¬= 0 THEN DO;
201                 WINTER_TEMP_FACTOR=EXP(-.5*((OPTIMUM_WINTER_ANNUAL_TEMP
                        -WEEKLY_MEAN_TEMP(CURRENT_WEEK))/
                            WINTER_GROWTH_TEMP_WIDTH)**2);
                                /* SEE FIGURE 7 FOR THE FOLLOWING FUNCTION  */
202                 WINTER_SOIL_WATER_FACTOR=CURVE(SOIL_WATER,SOIL_WATER_AXIS,
                        WINTER_SOIL_WATER_AXIS);
203                 WINTER_GROWTH_RATE=MAX_WINTER_GROWTH_RATE*WINTER_TEMP_FACTOR
                                    *WINTER_SOIL_WATER_FACTOR;
204                 WINTER_WEEKLY_PRODUCTION=WINTER_GROWTH_RATE*WINTER_GREEN_VEG;
205                 WINTER_GREEN_VEG=WINTER_GREEN_VEG+WINTER_WEEKLY_PRODUCTION;
206                 IF DEBUG_RUN THEN
207                    PUT SKIP(2) DATA (WINTER_TEMP_FACTOR,WINTER_SOIL_WATER_FACTOR,
                        WINTER_GROWTH_RATE,WINTER_WEEKLY_PRODUCTION,
                        WINTER_GREEN_VEG,WEEKLY_MEAN_TEMP(CURRENT_WEEK));
208                 END;
                /*                                                              */
                /******************************************************************/
                /*                                                              */
                /*      SEED SET FOR WINTER ANNUALS                             */
                /*                                                              */
                /******************************************************************/
                /*                                                              */
209             IF CURRENT_WEEK = WEEK_SUMMER_BEGINS THEN DO;
                                /* SEE FIGURE 8 FOR THE FOLLOWING FUNCTION  */
211                 WINTER_SEED_CROP = CURVE(WINTER_GREEN_VEG,
                        WINTER_GREEN_VEG_AXIS,WINTER_SEED_CROP_AXIS);
212                 WINTER_ANNUAL_SEED=WINTER_ANNUAL_SEED+WINTER_SEED_CROP;
213                 WINTER_GREEN_VEG=0;
214                 WINTER_WEEKLY_PRODUCTION=0;
215                 IF DEBUG_RUN THEN
216                    PUT SKIP(2) DATA (WINTER_SEED_CROP,WINTER_ANNUAL_SEED,
                        WINTER_GREEN_VEG);
217                 END;
```

```
                /*                                                                    */
                /**********************************************************************/
                /*                                                                    */
                /*      OUTPUT SECTION                                                */
                /*                                                                    */
                /**********************************************************************/
                /*                                                                    */
218             IF WINTER_GREEN_VEG ¬= 0 THEN DO;
220                 GERMINATION_BIOMASS=WINTER_GERMINATION_BIOMASS;
221                 PRODUCTION=WINTER_WEEKLY_PRODUCTION;
222                 VEG=WINTER_GREEN_VEG;
223                 WINTER_WEEKLY_PRODUCTION=0;
224                 WINTER_GERMINATION_BIOMASS=0;
225                 END;
226               ELSE DO;
227                 GERMINATION_BIOMASS=SUMMER_GERMINATION_BIOMASS;
228                 PRODUCTION=SUMMER_WEEKLY_PRODUCTION;
229                 VEG=SUMMER_GREEN_VEG;
230                 SUMMER_WEEKLY_PRODUCTION=0;
231                 SUMMER_GERMINATION_BIOMASS=0;
232                 END;
233             IF REPORT=1 THEN
234                     CALL PRTPLT(CURRENT_WEEK,WEEKLY_MEAN_TEMP(CURRENT_WEEK),
                            SOIL_WATER,GERMINATION_BIOMASS,PRODUCTION,VEG,
                            WINTER_ANNUAL_SEED,BOUNDS);
235             IF REPORT=2 THEN DO;
237                 IF WINTER THEN SEASON='WINTER';
239                     ELSE IF SPRING THEN SEASON='SPRING';
241                     ELSE IF SUMMER THEN SEASON='SUMMER';
243                     ELSE SEASON='AUTUMN';
244                 OUTPUT_LINE=OUTPUT_LINE+1;
245                 PUT STRING (OUTPUT_DATA(OUTPUT_LINE)) EDIT (YEAR_OF_RUN,
                        CURRENT_WEEK,SEASON, WEEKLY_MEAN_TEMP(CURRENT_WEEK),
                        RAIN_THIS_WEEK,POTENTIAL_EVAPOTRANSPIRATION,
                        ACTUAL_EVAPOTRANSPIRATION,SOIL_WATER,GERMINATION_BIOMASS,
                        PRODUCTION, VEG, WINTER_ANNUAL_SEED, SUMMER_ANNUAL_SEED)
                        (X(4),F(2),X(5),F(2),X(4),A,X(3),F(4,1),X(3),F(5,1),X(2),
                        F(4,1),X(3),F(4,1),X(4),F(6,2),X(5),F(6,3),X(8),F(5,2),X(8),
                        F(5,2),X(7),F(4,1),X(5),F(4,1));
246                 END;
247             IF REPORT=2 THEN
248                 IF MOD(CURRENT_WEEK_OF_RUN,52) = 0 THEN DO;
250                     PUT PAGE;
251                     DO LINE=1 TO OUTPUT_LINE;
252                         PUT EDIT (OUTPUT_DATA(LINE)) (SKIP,A);
253                         END;
254                     OUTPUT_LINE=4;
255                     END;
```

```
                /*                                                              */
                /*****************************************************************/
                /*                                                              */
                /*      END OF THE WEEKLY TIMING LOOP                           */
                /*                                                              */
                /*****************************************************************/
                /*                                                              */
256                 END;
                /*                                                              */
                /*****************************************************************/
                /*                                                              */
                /*      RESET CERTAIN VARIABLES, THEN GO TO THE BEGINNING OF THE */
                /*           PROGRAM AND READ IN DATE OF THE NEXT RAINFALL EVENT */
                /*                                                              */
                /*****************************************************************/
                /*                                                              */
257             RAIN=0;
258             IT_HAS_RAINED = FALSE;
259             WEEK_OF_LAST_RAIN = CURRENT_WEEK_OF_RUN;
260             GO TO NEW_DATE;
```

```
/*                                                                    */
/**********************************************************************/
/*                                                                    */
/*      HERE IS A COMPLETE LISTING OF THE DATA DECK USED TO           */
/*           GENERATE THE REPORT=2 OUTPUT:                            */
/*                                                                    */
/*      .0183 .0187 .0191 .0194 .0198 .0202 .0205 .0209               */
/*      .0212 .0215 .0217 .0220 .0222 .0223 .0224 .0225               */
/*      .0225 .0225 .0224 .0223 .0222 .0220 .0217 .0215               */
/*      .0212 .0209 .0205 .0202 .0198 .0194 .0191 .0187               */
/*      .0183 .0180 .0177 .0174 .0171 .0169 .0167 .0166               */
/*      .0164 .0164 .0164 .0164                                       */
/*      10.2 10.0 10.0 10.0 10.2 10.6 11.1 11.7 12.5 13.3             */
/*      14.3 15.3 16.4 17.6 18.7 20.0 21.2 22.3 23.5 24.6             */
/*      25.6 26.6 27.4 28.2 28.8 29.3 29.7 29.9 30.0 29.9             */
/*      29.7 29.3 28.8 28.2 27.4 26.6 25.6 24.6 23.5 22.3             */
/*      21.2 19.9 18.7 17.6 16.4 15.3 14.3 13.3 12.5 11.7             */
/*      11.1 10.6                                                     */
/*    #_YEARS_TO_RUN=1,FUNCTIONS_DESIRED='1'B,REPORT=2;               */
/*    NEXT_WEEK= 2, RAIN=  1.0;                                       */
/*    10 22 38 50                                                     */
/*    NEXT_WEEK= 4, RAIN=  4.6;                                       */
/*    NEXT_WEEK= 6, RAIN=  8.6;                                       */
/*    NEXT_WEEK= 7, RAIN=  7.1;                                       */
/*    NEXT_WEEK= 9, RAIN=  5.1;                                       */
/*    NEXT_WEEK=10, RAIN= 23.6;                                       */
/*    NEXT_WEEK=15, RAIN=  1.8;                                       */
/*    NEXT_WEEK=27, RAIN= 10.1;                                       */
/*    NEXT_WEEK=29, RAIN=  1.5;                                       */
/*    NEXT_WEEK=30, RAIN= 32.2;                                       */
/*    NEXT_WEEK=31, RAIN= 45.9;                                       */
/*    NEXT_WEEK=32, RAIN=  6.3;                                       */
/*    NEXT_WEEK=33, RAIN=  2.5;                                       */
/*    NEXT_WEEK=34, RAIN= 10.7;                                       */
/*    NEXT_WEEK=40, RAIN= 11.9;                                       */
/*    NEXT_WEEK=46, RAIN= 31.0;                                       */
/*    NEXT_WEEK=51, RAIN=  8.9;                                       */
/*    NEXT_WEEK=52, RAIN= 50.1;                                       */
/*    NEXT_WEEK= 2, RAIN= 22.9;                                       */
/*    10 22 38 50                                                     */
/*    NEXT_WEEK= 3, RAIN= 15.0;                                       */
/*    NEXT_WEEK= 6, RAIN=  4.8;                                       */
/*    NEXT_WEEK= 7, RAIN=  4.9;                                       */
/*    NEXT_WEEK= 8, RAIN= 10.7;                                       */
/*    NEXT_WEEK=10, RAIN=  5.8;                                       */
/*    NEXT_WEEK=11, RAIN=  1.5;                                       */
/*    NEXT_WEEK=12, RAIN=  5.4;                                       */
/*    NEXT_WEEK=15, RAIN=  9.1;                                       */
/*    NEXT_WEEK=18, RAIN=  6.6;                                       */
/*    NEXT_WEEK=19, RAIN=  0.8;                                       */
/*    NEXT_WEEK=27, RAIN=  1.5;                                       */
/*    NEXT_WEEK=28, RAIN=  1.5;                                       */
/*    NEXT_WEEK=29, RAIN= 16.1;                                       */
/*    NEXT_WEEK=30, RAIN= 15.2;                                       */
/*    NEXT_WEEK=31, RAIN=  4.3;                                       */
```

```
/*    NEXT_WEEK=32, RAIN=  4.1;                                    */
/*    NEXT_WEEK=33, RAIN= 38.1;                                    */
/*    NEXT_WEEK=34, RAIN=  6.9;                                    */
/*    NEXT_WEEK=35, RAIN=  0.5;                                    */
/*    NEXT_WEEK=36, RAIN= 18.3;                                    */
/*    NEXT_WEEK=37, RAIN=  9.7;                                    */
/*    NEXT_WEEK=42, RAIN=  2.8;                                    */
/*    NEXT_WEEK=43, RAIN=  3.0;                                    */
/*    NEXT_WEEK=45, RAIN= 23.1;                                    */
/*    NEXT_WEEK=46, RAIN= 35.1;                                    */
/*    NEXT_WEEK=48, RAIN=  2.5;                                    */
/*    NEXT_WEEK=49, RAIN= 10.9;                                    */
/*    NEXT_WEEK=50, RAIN=  2.5;                                    */
/*    NEXT_WEEK=52, RAIN= 10.2;                                    */
/*    NEXT_WEEK= 1;                                                */
/*                                                                 */
/*******************************************************************/
/*                                                                 */
```

```
                /*                                                                    */
                /**********************************************************************/
                /*                                                                    */
                /*      PUT_CURVE PROCEDURE                                           */
                /*                                                                    */
                /**********************************************************************/
                /*                                                                    */
261             PUT_CURVE: PROC(H,XVAL,YVAL);
262                        DCL XVAL(*),YVAL(*),H CHAR(*),GRAPH(20,40)CHAR(1);
263                        DCL(ISTORE,JSTORE)(20);
                /*THIS MAY BE TOO SMALL IN SOME CASES*/
264                        NDIM=DIM(XVAL,1);
265                           IF NDIM>20 THEN
266                              DO;
267                              PUT SKIP LIST(H, ' TOO SMALL');
268                              RETURN;
269                              END;
270                        XMIN=XVAL(1);
271                        XMAX=XVAL(NDIM);
272                        YMAX=YVAL(1);
273                        YMIN=YVAL(1);
274                           DO II=2 TO NDIM;
275                           YMIN=MIN(YMIN,YVAL(II));
276                           YMAX=MAX(YMAX,YVAL(II));
277                           END;
278                        XDIV=(XMAX-XMIN)*0.025;
279                        YDIV=(YMAX-YMIN)*0.05;
280                        GRAPH(*,*)=' ';
281                           DO K=1 TO NDIM;
282                           J=CEIL((XVAL(K)-XMIN)/XDIV-0.5);
283                              IF J>40 THEN
284                              J=40;
285                              IF J<1 THEN
286                              J=1;
287                           I=CEIL((YMAX-YVAL(K))/YDIV-0.5);
288                              IF I>20 THEN
289                              I=20;
290                              IF I<1 THEN
291                              I=1;
292                           ISTORE(K)=I;
293                           JSTORE(K)=J;
294                           GRAPH(I,J)='*';
295                           END;
296                           DO N=2 TO NDIM;
297                              DO J=JSTORE(N-1)+1 TO JSTORE(N)-1;
298                              GRAD=(ISTORE(N)-ISTORE(N-1))/(JSTORE(N)-JSTORE(N-1));
299                              C=ISTORE(N)-(GRAD*JSTORE(N));
300                              I=CEIL(GRAD*J+C);
301                              GRAPH(I,J)='+';
302                              END;
303                           END;
304                        PUT SKIP(2);
305                           DO M=1 TO 20;
306                              IF M=1 THEN
307                              PUT SKIP EDIT(YMAX,'+',GRAPH(1,*))
```

```
                                        (COL(20),E(11,4),COL(34),A,COL(35),40 A);
308                              ELSE
308                                 IF M=20 THEN
309                                 PUT SKIP EDIT(YMIN,'+',GRAPH(20,*))
                                              (COL(20),E(11,4),COL(34),A,COL(35),
                                               40 A);
310                                 ELSE
310                                 PUT SKIP EDIT('|',GRAPH(M,*))
                                              (COL(34),A,COL(35),40 A);
311                              END;
312                         PUT SKIP EDIT('+_____________________________________+')
                                         (COL(35),A);
313                         PUT SKIP EDIT(XMIN,XMAX)(COL(35),E(11,4),COL(64),E(11,4));
314                         PUT SKIP(2)EDIT(H)(COL(29),A);
315                         END PUT_CURVE;
                       /*                                                      */
                       /**********************************************************/
                       /*                                                      */
                       /*      INTERPOLATION PROCEDURE                         */
                       /*                                                      */
                       /**********************************************************/
                       /*                                                      */
316                    CURVE : PROC(X,XVAL,YVAL);
317                    DCL I FIXED BIN INTERNAL, X FLOAT DEC;
318                    DCL XVAL(*), YVAL(*);
319                      NDIM=DIM(XVAL,1);
320                    IF X <= XVAL(1) THEN RETURN (YVAL(1));
322                    IF X >= XVAL(NDIM) THEN RETURN (YVAL(NDIM));
324                    DO I = 1 TO NDIM;
325                      IF XVAL(I) > X THEN DO;
327                           AM = (YVAL(I)-YVAL(I-1))/(XVAL(I)-XVAL(I-1));
328                           C = YVAL(I)-AM*XVAL(I);
329                           RETURN (AM*X+C);
330                      END;
331                    END;
332                    END CURVE;
```

```
                /*                                                              */
                /****************************************************************/
                /*                                                              */
                /*      GRAPHING PROCEDURE                                      */
                /*                                                              */
                /****************************************************************/
                /*                                                              */
333             PRTPLT: PROC (X,W1,W2,W3,W4,W5,W6,BOUNDS);
334             DCL BOUNDS (*) FLOAT DEC;
335             DCL HEADING CHAR (40) VAR INIT
                    ('GRAPH OF ANNUALS MODEL OUTPUT');
336             DCL YLABEL (6) CHAR (11) VAR INIT
                    ('TEMPERATURE','MOISTURE','GERMINATION','PRODUCTION','GREEN VEG',
                    'WINTER SEED');
337             DCL XLABEL CHAR (10) VAR INIT
                    ('WEEK');
338             DCL ALINE CHAR (70);
339             DCL SYMB (6) CHAR (1) INIT
                    ('T','M','G','P','V','S');
340             DCL X FIXED BIN;
341             DCL XV (6) FLOAT DEC;
342             DCL NDIV (6) FIXED BIN (31,0);
343             DCL DIV (6) FLOAT DEC STATIC;
344             DCL FIRST_PASS BIT (1) STATIC INIT ('1'B);
345             IF FIRST_PASS THEN DO;
347             FIRST_PASS = '0'B;
                /*      HEADING                                                 */
348             PUT EDIT (HEADING) (COL(33),A);
349             PUT SKIP (3) EDIT ('SYMBOL','MINIMUM','GRAPHING','MAXIMUM')
                (COL(19),A,COL(33),A,COL(59),A,COL(92),A);
350             DO I = 1 TO 6;
351             J = 2 * I;
352             PUT SKIP EDIT (SYMB(I),BOUNDS(J),YLABEL(I),'.VS.',XLABEL,BOUNDS(J-1))
                    (COL(21),A,
                     COL(31),E(11,4),COL(51),A,COL(64),A,COL(71),A,COL(90),E(11,4));
353             END;
354             PUT SKIP(2) EDIT (XLABEL,YLABEL(1)) (COL(5),A,COL(15),A);
355             PUT SKIP (2);
                /*      SCALE                                                   */
356             DO I = 1 TO 6;
357             J = 2 * I;
358             DIV(I) = (BOUNDS(J-1)-BOUNDS(J))/70;
359             END;
360             END; /* END FIRST_PASS SECTION */
361             XV(1) = W1;
362             XV(2) = W2;
363             XV(3) = W3;
364             XV(4) = W4;
365             XV(5) = W5;
366             XV(6) = W6;
367             ALINE = '
                        ';
368             SUBSTR(ALINE,1,1)='.';
369             SUBSTR(ALINE,70,1)='.';
370             DO I = 1 TO 6;
```

```
371             J = 2 * I;
372             NDIV(I) = (XV(I)-BOUNDS(J))/DIV(I) + 0.5;
373             IF NDIV (I) <= 1 THEN NDIV(I) = 1;
375             IF NDIV (I) > 70 THEN NDIV(I) = 70;
377             SUBSTR(ALINE,NDIV(I),1)=SYMB(I);
378             END;
379             PUT EDIT (X,XV(1),ALINE)
                    (COL(4),F(4,0),COL(18),F(5,2),COL(31), A);
380             RETURN;
381             END PRTPLT;
382             END_PROGRAM:
                    PUT PAGE;
383                 END ANNUALS;       /*  END OF PROGRAM  */
```

ATTRIBUTE AND CROSS-REFERENCE TABLE

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
	ACTUAL_EVAPOTRANSPIRATION	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 149,150,152,245
338	ALINE	AUTOMATIC,UNALIGNED,STRING(70),CHARACTER 367,368,369,377,379
	AM	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 327,328,329
	ANNUAL_COVER_COEFFICIENT	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 146,147
1	ANNUALS	ENTRY,DECIMAL,FLOAT(SINGLE)
3	AUTUMN	AUTOMATIC,UNALIGNED,INITIAL,STRING(1),BIT 125,190
	AUTUMN_MOISTURE_FACTOR	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 192,193,197
4	AUTUMN_MOISTURE_FACTOR_AXIS	(4)AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 92,192
334	BOUNDS	(*)PARAMETER,ALIGNED,DECIMAL,FLOAT(SINGLE) 333,352,352,358,358,372
5	BOUNDS	(12)AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 234
	C	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 299,300,328,329
	CEIL	GENERIC,BUILT-IN FUNCTION 282,287,300
6	COEFFICIENT_AXIS	(2)AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 85,146
	COVER_COEFFICIENT	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 147,148
7	********* CURRENT_WEEK	AUTOMATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 107,111,112,122,122,123,123,124,124,125,125,148,148,164,170,172,201 207,209,234,234,245,245
8	********* CURRENT_WEEK_OF_RUN	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 105,106,107,118,248,259
316	CURVE	ENTRY,DECIMAL,FLOAT(SINGLE)

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
		146,155,165,174,183,192,202,211
9	DEBUG_RUN	AUTOMATIC,UNALIGNED,STRING(1),BIT 72,99,110,115,133,139,151,159,169,178,187,196,206,215
	DEEP_PERCOLATION	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 137,140
	DIM	GENERIC,BUILT-IN FUNCTION 264,315
343	DIV	(6)STATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 358,372
382	END_PROGRAM	STATEMENT LABEL CONSTANT 109
	EXP	GENERIC,BUILT-IN FUNCTION 164,201
10	FALSE	AUTOMATIC,UNALIGNED,INITIAL,STRING(1),BIT 72,73,258
	FIELD_CAPACITY	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 54,135,137,138,149
344	FIRST_PASS	STATIC,UNALIGNED,INITIAL,STRING(1),BIT 345,347
11	FUNCTIONS_DESIRED	AUTOMATIC,UNALIGNED,STRING(1),BIT 73,83
	GERMINATION_BIOMASS	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 220,227,234,245
	GRAD	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 298,299,300
262	GRAPH	(20,40)AUTOMATIC,UNALIGNED,STRING(1),CHARACTER 280,294,301,307,309,310
262	H	PARAMETER,UNALIGNED,STRING(*),CHARACTER 261,267,314
335	HEADING	AUTOMATIC,UNALIGNED,INITIAL,STRING(40),CHARACTER,VARYING 348
317	********* I	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 324,325,327,327,327,327,328,328
	********* I	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 287,288,289,290,291,292,294,300,301,350,351,352,352,356,357,358,370

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
		371,372,372,372,373,374,375,376,377,377
*********	II	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 274,275,276
12	INFILTRATION	AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 129,130,131,132,134
263 *********	ISTORE	(20)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 292,298,298,299
15	IT_HAS_RAINED	AUTOMATIC,UNALIGNED,INITIAL,STRING(1),BIT 118,119,126,153,181,190,258
2 *********	#_YEARS_TO_RUN	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 71,108
*********	J	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 282,283,284,285,286,293,294,297,300,301,351,352,352,357,358,358,371 372
263 *********	JSTORE	(20)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 293,297,297,298,298,299
*********	K	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 281,282,287,292,293
*********	LINE	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 251,252
*********	M	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 305,306,308,310
	MAX	GENERIC,BUILT-IN FUNCTION 276
13	MAX_FRACTION_SUMMER_GERMINATION	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 55,156
14	MAX_FRACTION_WINTER_GERMINATION	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 56,184,193
16	MAX_SUMMER_GROWTH_RATE	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 57,166
17	MAX_WINTER_GROWTH_RATE	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 58,203
	MIN	GENERIC,BUILT-IN FUNCTION 275
	MOD	GENERIC,BUILT-IN FUNCTION

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
		248
*********	N	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 296,297,297,298,298,298,298,299,299
*********	NDIM	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 264,265,271,274,281,296,319,322,323,324
342	NDIV	(6)AUTOMATIC,ALIGNED,BINARY,FIXED(31,0) 372,373,374,375,376,377
98	NEW_DATE	STATEMENT LABEL CONSTANT 260
*********	NEXT_WEEK	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 100,101,103,104
	OPTIMUM_SUMMER_ANNUAL_TEMP	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 59,164
	OPTIMUM_WINTER_ANNUAL_TEMP	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 60,201
18	OUTPUT_DATA	(60)AUTOMATIC,UNALIGNED,STRING(130),CHARACTER,VARYING 79,80,81,245,252
19 *********	OUTPUT_LINE	AUTOMATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 244,244,245,251,254
	PERENNIAL_COVER_COEFFICIENT	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 61,147
	POTENTIAL_EVAPOTRANSPIRATION	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 148,149,152,245
	PRODUCTION	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 221,228,234,245
333	PRTPLT	ENTRY,DECIMAL,FLOAT(SINGLE) 234
261	PUT_CURVE	ENTRY,DECIMAL,FLOAT(SINGLE) 85,86,88,89,91,92,94,95
	RAIN	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 62,100,120,257
20	RAIN_AXIS	(4)AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 86,91,92,155,183,192
21	RAIN_THIS_WEEK	AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 120,121,129,130,131,134,155,183,192,245

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
	REPORT	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 74,77,233,235,247
22	RUNOFF	AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 131,134
23	SEASON	AUTOMATIC,UNALIGNED,STRING(6),CHARACTER 238,240,242,243,245
	SOIL_DEPTH	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 63,132,137,150
	SOIL_WATER	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 64,132,132,134,135,137,138,140,149,150,150,152,165,202,234,245
24	SOIL_WATER_AXIS	(5)AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 88,94,165,202
25	SPRING	AUTOMATIC,UNALIGNED,INITIAL,STRING(1),BIT 123,181,239
	SPRING_MOISTURE_FACTOR	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 183,184,188
26	SPRING_MOISTURE_FACTOR_AXIS	(4)AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 91,183
	SUBSTR	GENERIC,BUILT-IN FUNCTION 368,369,377
27	SUMMER	AUTOMATIC,UNALIGNED,INITIAL,STRING(1),BIT 124,128,143,153,241
	SUMMER_ANNUAL_SEED	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 65,156,157,157,160,175,175,179,245
28	SUMMER_GERMINATION_BIOMASS	AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 156,157,158,160,227,231
	SUMMER_GREEN_VEG	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 66,144,158,158,160,162,167,168,168,170,174,176,179,229
29	SUMMER_GREEN_VEG_AXIS	(3)AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 89,174
	SUMMER_GROWTH_RATE	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 166,167,170
	SUMMER_GROWTH_TEMP_WIDTH	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 67,164

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
30	SUMMER_INFILTRATING_FRACTION	AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 129
	SUMMER_MOISTURE_FACTOR	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 155,156,160
31	SUMMER_MOISTURE_FACTOR_AXIS	(4)AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 86,155
32	SUMMER_SEED_CROP	AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 174,175,179
33	SUMMER_SEED_CROP_AXIS	(3)AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 89,174
34	SUMMER_SOIL_WATER_AXIS	(5)AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 88,165
	SUMMER_SOIL_WATER_FACTOR	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 165,166,170
	SUMMER_TEMP_FACTOR	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 164,166,170
35	SUMMER_WEEKLY_PRODUCTION	AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 167,168,170,177,228,230
339	SYMB	(6)AUTOMATIC,UNALIGNED,INITIAL,STRING(1),CHARACTER 352,377
	SYSIN	FILE,EXTERNAL 75,76,98,114
	SYSPRINT	FILE,EXTERNAL 53,87,90,93,96,100,111,116,134,140,152,160,170,179,188,197,207,216 250,252,267,304,307,309,310,312,313,314,348,349,352,354,355,379,382
36	THIS_WEEK	AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FIXED(5,0) 101,103
37	TRUE	AUTOMATIC,UNALIGNED,INITIAL,STRING(1),BIT
	VEG	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 144,145,146,222,229,234,245
38	VEG_AXIS	(2)AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 85,146
	W1	PARAMETER,ALIGNED,DECIMAL,FLOAT(SINGLE) 333,361
	W2	PARAMETER,ALIGNED,DECIMAL,FLOAT(SINGLE)

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
		333,362
	W3	PARAMETER,ALIGNED,DECIMAL,FLOAT(SINGLE) 333,363
	W4	PARAMETER,ALIGNED,DECIMAL,FLOAT(SINGLE) 333,364
	W5	PARAMETER,ALIGNED,DECIMAL,FLOAT(SINGLE) 333,365
	W6	PARAMETER,ALIGNED,DECIMAL,FLOAT(SINGLE) 333,366
	WEEK_AUTUMN_BEGINS	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 114,116,124,125,172
41	********* WEEK_OF_LAST_RAIN	AUTOMATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 105,259
42	********* WEEK_OF_NEXT_RAIN	AUTOMATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 104,105,118
	WEEK_SPRING_BEGINS	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 114,116,122,123
	WEEK_SUMMER_BEGINS	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 114,116,123,124,209
	WEEK_WINTER_BEGINS	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 114,116,122,125
39	WEEKLY_FRACTION_ANNUAL_DAY_HRS	(52)AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 75,148
40	WEEKLY_MEAN_TEMP	(52)AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 75,148,164,170,201,207,234,245
43	WINTER	AUTOMATIC,UNALIGNED,INITIAL,STRING(1),BIT 122,237
	WINTER_ANNUAL_SEED	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 68,184,185,185,188,193,194,194,197,212,212,216,234,245
44	WINTER_GERMINATION_BIOMASS	AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 184,185,186,188,193,194,195,197,220,224
	WINTER_GREEN_VEG	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 69,145,186,186,188,195,195,197,199,204,205,205,207,211,213,216,218 222
45	WINTER_GREEN_VEG_AXIS	(3)AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE)

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
		95,211
	WINTER_GROWTH_RATE	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 203,204,207
	WINTER_GROWTH_TEMP_WIDTH	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 70,201
46	WINTER_INFILTRATING_FRACTION	AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 130
47	WINTER_SEED_CROP	AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 211,212,216
48	WINTER_SEED_CROP_AXIS	(3)AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 95,211
49	WINTER_SOIL_WATER_AXIS	(5)AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 94,202
	WINTER_SOIL_WATER_FACTOR	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 202,203,207
	WINTER_TEMP_FACTOR	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 201,203,207
50	WINTER_WEEKLY_PRODUCTION	AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FLOAT(SINGLE) 204,205,207,214,221,223
340 *********	X	PARAMETER,ALIGNED,BINARY,FIXED(15,0) 333,379
317	X	PARAMETER,ALIGNED,DECIMAL,FLOAT(SINGLE) 316,320,322,325,329
	XDIV	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 278,282
337	XLABEL	AUTOMATIC,UNALIGNED,INITIAL,STRING(10),CHARACTER,VARYING 352,354
	XMAX	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 271,278,313
	XMIN	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 270,278,282,313
341	XV	(6)AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 361,362,363,364,365,366,372,379
318	XVAL	(*)PARAMETER,ALIGNED,DECIMAL,FLOAT(SINGLE) 316,319,320,322,325,327,327,328

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
262		XVAL	(*)PARAMETER,ALIGNED,DECIMAL,FLOAT(SINGLE) 261,264,270,271,282
		YDIV	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 279,287
51	*********	YEAR_OF_NEXT_RAIN	AUTOMATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 102,102,104
52	*********	YEAR_OF_RUN	AUTOMATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 106,107,108,245
336		YLABEL	(6)AUTOMATIC,UNALIGNED,INITIAL,STRING(11),CHARACTER,VARYING 352,354
		YMAX	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 272,276,276,279,287,307
		YMIN	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 273,275,275,279,309
318		YVAL	(*)PARAMETER,ALIGNED,DECIMAL,FLOAT(SINGLE) 316,321,323,327,327,328
262		YVAL	(*)PARAMETER,ALIGNED,DECIMAL,FLOAT(SINGLE) 261,272,273,275,276,287

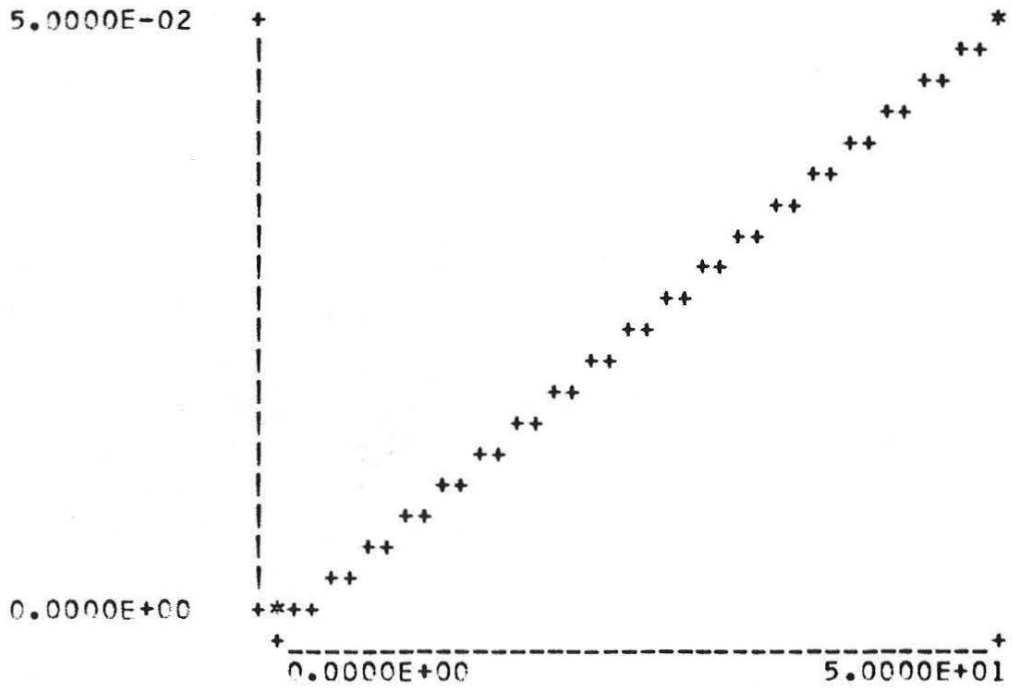

FIG. 1, ANNUAL VEG(X) VS COVER COEF(Y)

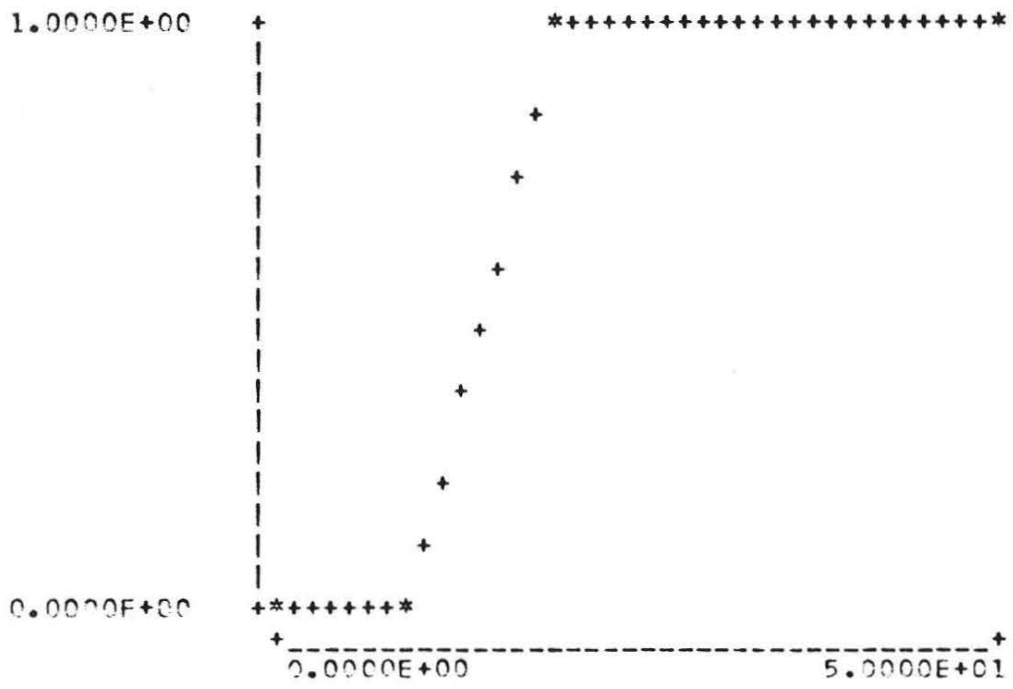

FIG. 2, RAINFALL(X) VS SUMMER GERMINATION FACTOR(Y)

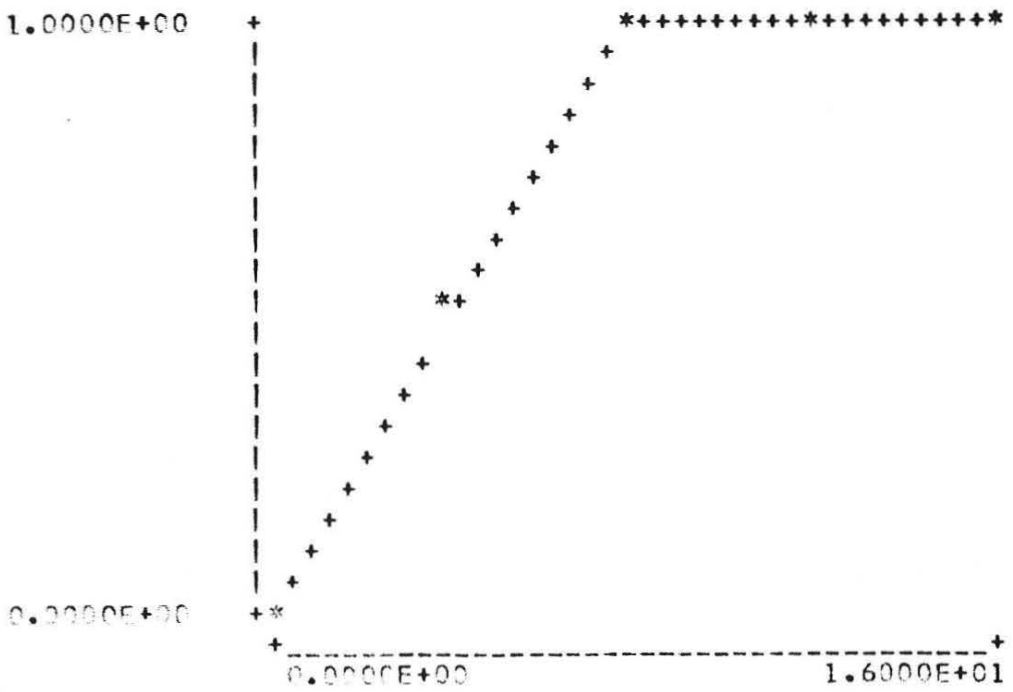

FIG. 3, SOIL WATER(X) VS SUMMER GROWTH FACTOR(Y)

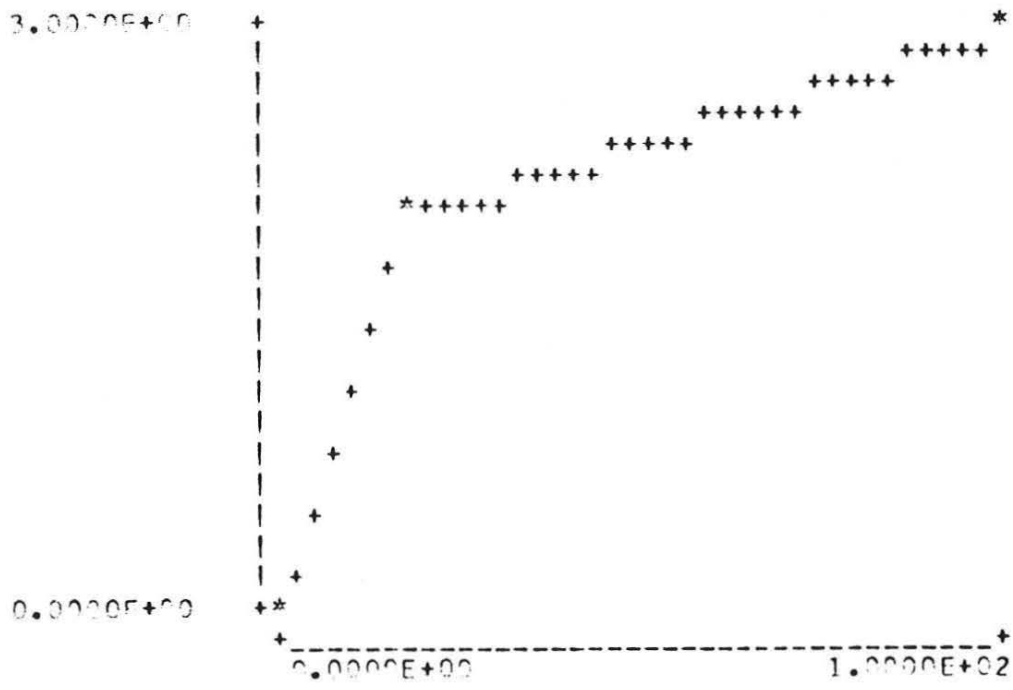

FIG. 4, SUMMER GREEN VEG(X) VS SEED CROP(Y)

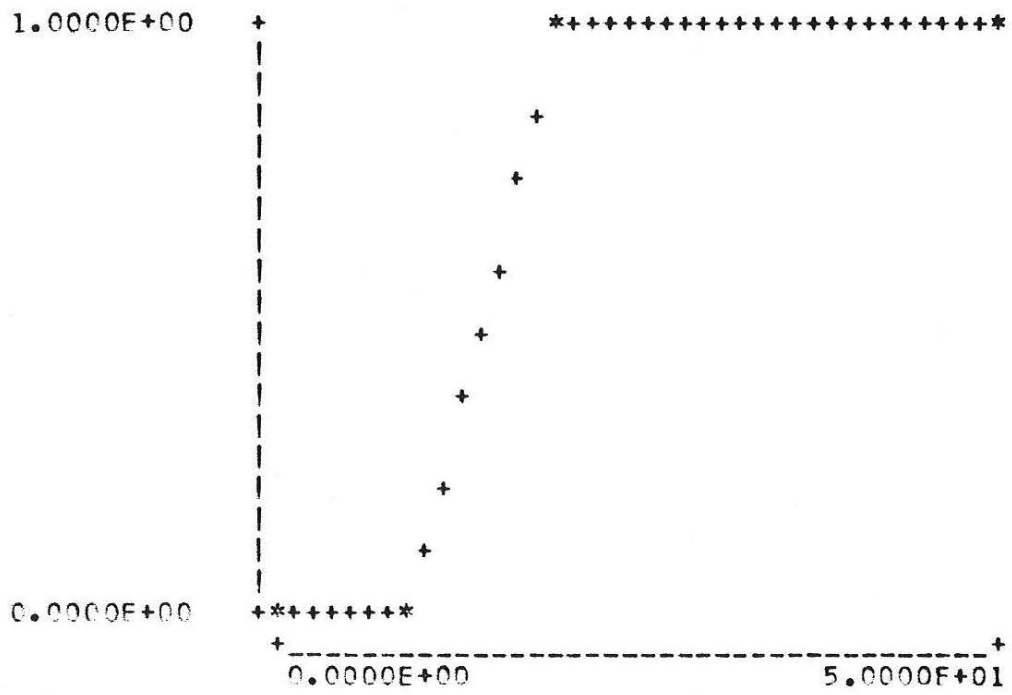

FIG. 5, RAINFALL(X) VS SPRING GERMINATION FACTOR(Y)

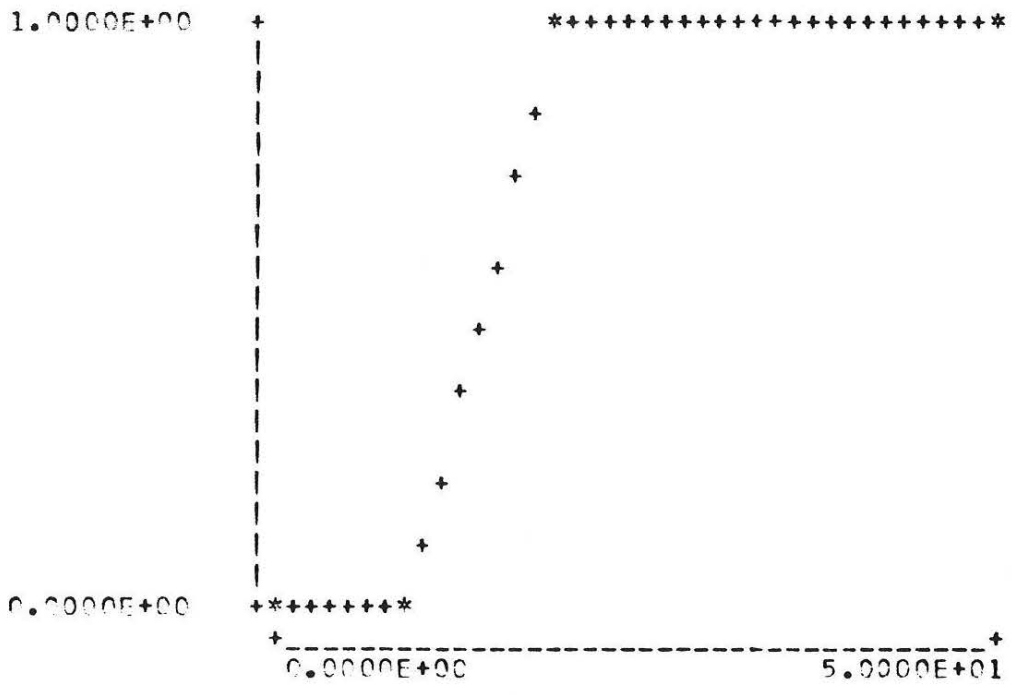

FIG. 6, RAINFALL(X) VS AUTUMN GERMINATION FACTOR(Y)

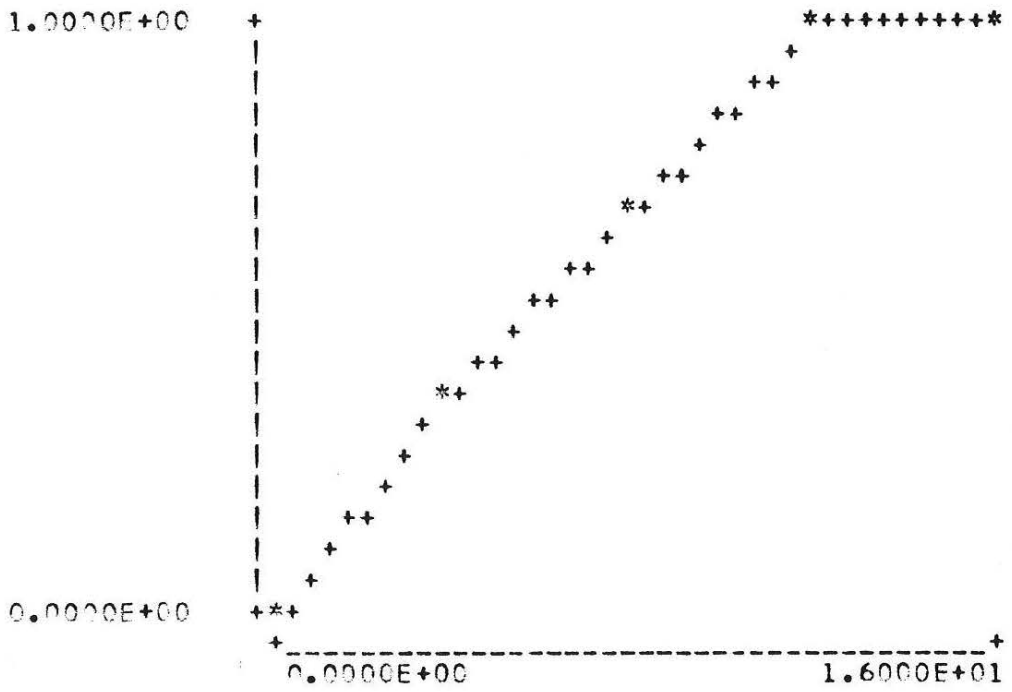

FIG. 7, SOIL WATER(X) VS WINTER GROWTH FACTOR(Y)

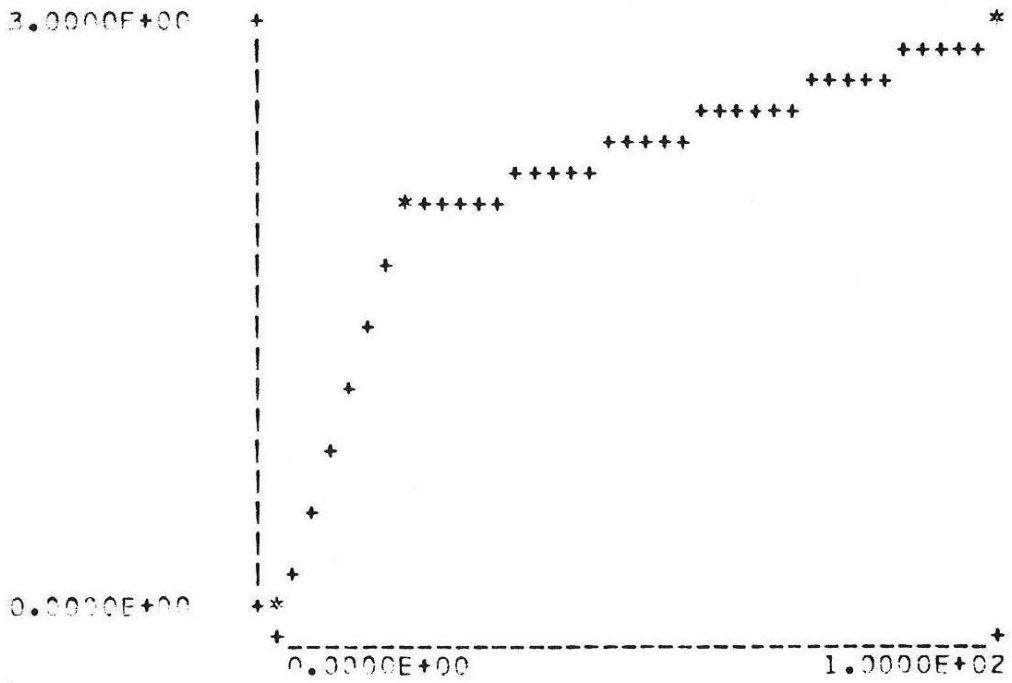

FIG. 8, WINTER GREEN VEG(X) VS SEED CROP(Y)

YEAR	WEEK	SEASON	TEMP (C)	RAIN (MM)	PET (MM)	AET (MM)	SOIL WATER (%)	GERMINATION BIOMASS (G/SQ.M)	WEEKLY PRODUCTION (G/SQ.M)	GREEN VEGETATION (G/SQ.M)	WINTER SEED (G/SQ.M)	SUMMER SEED (G/SQ.M)
1	1	WINTER	10.2	0.0	5.3	4.4	13.53	0.000	0.09	3.09	1.0	1.0
1	2	WINTER	10.0	1.0	5.3	4.1	12.43	0.000	0.08	3.17	1.0	1.0
1	3	WINTER	10.0	0.0	5.4	3.7	11.19	0.000	0.08	3.25	1.0	1.0
1	4	WINTER	10.0	4.6	5.4	3.8	11.17	0.000	0.08	3.33	1.0	1.0
1	5	WINTER	10.2	0.0	5.5	3.4	10.02	0.000	0.09	3.42	1.0	1.0
1	6	WINTER	10.6	8.6	5.7	3.9	11.01	0.000	0.13	3.55	1.0	1.0
1	7	WINTER	11.1	7.1	5.9	4.2	11.49	0.000	0.20	3.74	1.0	1.0
1	8	WINTER	11.7	0.0	6.2	3.9	10.18	0.000	0.27	4.01	1.0	1.0
1	9	WINTER	12.5	5.1	6.4	4.1	10.16	0.000	0.44	4.45	1.0	1.0
1	10	SPRING	13.3	23.6	6.8	6.2	14.40	0.100	0.85	5.40	0.9	1.0
1	11	SPRING	14.3	0.0	7.2	5.7	12.49	0.000	1.44	6.84	0.9	1.0
1	12	SPRING	15.3	0.0	7.5	5.2	10.75	0.000	2.09	8.93	0.9	1.0
1	13	SPRING	16.4	0.0	8.0	4.8	9.15	0.000	2.68	11.61	0.9	1.0
1	14	SPRING	17.6	0.0	8.5	4.3	7.70	0.000	2.94	14.55	0.9	1.0
1	15	SPRING	18.7	1.8	9.0	4.1	6.81	0.000	2.82	17.37	0.9	1.0
1	16	SPRING	20.0	0.0	9.7	3.7	5.60	0.000	1.96	19.33	0.9	1.0
1	17	SPRING	21.2	0.0	10.2	3.2	4.54	0.000	1.09	20.42	0.9	1.0
1	18	SPRING	22.3	0.0	10.7	2.7	3.65	0.000	0.52	20.94	0.9	1.0
1	19	SPRING	23.5	0.0	11.1	2.2	2.90	0.000	0.19	21.13	0.9	1.0
1	20	SPRING	24.6	0.0	11.6	1.9	2.28	0.000	0.06	21.19	0.9	1.0
1	21	SPRING	25.6	0.0	11.9	1.5	1.77	0.000	0.02	21.21	0.9	1.0
1	22	SUMMER	26.6	0.0	11.3	1.1	1.40	0.000	0.00	0.00	2.9	1.0
1	23	SUMMER	27.4	0.0	11.6	0.9	1.10	0.000	0.00	0.00	2.9	1.0
1	24	SUMMER	28.2	0.0	11.8	0.7	0.86	0.000	0.00	0.00	2.9	1.0
1	25	SUMMER	28.8	0.0	12.0	0.6	0.67	0.000	0.00	0.00	2.9	1.0
1	26	SUMMER	29.3	0.0	12.1	0.5	0.52	0.000	0.00	0.00	2.9	1.0
1	27	SUMMER	29.7	10.1	12.2	1.3	1.45	0.001	0.00	0.00	2.9	1.0
1	28	SUMMER	29.9	0.0	12.2	1.0	1.12	0.000	0.00	0.00	2.9	1.0
1	29	SUMMER	30.0	1.5	12.1	0.9	1.02	0.000	0.00	0.00	2.9	1.0
1	30	SUMMER	29.9	32.2	12.0	3.5	4.14	0.100	0.04	0.14	2.9	0.9
1	31	SUMMER	29.7	45.9	11.8	6.7	8.02	0.090	0.17	0.40	2.9	0.8
1	32	SUMMER	29.3	6.3	11.6	5.7	6.96	0.000	0.27	0.68	2.9	0.8
1	33	SUMMER	28.8	2.5	11.3	4.6	5.76	0.000	0.38	1.06	2.9	0.8
1	34	SUMMER	28.2	10.7	11.0	4.4	5.72	0.006	0.61	1.68	2.9	0.8
1	35	SUMMER	27.4	0.0	10.7	3.4	4.59	0.000	0.76	2.44	2.9	0.8
1	36	SUMMER	26.6	0.0	10.3	2.6	3.71	0.000	0.87	3.31	2.9	0.8
1	37	SUMMER	25.6	0.0	9.9	2.1	3.03	0.000	0.89	4.20	2.9	0.8
1	38	AUTUMN	24.6	0.0	9.4	1.6	2.50	0.000	0.00	0.00	2.9	1.3
1	39	AUTUMN	23.5	0.0	9.0	1.3	2.08	0.000	0.00	0.00	2.9	1.3
1	40	AUTUMN	22.3	11.9	8.6	2.5	4.42	0.055	0.00	0.06	2.9	1.3
1	41	AUTUMN	21.2	0.0	8.2	2.0	3.76	0.000	0.00	0.06	2.9	1.3
1	42	AUTUMN	19.9	0.0	7.8	1.6	3.22	0.000	0.00	0.06	2.9	1.3
1	43	AUTUMN	18.7	0.0	7.4	1.3	2.78	0.000	0.00	0.07	2.9	1.3
1	44	AUTUMN	17.6	0.0	7.0	1.1	2.41	0.000	0.01	0.07	2.9	1.3
1	45	AUTUMN	16.4	0.0	6.7	0.9	2.12	0.000	0.01	0.08	2.9	1.3
1	46	AUTUMN	15.3	31.0	6.4	3.7	9.15	0.286	0.10	0.46	2.6	1.3
1	47	AUTUMN	14.3	0.0	6.1	3.1	8.11	0.000	0.08	0.54	2.6	1.3
1	48	AUTUMN	13.3	0.0	5.9	2.7	7.23	0.000	0.06	0.61	2.6	1.3
1	49	AUTUMN	12.5	0.0	5.7	2.3	6.46	0.000	0.04	0.65	2.6	1.3
1	50	WINTER	11.7	0.0	5.5	2.0	5.80	0.000	0.03	0.67	2.6	1.3
1	51	WINTER	11.1	8.9	5.4	2.5	7.35	0.000	0.02	0.70	2.6	1.3
1	52	WINTER	10.6	50.1	5.3	5.3	16.22	0.000	0.03	0.73	2.6	1.3

GRAPH OF ANNUALS MODEL CUTPUT

SYMBOL	MINIMUM	GRAPHING			MAXIMUM
T	0.0000E+00	TEMPERATURE	.VS.	WEEK	4.0000E+01
M	0.0000E+00	MOISTURE	.VS.	WEEK	2.5000E+01
G	0.0000E+00	GERMINATION	.VS.	WEEK	5.0000E-01
P	0.0000E+00	PRODUCTION	.VS.	WEEK	1.0000E+01
V	0.0000E+00	GREEN VEG	.VS.	WEEK	5.0000E+01
S	0.0000E+00	WINTER SEED	.VS.	WEEK	5.0000E+00

WEEK	TEMPERATURE
1	10.20
2	10.00
3	10.00
4	10.00
5	10.20
6	10.60
7	11.10
8	11.70
9	12.50
10	13.30
11	14.30
12	15.30
13	16.40
14	17.60
15	18.70
16	20.00
17	21.20
18	22.30
19	23.50
20	24.60
21	25.60
22	26.60
23	27.40
24	28.20
25	28.80
26	29.30
27	29.70
28	29.90
29	30.00
30	29.90
31	29.70
32	29.30
33	28.80
34	28.20
35	27.40
36	26.60
37	25.60
38	24.60
39	23.50
40	22.30
41	21.20
42	19.90
43	18.70
44	17.60
45	16.40
46	15.30
47	14.30
48	13.30
49	12.50
50	11.70
51	11.10
52	10.60