1973

Sensitivity Analysis

I. Noy-Meir

D. Goodall

RM 73-58

SENSITIVITY ANALYSIS

I. Noy-Meir and D. Goodall

Utah State University

SEPTEMBER 1973

Report Volume 1

Page 2.1.3.1.3.

III

Sensitivity Analysis

Imanuel Noy-Meir and David W. Goodall

Theoretical considerations:

"Sensitivity analysis" of a simulation model means testing the effects of varying certain parameters in the model on certain outcomes or response variables. A "parameter" in this context may be anything that is constant for a single run of the model, e.g. coefficients expressing rates (e.g. maximum infiltration rate) or initial values of state variables (e.g. number of rabbits on Oct. 1). A "response" may be the value of a state variable at some point in time (e.g. biomass of rabbits on July 1) or a function of one or several state variables (e.g. total living biomass of plants on July 1, or value of lambs produced during the season) which is selected as an outcome of prime interest. The "sensitivity" of a response r to a parameter p is defined as the change in r (Δr) produced by a given change in p (Δp). In linear models, the sensitivities of each response to each parameter (expressed as partial derivatives $\frac{\partial r}{\partial p}$) may be calculated directly from coefficient matrices (Kerlin, 1967). But in non-linear models (as most ecological models are) this becomes difficult and the simplest way to evaluate a sensitivity is to run the simulation model with different values of the parameter and record the different values of the response variables obtained.

The sensitivity of a response r to a single parameter p (all other parameters remaining constant) may then be expressed graphically (a plot of r versus p) or numerically (e.g. as the slope of the graph, or the relative change in r produced by a 10% change in p). However, there is the possibility that sensitivity of r to p_1 depends also on the values of other parameters p_2, p_3, etc.. For instance, a 10% increase in the "maximum infiltration rate" may produce a 5% increase in plant production if "root depth" is 20 cm , but a 15% increase if it is 50 cm. In this case the two parameters are said to "interact" in their effects on the response; in other words their sensitivities are not simply additive, or the curves expressing r as a function of p^i, each at a different value of p^i, will not be parallel. When two parameters interact strongly it is not sufficient to test the sensitivity to each of them separately while leaving the other constant; it is necessary to test various combinations as well.

A special kind of sensitivity analysis is "structural" sensitivity analysis, i.e. testing the effect on the responses of deleting an entire component, or link, or interaction from the structure of the model (e.g. neglecting runoff, or the effect of herbivory on primary production, etc.). Often this can be formally treated as a special case of parameter sensitivity analysis, by setting one or several parameters to zero. Its implications may, of course, be more far-reaching. Sensitivity analysis of a simulation model may be used in a number of ways and for different purposes. Two of these uses are discussed:

1) Exploratory sensitivity analysis. The purpose is to test the response of a model over a wide range of conditions, in order to get some feeling for its general behavior and to see whether this behavior is at least qualitatively realistic over this range. In this case, each parameter tested is given values covering a range which represents the range of sites, species, or conditions to which the model is likely to be applied (i.e. infiltration rates characteristic of sandy, silty and rocky soils; maximum gas exchange rates characteristic of annuals, C3-shrubs, C4-shrubs and cacti, etc.). Interactions between parameters may also be tested, using such combinations as are likely to occur in reality. Even if the parameter values used are not accurate values for any particular species or site, the trends in the model responses should indicate at least any peculiarities in the behavior of the model, which might lead to revision of its structure (Noy Meir).

2) Error sensitivity analysis. Once there is reasonable confidence in the general structure and behavior of a model (from validation, exploratory sensitivity analysis, or *a priori* considerations), it may be of interest to establish the main sources of inaccuracy in the prediction of the important response variables. Prediction errors will then in general arise from errors in the estimation of parameters (or initial values of state variables, or climatic input variables). But while small errors in some parameters may cause large differences in predicted responses ("sensitive" parameters), the effect of large inaccuracies in other parameters on the responses may be negligible. The simplest use of sensitivity analysis in this context is then to test three values of each parameter (for each species or site): the best or most likely estimate based on present information (the "standard" value), and the lower and upper limits of the "range of inaccuracy" in this estimation (e.g. an estimated 90% confidence interval). The Δp will then be different for each parameter but it is a comparable measure of the current "level of ignorance" for all parameters.

Thus, the resulting values of Δr express directly the "price of ignorance" for each parameter. Parameters for which Δr is highest will have to be estimated with greater accuracy than the present (even though their absolute accuracy may already be high), while parameters causing a very low Δr are already known accurately enough. Here too, tests of possible interactions between sensitivities of different parameters are necessary before final conclusions can be drawn.

Thus, a thorough error sensitivity analysis can be used to indicate which parameter measurements are to be given highest priority. But the objective validity of such conclusions depends strongly on: (a) The confidence that the structure of the model and all the functions in it are correct and no important effects have been omitted. (b) The selection of response variables, which is a subjective or in any case *a priori* decision.

TECHNICAL PROBLEMS:

There are serious technical difficulties in conducting a sensitivity analysis comprehensive enough to be meaningful on a complex model. The number (N) of runs of the model which are needed rises sharply with the number of parameters to be tested (m) and the number of values tested for each parameter (k), particularly if interactions have also to be considered.

No interactions $N_1 = 1 + m\ (k - 1)$

Pairwise interactions $N_2 = 1 + m\ (k - 1) + m\ \frac{(m-1)(k-1)^2}{2}$

All possible interactions (complete factorial) $N_3 = K^m$

Table 1 illustrates these quantities for different values of m and K.

Table 1.

m	K	N_1	N_2	N_3
5	2	6	16	32
5	3	11	31	243
5	4	16	106	1024
10	2	11	56	1024
10	3	21	111	60000
10	4	31	166	10^b
20	3	41	801	10^6
20	2	21	211	35.10^8
20	4	61	1771	10^{12}

It is obvious that an exhaustive sensitivity analysis is at all feasible only for models which have a rather small number of parameters which need to be tested and for which a single run does not take too long on the computer. The number of runs can be reduced considerably by taking into account only pairwise interactions for those pairs of parameters where there is reason to suspect an important interaction. If h_i is the number of those pairs, the numbers of runs necessary in such a "partial factorial design" (Zusman and Amiad, 1966) is:

$$N = 1 + m\ (k - 1) + h_i\ (k - 1)^2$$

For a "medium sized" ecological model, this may still be many hundreds of runs, but if time for a single run is reasonably short, this is not prohibitively expensive. However, to conduct the sensitivity analysis by many hundreds of separate re-runs of the same computer job, each time changing one parameter, would be rather expensive in both man-time and computer-time. Preparing many hundreds of complete parameter lists for re-running the model within a single job would still be rather inefficient. These problems seem to be part of the reason why many ecological models have never been tested for sensitivities; in cases where it was done (e.g. Goodall, 1970) only a few of the total set of parameters could be tested. The subroutines described below allow an efficient sensitivity analysis of a partial factorial design, with as many model-runs as necessary, in a single job-run and with simple input specifications.

These subroutines presuppose that the quantities to be varied, and those from which response variables are to be drawn, are stored in such a way that they can be addressed as single arrays, by equivalencing, as follows:

P: absolute or time-varying constants of the system, used in calculating process rates

STATE: the state variables of the system

STNG: accumulated exchanges between the ecosystem and its surroundings

SUMS: any sums of state vairables which may be used as response variables

This may most easily be attained by storing these quantities in COMMON blocks in the main program.

Facilities are provided for varying individual values in the STATE or P arrays; for exploring pairwise interactions between such individual values (either within one array, or between two arrays); for simultaneous modification of a set of values in one or the other array (but not both), either to new arbitrary and undated sets of values, or by multiplying all the values of a common factor and then dividing by the same factor; and finally for exploring the interactions between two such sets of values. The response variables reported may be any quantities in the arrays STATE, STNG, or SUMS, or a weighted sum of a number of values within the same array, and at any arbitrary date during the simulation.

SUBROUTINES:

a) Subroutine SENSIT (IRUN)

The subroutine is called for the first time before the first run of the model (IRUN = 1), but after input operations. It then reads in the specifications for sensitivity analysis, and places in mass storage (unit 0) the arrays of initial values of state variables (STATE) and of parameters (P). At the beginning of each subsequent (IRUN-th) run, the subroutine reads back the arrays STATE and P from drum, then changes values of parameters (or state variables) according to the specifications, and returns control to the main program.

Changes may be introduced by this subroutine into either the set of constants used by the process subroutines, or the initial values of state variables. In what follows, all these quantities which may be modified by this subroutine are referred to as "parameters".

INPUT ORGANIZATION:

The input cards required when the subroutine is first called are as follows:

I. A card in format (16I5), with successive fields of five columns containing the following specifications:

(1) The number of parameters to be changed singly or in interacting pairs;

(2) The number of pairwise interactions to be tested among the parameters included in (1);

(3) The number of response variables to be recorded in each run;

(4) The number of sets of parameters to be changed into arbitrary new sets of values.

(5) The number of sets of parameters to be changed simultaneously by multi-

II. If the figure in I.(1) above was positive a number of sets of cards is read equal in number to that in I (1). Each such set of cards consists of:

(A) A card *n* (16I5) format with the following specifications:

(1) Type of parameter: 1 for a constant used by a process subroutine, 2 for an initial state variable value.

(2) Address of parameter, in the array P or STATE respectively.

(3) The number of values, additional to the "standard" value, to be tested for this parameter.

(B) A card or cards in format (8F10.5) with the alternative values for this parameter, equal in number to the figure at II (A) (3).

III. Cards equal in number to the figure at I(2) above, each containing in format (16I5) two figures only: the sequence numbers of two of the parameters specified in II for which pairwise interactions are required. These numbers are not addresses in the P or STATE arrays, but identify by their ordinal position parameters already defined in II.

IV. Sets of parameters to be changed arbitrarily. For each such set, a batch of cards is read in, including:

(A) One or more cards in format (16I5) with the following specifications in successive fields of five columns.

(1) Parameter type (as in II (A) (1) above)

(2) The number of parameters to be changed

(3) The number of alternative values to be used for each parameter

(4) etc. The addresses (in array STATE or P) of the parameters to be changed.

(B) A card with a single value in format (F10.4), giving a factor by which all parameters in the set are to be multiplied and then divided.

The number of batches of cards (A) and (B) required for this section is equal to the number in I(5).

VI. Pairs of sets of which the interactions are to be tested. The number of cards is equal to that in I(6), and each card contains, in (16I5) format, the numbers of two sets (the ordinal numbers, in sections IV and V above, counted as a single sequence) the interactions between which are required.

VII. Response cards, equal in number to the figure at I(3) above. Each card, in format (16I5), contains the following specifications in successive fields of five columns:

(1) The type of response variable, coded thus:

1. State variable (array STATE)
2. Sum of state variables (array SUMS)
3. Not used
4. Accumulated gains or losses for the system (array STNG)
5. Derived varaible (see below)

(2) The address of the response variable in its array. If the type option specified in VII(1) above is '5', this field is ignored.

(3) The day of simulation (from the beginning of the first year) on which the response is to be recorded.

If option '5' is selected under VII(1), a subrouting DERIVD is called which defines a new response variable to be calculated as a weighted sum of specified values in the system (see below).

OPERATION

Under the control of the subroutine SENSIT, the first time the model is run the original ('standard') values of all entries in the arrays P and STATE are used. Next, each of the n individual parameters to be modified is changed to each of its different values in turn. The number of times the model is run to this point is accordingly

$$M_2 = 1 + \sum_{i=1}^{n_1} m_i$$

where m_i is the number of alternative values to be tested for the i'th parameter. In each of these runs except the first, accordingly, one parameter has a non-standard value. Then, for each interaction between two individual parameters, these two parameters are given in turn all their alternative values specified, in all combinations. The number of model runs required to effect this is

$$M_2 = \sum m_i \; m_j$$

where i and j take only those pair of values specified for interactions tests. In each of these runs, two parameters will have a non-standard value.

The sets of parameters to be modified arbitrarily require a further number of runs

$$M_3 = \sum_{k=1}^{n_2} s_k$$

where n_2 is the number of such sets, and s_k the number of alternative sets of values for the parameters in the k'th set; the sets of parameters to be modified by a common factor will require

$$M_4 = 2 \sum_{k=1}^{n_3} t_k$$

runs, where n_3 is the number of such sets, and t_k the number of powers of the common factor to be used for the k'th set. Finally, for interactions between sets of parameters, the number of runs required will be

$$M_5 = \sum u_i \ u_j$$

where summation is over only those pairs of sets i and j specified for interaction tests, and

$$u_i = s_i$$

if i is among the n_2 sets of the first type, or

$$u_i = s_i$$

if i is among the n_2 sets of the second type.

The total number of runs to be performed is accordingly

$$1 + M_1 + M_2 + M_3 + M_4 + M_5$$

After the modifications in parameters have been made by the subroutine SENSIT, and before control is returned to the main program, the current initial values of the state variables are transferred to mass storage (unit 8).

ARRAY DIMENSIONS

Use of this subroutine is limited by the dimensions allotted to the arrays, which may need to be modified to meet user requirements. Below is a lsit of the arrays used in the subroutine, which may be used as a guide if dimension changes are called for.

ANEW (a,d)
FACTOR(e)
IDAYT (f)
IDPAQ (g)
IDPAR (g)
INSETS (h,i)
INUM (f)
IPA (j)
IPAR (k)
IPB (j)
IQA (g)
IQB (g)
ISTORE (h)
ITYP (f)
IY (b)
JPAR (b)
MADDR (c)
NALTI (i)
NDIF (a)
NDIFFI (i)
NOPAR (b)
P (k)
PDIF (a,d)
VPAQ (g)
VPAR (g)

The dimensions indicated by letters define the maximum values possible for the following quantities:

		FORTRAN equivalent
a	number of parameters to be varied singly or in interacting pairs	NPAR
b	the total number of parameters subject to modification	MPAR
c	total number of runs	NRUN
d	number of alternative values for a parameter to be varied singly or in interacting pairs	———
e	number of sets of parameters to be changed by a common factor	NSETS2
f	number of response variables to be tested	NRESP
g	number of runs with single parameters or interacting parameter pairs varied	MRUN,NRUU
h	number of pairwise interactions between sets of parameters to be tested	INTSET
i	total number of sets of parameters to be changed simultaneously	NSETS3
j	number of pairwise interactions between single parameters to be tested	NINTER
k	the number of parameters in process subroutines	———

B) <u>SUBROUTINE SENOUT (ISW, IDAY, IRUN):</u>

This subroutine controls the storage and output of response variables. It is called in two different cases:

1) ISW = 1

This call is made on every day (IDAY) of every run. The subroutine checks whether any response variable is defined on this day ((IDAY = IDAYT(I)?). and, if it is, stores it in the appropriate address in array RX.

2) ISW = 2

At the end of each run, the values for this run of parameters subject to modification are stored in the appropriate row of array PARXX, the initial values of state variable for the run having first been read back from mass storage (unit 8).

After all runs have been completed, the values of parameters (PARXX) and responses (RX) for all runs are printed in summary table.

3) ARRAY DIMENSIONS:

In this subroutine, as in SENSIT, limitations are imposed by array dimensions, which accordingly may need to be modified by the user. These limitations are indicated below:

PARXX(a,b)

RX(a,b)

VNEW(c)

where the letters define maximum values for the following quantities:

a	the total number of parameters subject to modification	NPARAM
b	the total number of response variables to be tested	NRESP
c	the number of response variables to be calculated by the subrouting DERIVD	_________

C) SUBROUTINE DERIVD (INA, ISV):

This subroutine calculates a response variable as a function of other variables. In the present version a weighted sum is calculated, but it could be replaced (or supplemented) by any other transformations. The subroutine is called in two cases:

1) ISV = 1

Each time a response variable card with the type specified as "5" is read in by the SENSIT subroutine (before the first run), DERIVD is called by SENSIT and reads in specifications for the derived response variable in question. When the subroutine is called with 1 as the first argument, the following input is required:

INPUT ORGANIZATION

A. A specification card in format (16I5) with the following information in the first two fields of five columns:

(1) The type of variable to be used in calculating the new response variable (see under subroutine SENSIT above, section VII(1)).

(2) The number of individual variables to be used in the calculation.

B. One or more cards in format (16I5) are required, identifying the variables to be used in calculation. These cards should contain as many entries as the number in (A) (2) above, and give the addresses of these variables in the array STATE, SUMS, or STNG.

(C) One or more cards in format (8F10.4) giving the multipliers (weights) to be applied to the variables specified in (B).

2) ISV = 2

Each time SENOUT (with ISW = 1) finds that, on the day in question, the value of a response variable of type "5" is to be recorded, DERIVD is called from SENOUT. It then calculates the weighted sum according to the specifications, and returns control to SENOUT which stores it in the appropriate address in the array RX.

3) ARRAY DIMENSIONS:

In this subroutine, as in the others, array dimensions impose restrictions on its use, as follows:

ITYP(a)	NVAR(a)
IVAR(a,b)	WVAR(a,b)

where the limits have the following meanings:

a the number of response variables in whose calculation this subroutine is to be used;

b the maximum number of parameters to be used in the calculation of a response variable.

Level 1 FEATURES OF THE CALLING PROGRAM NEEDED FOR USE WITH SENSITIVITY SUBROUTINES:

The following features are necessary in the main program in connection with sensitivity subroutines:

1) It is suggested that a switch (say, ISENSE) be incorporated in the main program so that the same programs can be used either for sensitivity tests or for simulation without sensitivity testing.

2) Common blocks:

(a) Common blocks are used to allow use of arrays STATE (state variables), SUMS (various sums of those) and STNG (gains or losses to the system) by subroutines SENSIT, SENOUT, DERIVD, and the MAIN program.

(b) Another block makes array P (parameters) common to SENSIT, SENOUT and the subroutines where the parameters are defined and used.

(c) Common block /RESP/ which includes specifications and arrays used in sensitivity analysis is common to SENSIT and SENOUT.

(d) Array VNEW (derived response variables) is common to SENOUT and DERIVD.

3) At the beginning of the first run IRUN is set to 1 and (after input of parameters and state variables), if INSENSE = 1, SENSIT is called to read in sensitivity specifications and store standard values.

4) At the beginning of runs after the first (if ISENSE = 1), dates and other specifications in the main program are re-initialized, and SENSIT is called to specify parameters for the run.

5) On each day of each run, SENOUT is called with ISW = 1 to check and store response variables if needed.

6) At the end of each run, SENOUT (ISW = 2) is called to store parameter and response values, IRUN is incremented by one, and control returns to 3. After the last run, SENOUT proceeds to print the summary table.

The sensitivity subroutines at present are designed to be called from the MAIN program of the multi-purpose model, but are easily accessible for use by any other calling program.

REFERENCES

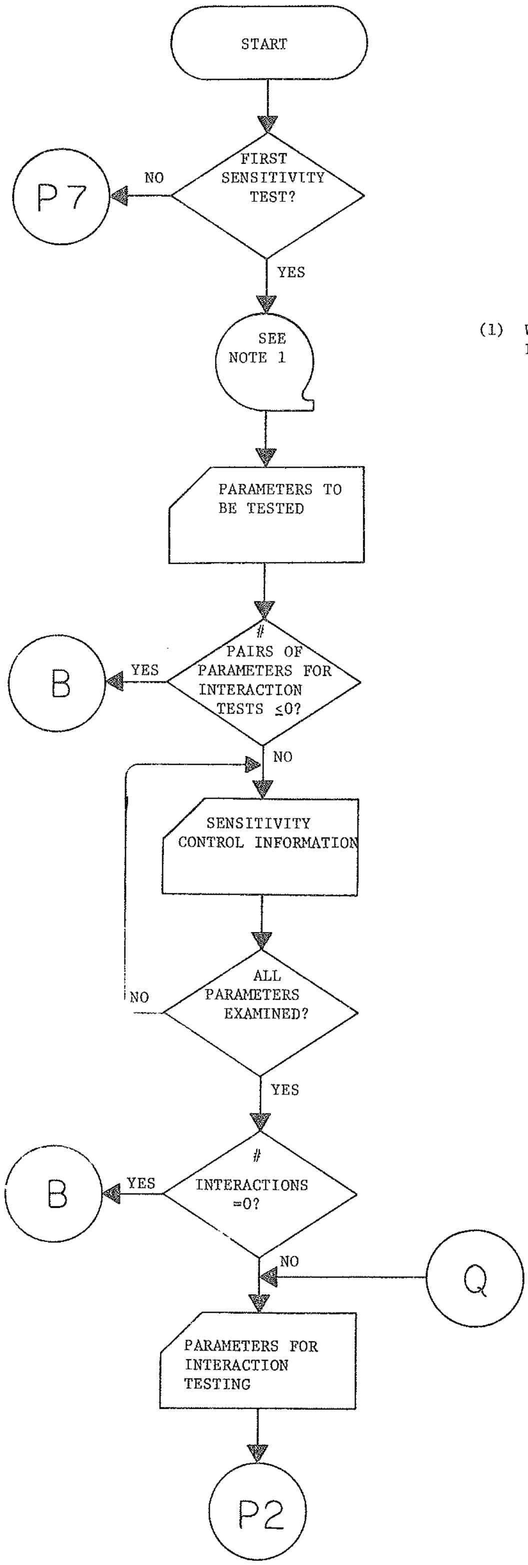
Goodall, D. W. 1969. Simulating the grazing situation. In: "Concepts and models of biomathematics. Simulation techniques and methods" (ed. F. Heinmets) Marcel Dekker, New York. p. 211-236.

Kerlin, T. W. 1967. Sensitivities by the state variable approach. Simulation 337-345.

Noy-Meir, I. (in preparation) Exploratory sensitivity analysis of a simple ecological model: water-limited production.

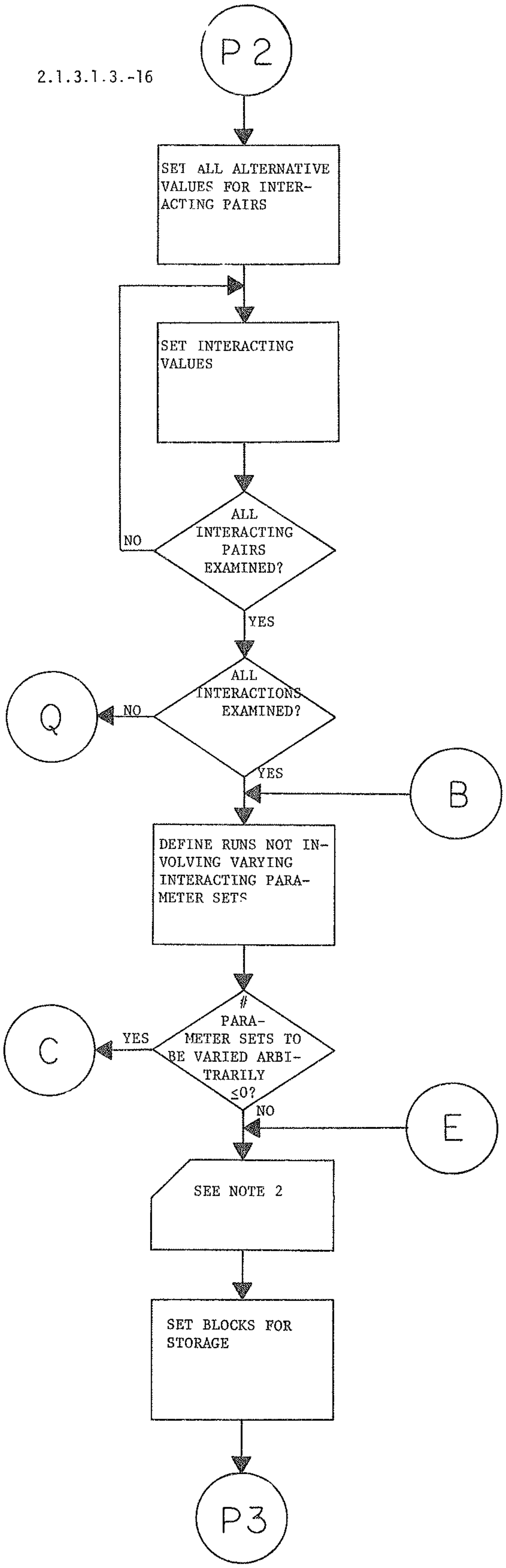
Zusman, P. and Amiad, A. 1966. Simulation -- a tool in farm planning and management under conditions of low and unstable rainfall. Volcani Inst. Agric. Res. Pamph. 102. (Hebrew, with English summary).

FLOW CHART FOR SENSIT



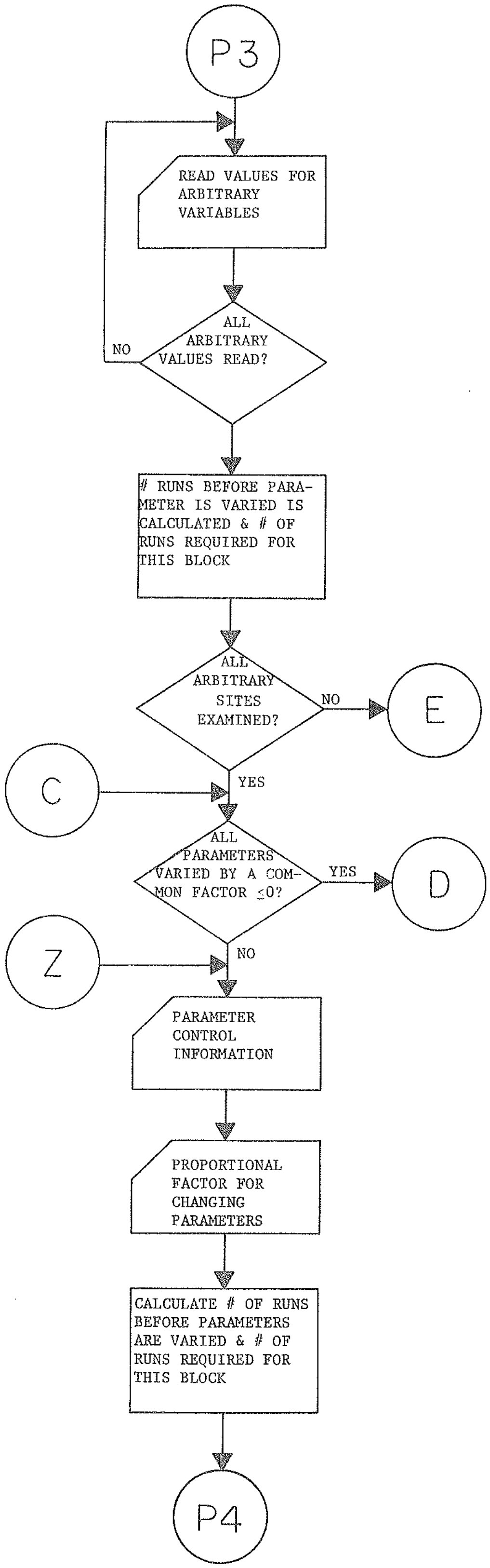
(1) WRITE STATE VARIABLES TO BE EXAMINED FOR SENSITIVITY ANALYSIS LOGICAL (0).

INTERACTIONS BETWEEN PAIRS OF PARAMETERS ARE DETERMINED

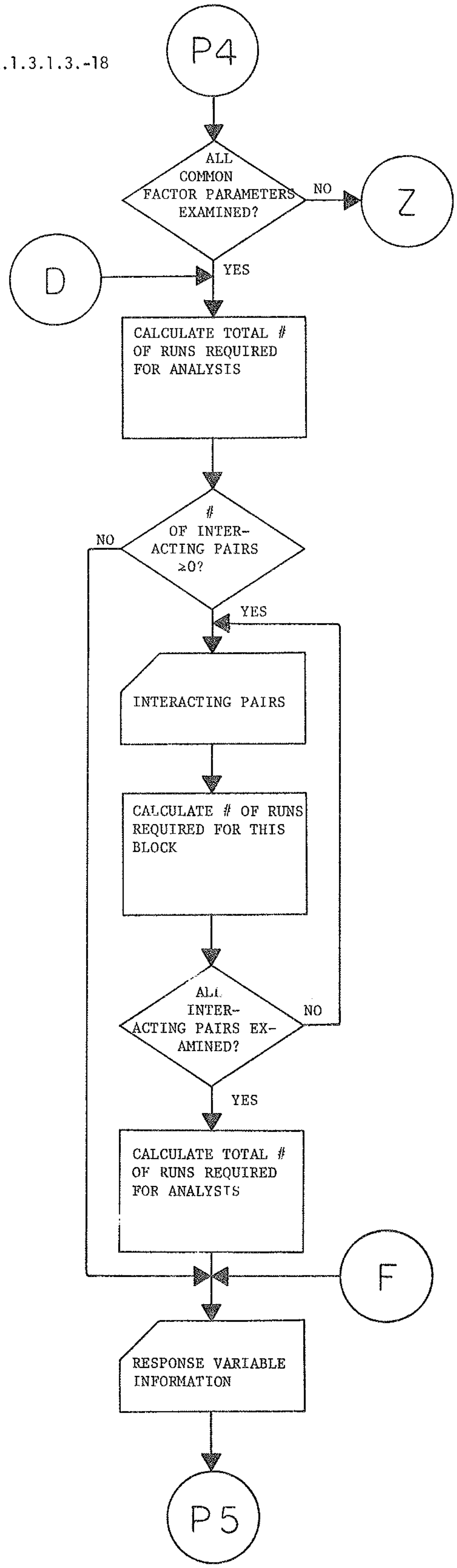


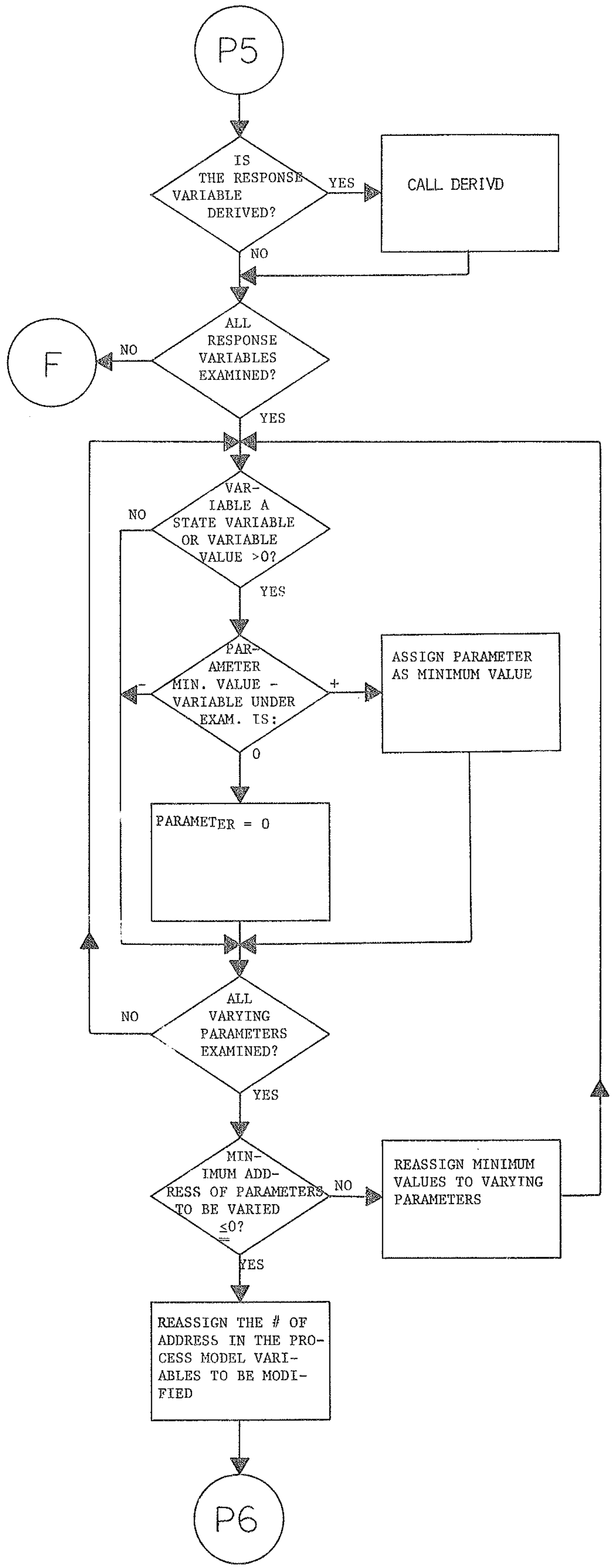
PARAMETERS TO BE CHANGED TO NEW EMPIRICAL SETS ARE DETERMINED AND SET

(2) KIND, # OF PARAMETERS TO BE ALTERED, # OF DIFFERENT VALUES TO BE ASSIGNED TO PARAMETERS AND PARAMETER ADDRESSES

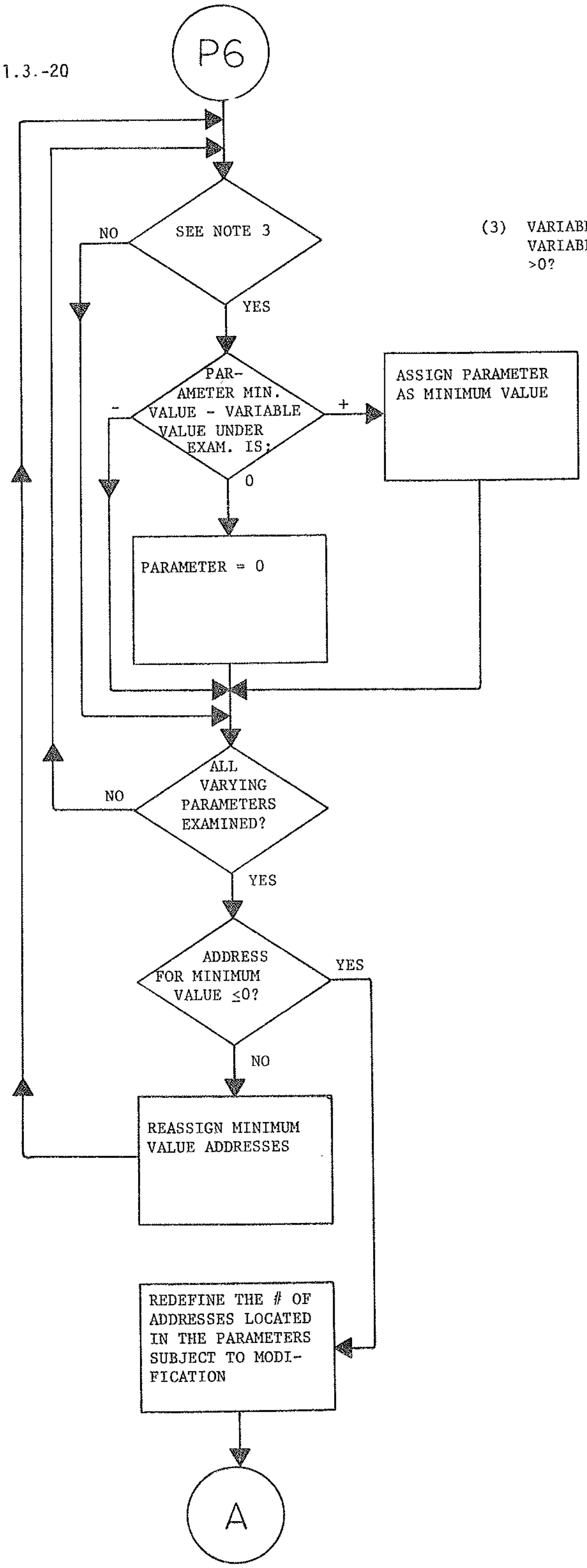

P3
READ VALUES FOR ARBITRARY VARIABLES
ALL ARBITRARY VALUES READ?
NO
RUNS BEFORE PARAMETER IS VARIED IS CALCULATED & # OF RUNS REQUIRED FOR THIS BLOCK
ALL ARBITRARY SITES EXAMINED?
NO
E
C
YES
ALL PARAMETERS VARIED BY A COMMON FACTOR ≤0?
YES
D
Z
NO
PARAMETER CONTROL INFORMATION
PROPORTIONAL FACTOR FOR CHANGING PARAMETERS
CALCULATE # OF RUNS BEFORE PARAMETERS ARE VARIED & # OF RUNS REQUIRED FOR THIS BLOCK
P4

2.1.3.1.3.-18

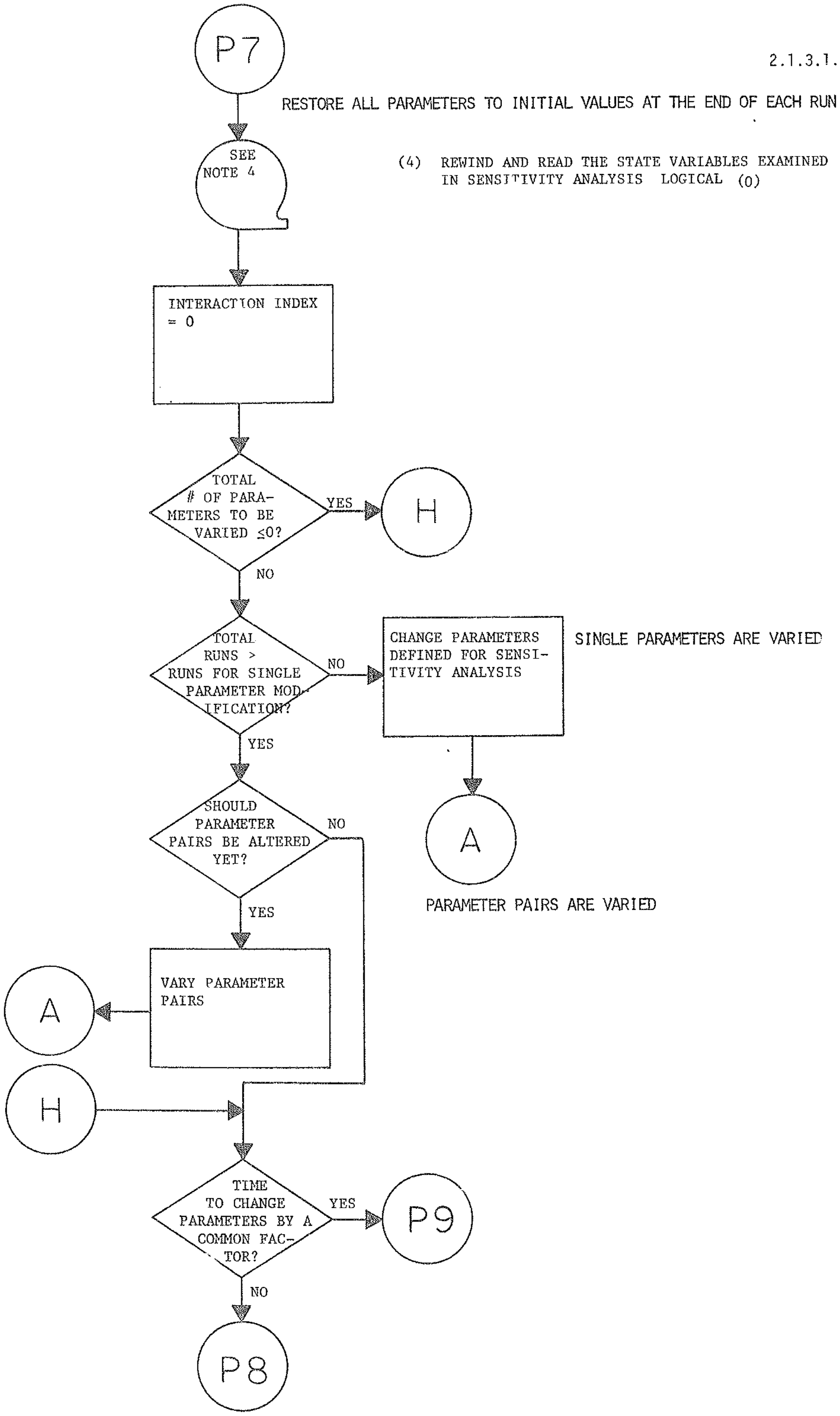

P4
ALL COMMON FACTOR PARAMETERS EXAMINED?
NO
Z
D
YES
CALCULATE TOTAL # OF RUNS REQUIRED FOR ANALYSIS
OF INTERACTING PAIRS ≥0?
NO
YES
INTERACTING PAIRS
CALCULATE # OF RUNS REQUIRED FOR THIS BLOCK
ALL INTERACTING PAIRS EXAMINED?
NO
YES
CALCULATE TOTAL # OF RUNS REQUIRED FOR ANALYSIS
F
RESPONSE VARIABLE INFORMATION
P5

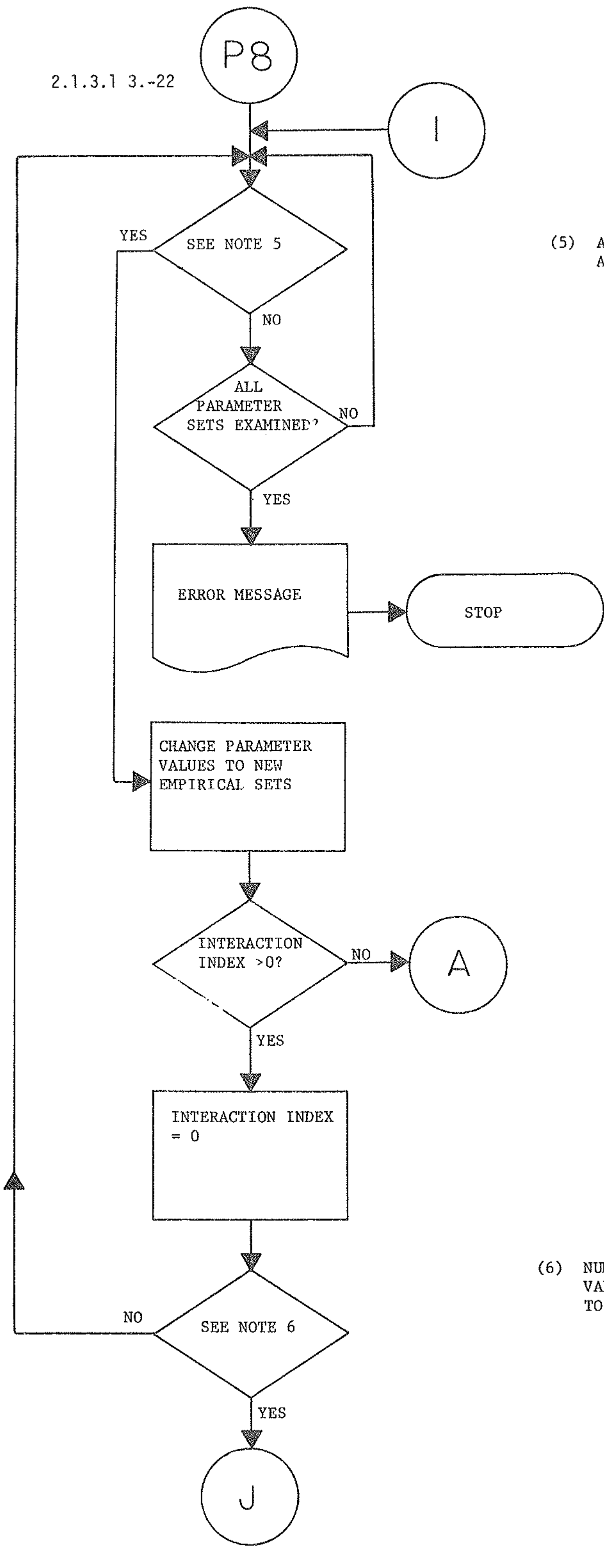

P5
IS THE RESPONSE VARIABLE DERIVED?
YES
CALL DERIVD
NO
ALL RESPONSE VARIABLES EXAMINED?
NO
F
YES
VARIABLE A STATE VARIABLE OR VARIABLE VALUE >0?
NO
YES
PARAMETER MIN. VALUE - VARIABLE UNDER EXAM. IS:
-
+
0
ASSIGN PARAMETER AS MINIMUM VALUE
PARAMETER = 0
ALL VARYING PARAMETERS EXAMINED?
NO
YES
MINIMUM ADDRESS OF PARAMETERS TO BE VARIED ≤0?
NO
REASSIGN MINIMUM VALUES TO VARYING PARAMETERS
YES
REASSIGN THE # OF ADDRESS IN THE PROCESS MODEL VARIABLES TO BE MODIFIED
P6

2.1.3.1.3.-20



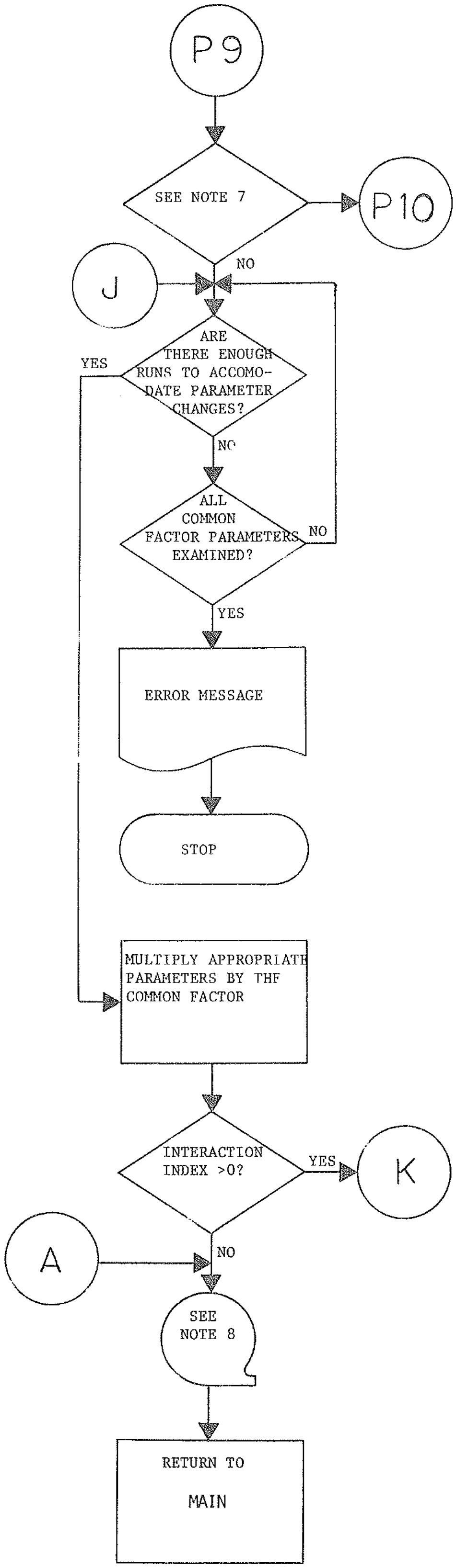
(3) VARIABLE TYPE A SUM OF STATE VARIABLE OR VALUE OF VARIABLE >0?


P7
RESTORE ALL PARAMETERS TO INITIAL VALUES AT THE END OF EACH RUN
SEE NOTE 4
(4) REWIND AND READ THE STATE VARIABLES EXAMINED IN SENSITIVITY ANALYSIS LOGICAL (0)
INTERACTION INDEX = 0
TOTAL # OF PARAMETERS TO BE VARIED ≤0?
YES
H
NO
TOTAL RUNS > RUNS FOR SINGLE PARAMETER MODIFICATION?
NO
CHANGE PARAMETERS DEFINED FOR SENSITIVITY ANALYSIS
SINGLE PARAMETERS ARE VARIED
A
YES
SHOULD PARAMETER PAIRS BE ALTERED YET?
NO
PARAMETER PAIRS ARE VARIED
YES
VARY PARAMETER PAIRS
A
H
TIME TO CHANGE PARAMETERS BY A COMMON FACTOR?
YES
P9
NO
P8



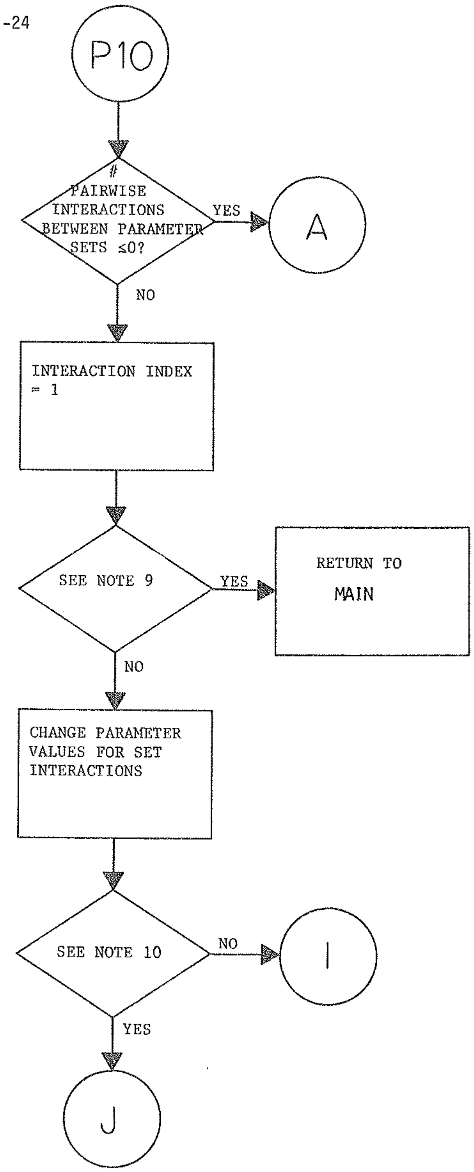
(5) ARE THERE ENOUGH RUNS TO ACCOMODATE ALL PARAMETER CHANGES?

(6) NUMBER PAIRWISE INTERACTING STATE VARIABLES > NUMBER PARAMETER SETS TO BE VARIED ARBITARILY?



(7) TOTAL SENSITIVITY RUNS > TOTAL RUNS APART FROM THOSE INVOLVING SET INTER- ACTION?

(8) REWIND LOGICAL (8) AND WRITE THE INITIAL VALUES OF THE STATE VAR- IABLES BEING TESTED FOR SENSITIVITY.



(9) CURRENT RUN NUMBER ≤ FINAL RUN NUMBER FOR PARAMETER SET INTERACTIONS?

(10) NUMBER OF PAIRWISE INTERACTING STATE VARIABLES > NUMBER OF PARAMETER SETS TO BE VARIED ARBITRARILY?

SENSIT

PROGRAM LISTING

```
      C THIS SUBROUTINE READS IN THE SPECIFICATIONS OF ANY SENSITIVITY TESTS   SENS0010
      C TO BE PERFORMED, AND THEN MODIFIES PROCESS PARAMETERS OR THE INITIAL  SENS0020
      C VALUES OF STATE VARIABLES AS SPECIFIED (IN SUBSEQUENT COMMENTS        SENS0030
      C REFERRED TO COLLECTIVELY AS 'PARAMETERS') AT THE BEGINNING OF EACH    SENS0040
      C REPEATED RUN.                                                          SENS0050
                                                                               SENS0060
      C THE FOLLOWING ARE DEFINITIONS OF VARIABLE NAMES USED IN THIS          SENS0070
      C SUBROUTINE AND NOT IN THE MAIN PROGRAMME.  VARIABLES USED ONLY FOR    SENS0080
      C TEMPORARY PURPOSES, OR WITH DIFFERENT MEANINGS AT DIFFERENT TIMES,    SENS0090
      C ARE IN THE MAIN OMITTED.                                               SENS0100
                                                                               SENS0110
                                                                               SENS0120
      C ANEW(I,J)         THE J'TH ALTERNATIVE VALUE FOR THE I'TH PARAMETER IN SENS0130
      C                   SETS TO BE VARIED ARBITRARILY                        SENS0140
      C FACTOR(I)         MULTIPLYING FACTOR FOR THE I'TH PARAMETER SET        SENS0150
      C IDA               TEMPORARY STORAGE OF PARAMETER ADDRESS               SENS0160
      C IDAYT(I)          THE TIME AT WHICH THE I'TH RESPONSE VARIABLE IS TO   SENS0170
      C                   BE CALCULATED                                        SENS0180
      C IDB               TEMPORARY STORAGE OF PARAMETER ADDRESS               SENS0190
      C IDPAQ(I)          THE ADDRESS OF THE SECOND PARAMETER TO BE TESTED     SENS0200
      C                   IN INTERACTION IN THE I'TH RUN                       SENS0210
      C IDPAP(I)          THE ADDRESS OF THE FIRST PARAMETER TO BE TESTED IN   SENS0220
      C                   INTERACTION IN THE I'TH RUN                          SENS0230
      C INSETS(I,J)       THE SEQUENCE NUMBER OF THE J'TH PARAMETER SET IN THE SENS0240
      C                   I'TH INTERACTION PAIR                                SENS0250
      C INTSET            NUMBER OF PAIR-WISE INTERACTIONS BETWEEN SETS OF     SENS0260
      C                   PARAMETERS TO BE TESTED                              SENS0270
      C INTXXX            INDEX TO DISTINGUISH INTERACTION OPERATIONS ON SETS  SENS0280
      C INUM(I)           THE ADDRESS OF THE I'TH RESPONSE VARIABLE            SENS0290
      C IP                ADDRESS (IN P OR STATE) OF A SINGLE-VALUE PARAMETER  SENS0300
      C                   TO BE VARIED                                         SENS0310
      C IPA(I)            SEQUENCE NUMBER OF FIRST PARAMETER OF THE I'TH PAIR  SENS0320
      C                   FOR INTERACTION TESTING                              SENS0330
      C IPAR(I)           THE ADDRESS OF THE I'TH PARAMETER SUBJECT TO         SENS0340
      C                   MODIFICATION                                         SENS0350
      C IPB(I)            SEQUENCE NUMBER OF SECOND PARAMETER OF THE I'TH PAIR SENS0360
      C                   FOR INTERACTION TESTING                              SENS0370
      C IQ                TEMPORARY STORAGE OF PARAMETER SEQUENCE NUMBER       SENS0380
      C IQA(I)            THE SERIAL NUMBER OF A SINGLE-VALUE PARAMETER TO BE  SENS0390
      C                   VARIED IN THE I'TH RUN                               SENS0400
      C IQB(I)            THE SERIAL NUMBER OF A SECOND SINGLE-VALUE PARAMETER SENS0410
      C                   TO BE VARIED IN THE I'TH RUN                         SENS0420
      C IQQ               TEMPORARY STORAGE OF PARAMETER SEQUENCE NUMBER       SENS0430
      C ISTORE(I)         THE SEQUENCE NUMBER OF THE FINAL RUN FOR THE I'TH    SENS0440
      C                   PARAMETER SET INTERACTION                            SENS0450
      C ITY               PARAMETER TYPE                                       SENS0460
      C ITYP(I)           THE TYPE (STATE VARIABLE SUM OR CALCULATED FUNCTION) SENS0470
      C                   OF THE I'TH RESPONSE VARIABLE                        SENS0480
      C IY(I)             THE TYPE (PROCESS-MODEL CONSTANT, OR STATE VARIABLE) SENS0490
      C                   OF THE I'TH PARAMETER UNDERGOING MODIFICATION        SENS0500
      C IYY               TEMPORARY STORAGE FOR PARAMETER TYPE                 SENS0510
      C JPAR(I)           TEMPORARY STORAGE OF IPAR (I) FOR ORDERING PURPOSE   SENS0520
      C MADDR(I)          NUMBER OF RUNS BEFORE THE I'TH SET OF PARAMETERS     SENS0530
      C                   VARIED                                               SENS0540
      C MAPAR             NUMBER OF SINGLE-VALUE PARAMETERS, PLUS THOSE IN     SENS0550
      C                   SETS TO BE VARIED ARBITRARILY                        SENS0560
```

```
C MARUN      SEQUENCE NUMBER OF LAST RUN BEFORE SETS TO BE VARIED   SENS0570
C            BY A COMMON FACTOR                                     SENS0580
C MINAD      TEMPORARY STORAGE OF ADDRESS OF MINIMUM VALUE IN IPAR  SENS0590
C MINPAR     TEMPORARY STORAGE OF MINIMUM VALUE IN IPAR             SENS0600
C MPAR       TOTAL NUMBER OF PARAMETERS TO BE VARIED                SENS0610
C NALT       NUMBER OF PARAMETERS IN A SET SUBJECT TO ALTERATION    SENS0620
C NALT1(I)   STORED VALUE OF NALT FOR THE I'TH SET                  SENS0630
C NDT        NUMBER OF VALUES OF A PARAMETER TO BE USED IN          SENS0640
C            ADDITION TO THE ORIGINAL VALUE                         SENS0650
C NDIF(I)    THE NUMBER OF ALTERNATIVE VALUES OF THE I'TH PARAMETER SENS0660
C            TO BE MODIFIED                                         SENS0670
C NDIFF      NUMBER OF ALTERNATIVE SETS OF VALUESFOR A SET OF       SENS0680
C            PARAMETERS VARIED ARBITRARILY, OR NUMBER OF POWERS OF  SENS0690
C            THE MULTIPLYING FACTOR TO BE USED                      SENS0700
C NDIFF1(I)  STORED VALUE OF NDIFF FOR THE I'TH SET                 SENS0710
C NINTER     NUMBER OF PAIRS OF SINGLE-VALUE PARAMETERS FOR         SENS0720
C            INTERACTION TESTS                                      SENS0730
C NNA        NUMBER OF ALTERNATIVE VALUES FOR THE FIRST OF A PAIR   SENS0740
C            OF SINGLE-VALUE PARAMETERS SUBJECT TO INTERACTIVE      SENS0750
C            TESTING                                                SENS0760
C NNB        NUMBER OF ALTERNATIVE VALUES FOR THE SECOND OF A PAIR  SENS0770
C            OF SINGLE-VALUE PARAMETERS SUBJECT TO INTERACTIVE      SENS0780
C            TESTING                                                SENS0790
C NOPAR(I)   ORDERED VALUES FROM IPAR                               SENS0800
C NPAR       NUMBER OF SINGLE-VALUE PARAMETERS TO BE VARIED         SENS0810
C NPARAM     NUMBER OF ADDRESSES OF PROCESS-MODEL CONSTANTS IN      SENS0820
C            NOPAR                                                  SENS0830
C NPARA1     NPARAM + 1 (NEEDED FOR DO-LOOP OPERATIONS)             SENS0840
C NRA        RUN NUMBER DURING SINGLE-PARAMETER MODIFICATION        SENS0850
C NRESP      NUMBER OF RESPONSE VARIABLES TO BE TESTED              SENS0860
C NRU        RUN NUMBER DURING TESTING OF SINGLE-PARAMETER          SENS0870
C            INTERACTIONS                                           SENS0880
C NRUN       TOTAL NUMBER OF RUNS                                   SENS0890
C NRUNN      TOTAL NUMBER OF RUNS, APART FROM THOSE INVOLVING       SENS0900
C            SET INTERACTION                                        SENS0910
C NRUU       TOTAL NUMBER OF RUNS NOT VARYING SETS OF PARAMETERS    SENS0920
C NSETS1     NUMBER OF SETS OF PARAMETERS TO BE VARIED ARBITRARILY  SENS0930
C NSETS2     NUMBER OF SETS OF PARAMETERS TO BE VARIED BY A COMMON  SENS0940
C            FACTOR                                                 SENS0950
C NSETS3     TOTAL NUMBER OF SETS OF PARAMETERS TO BE VARIED
C NSETS4     SEQUENCE NUMBER OF FIRST SET OF PARAMETERS TO BE
C            VARIED BY A COMMON FACTOR
C NSTATE     NUMBER OF STATE-VARIABLE ADDRESSES IN NOPAR            SENS0990
C NUM        NUMBER OF RUNS REQUIRED FOR A SET INTERACTION          SENS1000
C P(I)       THE I'TH PARAMETER IN PROCESS SUB-ROUTINES             SENS1010
C PDIF(I,J)  THE J'TH ALTERNATIVE VALUE TO BE TESTED FOR THE I'TH   SENS1020
C            SINGLE-VALUE PARAMETER SUBJECT TO MODIFICATION         SENS1030
C VPAQ(I)    THE VALUE FOR THE I'TH RUN OF THE PARAMETER OF WHICH   SENS1040
C            THE ADDRESS IS GIVEN IN IOPAQ(I) FOR THE I'TH RUN      SENS1050
C VPAR(I)    THE VALUE FOR THE I'TH RUN OF THE PARAMETER OF WHICH   SENS1060
C            THE ADDRESS IS GIVEN IN IOPAR(I)                       SENS1070

      SUBROUTINE SENSIT (IRUN)                                      SENS1080
      COMMON /PARAM/ P(1000)                                        SENS1090
      DIMENSION NDIF(50),PDIF(50,10),IPAR(100),IPA(10),IPB(10)      SENS1100
      DIMENSION IOPAR(300), IOPAQ(300),                             SENS1110
     1 VPAR(100), VPAQ(100), IQA(300), IQB(300)                     SENS1120
```

```
      DIMENSION  Y(50), NSETS(10,2),ITSTOP(10)                           SENS1130
      DIMENSION  ANOR(10), NALT1(10), NOTEF1(10), TE1(5,10)              SENS1140
     1 ,ANEW(100,5), FACTOR(10)                                          SENS1150
      COMMON/STAT/STATE(1670)                                            SENS1160
      COMMON/TOTAL/SUMS(1450)                                            SENS1170
      COMMON/ACC/ ENO(10)                                                SENS1180
C------------------------------------------------------------------------SENS1190
C     THE COMMON BLOCK /ESN / CONTAINS INFORMATION ON RESPONSES FOR      SENS1200
C     COMMUNICATION WITH THE SUBROUTINE SENOUT.                          SENS1210
C------------------------------------------------------------------------SENS1220
      COMMON/ES / ITYP(20), INUM(20), IPAYT(20), NRESP,                  SENS1230
     1 NPAR, NRUN, NCPAR(100),  PARAM, NPARA1, NSTATE                    SENS1240
      DIMENSION JPAR(100)                                                SENS1250
      NRU=NRUN                                                           SENS1360
      IF(IRUN.GT.1)GO TO 300                                             SENS1370
C------------------------------------------------------------------------SENS1380
C     AT THE BEGINNING OF THE FIRST RUN, INITIAL VALUES OF THE           SENS1390
C     PARAMETERS ARE STORED.                                             SENS1400
C------------------------------------------------------------------------SENS1410
      WRITE(8)STATE,P                                                    SENS1420
C------------------------------------------------------------------------SENS1430
C     THE NUMBERS OF SENSITIVITY TESTS OF DIFFERENT KINDS TO BE          SENS1440
C     PERFORMED ARE SPECIFIED.                                           SENS1450
C------------------------------------------------------------------------SENS1460
      READ (5,40) NPAR,NINTER,NRESP,NSETS1, NSETS2, INTSET               SENS1470
   40 FORMAT(10I5)                                                       SENS1480
C------------------------------------------------------------------------SENS1490
C     PARTICULARS OF THE PARAMETER VARIATIONS TO BE PERFROMED ARE        SENS1500
C     READ IN.                                                           SENS1510
C------------------------------------------------------------------------SENS1520
      NPA=1                                                              SENS1530
      NRU = 1                                                            SENS1540
      IF (NPAR.LE.0) GO TO 110                                           SENS1550
                                                                         SENS1560
C..........SINGLE PARAMETERS                                             SENS1570
      DO 80 I=1,NPAR                                                     SENS1580
      READ (5,40) ITY, IP, NDI                                           SENS1590
      READ (5,50) (PDIF(I,J), J=1,NDI)                                   SENS1600
   50 FORMAT (8F10.5)                                                    SENS1610
      IPAR(I)=IP                                                         SENS1620
      NOTEF(I)=NDI                                                       SENS1630
   60 CONTINUE                                                           SENS1640
      DO 70 J=1,NDI                                                      SENS1650
      NPA=NPA+1                                                          SENS1660
      IPAR(NPA)                                                          SENS1670
      PAR(NPA)      (I,J)                                                SENS1680
      IPA(NPA)=IP                                                        SENS1690
      TY(I)=ITY                                                          SENS1700
   70 CONTINUE                                                           SENS1710
   80 CONTINUE                                                           SENS1720
      NCI=0                                                              SENS1730
      NRU =NPA                                                           SENS1740
      IF (NINTER.EQ.0) GO TO 110                                         SENS1750
                                                                         SENS1760
C..........INTERACTIONS BETWEEN PAIRS OF PARAMETERS.                     SENS1770
      DO 100 I=1,NINTER                                                  SENS1780
      READ (5,40) IPA(I),IPB(I)                                          SENS1790
```

```
      IA=IPAR(I)                                                       SENS1800
      IB=IPAR(I)                                                       SENS1810
      NNA=NDIFF(IA)                                                    SENS1820
      NNB=NDIFF(IB)                                                    SENS1830
      DO 90 J=1,NNA                                                    SENS1840
      DO 90 K=1,NNB                                                    SENS1850
      NRI=NRI+1                                                        SENS1860
      NRU=NRU+1                                                        SENS1870
      IPPAR(NRU)=IPAR(IA)                                              SENS1880
      VPAR(NRU)=PDIF(IA,J)                                             SENS1890
      IPPA2(NRU)=IPAR(IB)                                              SENS1900
      VPA2(NRU)=PDIF(IB,K)                                             SENS1910
      ISA(NRU)=IA                                                      SENS1920
      ISB(NRU)=IB                                                      SENS1930
   90 CONTINUE                                                         SENS1940
  100 CONTINUE                                                         SENS1950
  110 CONTINUE                                                         SENS1960
      NRUU = NRU                                                       SENS1970
      NRUN = NRU                                                       SENS1980
      MPAR = NPAR                                                      SENS1990
      MRUN = NRUN                                                      SENS2000
      MADDR(1) = NRUN                                                  SENS2010
      NSETS3 = NSETS2 + NSETS1                                         SENS2020
      NSETS4 = NSETS1 + 1                                              SENS2030
      IF (NSETS1.LE.0) GO TO 160                                       SENS2040
                                                                       SENS2050
C..........SETS OF PARAMETERS TO BE CHANGED TO NEW EMPIRICAL SETS OF   SENS2060
C..........VALUES.                                                     SENS2070
      K = 0                                                            SENS2080
      K1 = MPAR                                                        SENS2090
      K2 = 0                                                           SENS2100
      DO 150 I = 1, NSETS1                                             SENS2110
      I1 = I + MPAR                                                    SENS2120
      READ (5,40)  ITY, NALT,NDIFF, (IP1(I,J), J = 1, NALT)            SENS2130
      K2 = K2 + NALT                                                   SENS2140
      MPAR = MPAR + NALT                                               SENS2150
      DO 120 J = 1, NALT                                               SENS2160
      K1 = K1 + 1                                                      SENS2170
      K2 = IP1(I,J)                                                    SENS2180
      IPAR(K1) = K2                                                    SENS2190
      TY(K1) = ITY                                                     SENS2200
  120 CONTINUE                                                         SENS2210
      NDIFF1(I)= NDIFF                                                 SENS2220
      NALT1(I) = NALT                                                  SENS2230
  130 K = K + 1                                                        SENS2240
      READ (5,140) (ANEW(K,J), J=1,NDIFF)                              SENS2250
  140 FORMAT (8E10.4)                                                  SENS2260
      IF (K.LT.(K1-MPAR)) GO TO 130                                    SENS2270
      MADDR(I+1) = NDIFF + MADDR(I)                                    SENS2280
      NRUN = NRUN + NDIFF                                              SENS2290
  150 CONTINUE                                                         SENS2300
  160 MARUN = NRUN                                                     SENS2310
      MAPAR = MPAR                                                     SENS2320
      IF (NSETS2.LE.0) GO TO 200                                       SENS2330
                                                                       SENS2340
C..........SETS OF PARAMETERS TO BE MULTIPLIED BY A COMMON FACTOR.     SENS2350
      K1 = MAPAR                                                       SENS2360
```

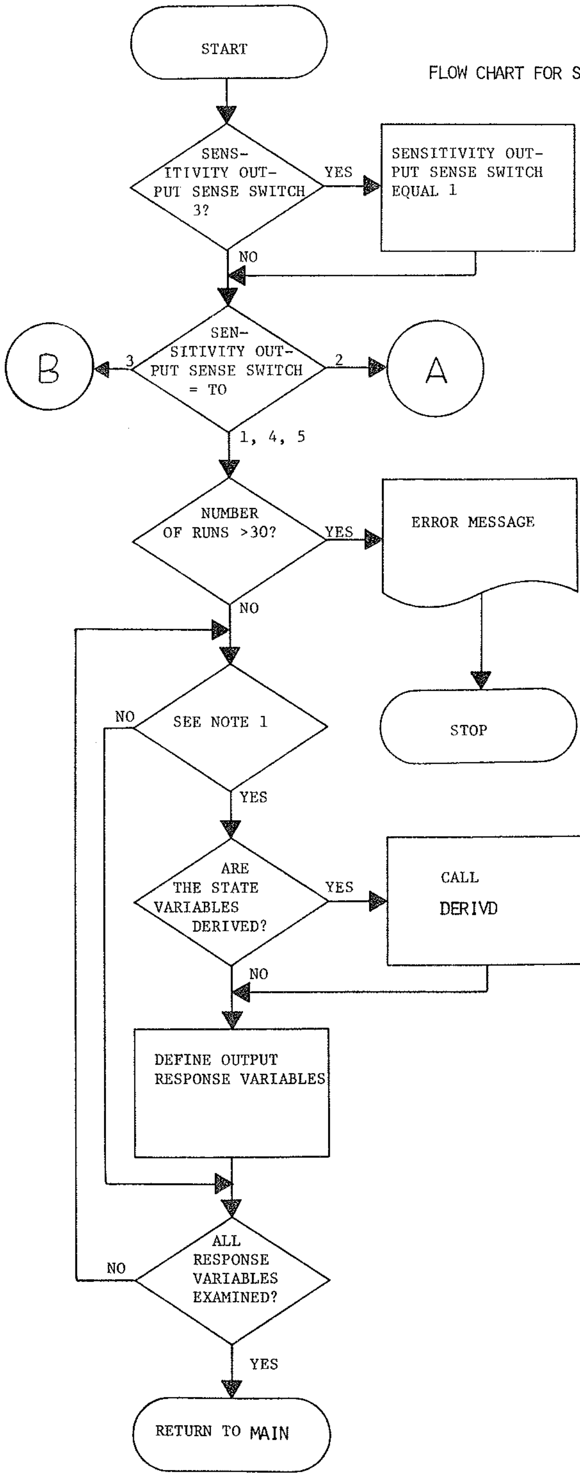
```
      DO 190 I = NSETS4, NSETS3                                         SENS2370
      READ (5,40) ITY, NALT, NDIFF, (IP1(I,J), J=1,NALT)                SENS2380
      MPAR = MPAR + NALT                                                SENS2390
      DO 170 J = 1, NALT                                                SENS2400
      K1 = K1 + 1                                                       SENS2410
      K2 = IP1(I,J)                                                     SENS2420
      IPAR(K1) = K2                                                     SENS2430
      TY(K1) = ITY                                                      SENS2440
  170 CONTINUE                                                          SENS2450
      NALT1(I) = NALT                                                   SENS2460
      NDIFF1(I) = NDIFF                                                 SENS2470
      READ (5,140) FACTOR (I)                                           SENS2480
      IF (FACTOR(I).GT.0.) GO TO 180                                    SENS2490
      NRUN = NRUN + 1                                                   SENS2500
      GO TO 190                                                         SENS2510
  180 NRUN = NRUN + 2*NDIFF                                             SENS2520
  190 MADDR(I+1) = NRUN                                                 SENS2530
  200 NRUNN = NRUN                                                      SENS2540
                                                                        SENS2550
C.........INTERACTIONS BETWEEN SETS OF PARAMETERS.                      SENS2560
      IF (INTSET.LE.0) GO TO 220                                        SENS2570
      DO 210 I = 1, INTSET                                              SENS2580
      READ (5,40) J,K                                                   SENS2590
      NUM = NDIFF1(J) * NDIFF1(K)                                       SENS2600
      IF (J.GT.NSETS1) NUM = NUM * 2                                    SENS2610
      IF (K.GT.NSETS1) NUM = NUM * 2                                    SENS2620
      NPUN = NRUN + NUM                                                 SENS2630
      ISTORE(I) = NRUN                                                  SENS2640
      INSETS(I,1) = J                                                   SENS2650
  210 INSETS(I,2) = K                                                   SENS2660
      I91 = 1                                                           SENS2670
C-----------------------------------------------------------------------SENS2680
C     RESPONSE VARIABLES ARE IDENTIFIED.                                SENS2690
C-----------------------------------------------------------------------SENS2700
  220 CONTINUE                                                          SENS2710
      DO 230 I=1,NRESP                                                  SENS2720
      READ (5,40) ITYP(I),INUM(I),IDAYT(I)                              SENS2730
      INU=INUM(I)                                                       SENS2740
      ISV=1                                                             SENS2750
      IF(ITYP(I).EQ.5)CALL DERIVD(INU,ISV)                              SENS2760
  230 CONTINUE                                                          SENS2770
      DO 240 I = 1, MPAR                                                SENS2780
  240 JPAR(I) = IPAR(I)                                                 SENS2790
      K = 0                                                             SENS2800
  250 MINADD = 0                                                        SENS2810
      MTNPAR = 100000                                                   SENS2820
      MTNPAS = MINPAR                                                   SENS2830
      DO 280 I = 1, MPAR                                                SENS2840
      IF ((IY(I).NE.1).OR.(JPAR(I).LE.0)) GO TO 280                     SENS2850
      IF (MINPAS - JPAR(I))   280, 260, 270                             SENS2860
  260 JPAR(I) = 0                                                       SENS2870
      GO TO 280                                                         SENS2880
  270 MTNPAS = JPAR(I)                                                  SENS2890
      MTNADD = I                                                        SENS2900
  280 CONTINUE                                                          SENS2910
      IF (MINADD.LE.0) GO TO 310                                        SENS2920
  290 K = K + 1                                                         SENS2930
```

```
      [illegible]
  3?? [illegible]
          [illegible]
  31? [illegible]
          [illegible]
  32? MINADD = 0
      MINPAC = [illegible]
      MINPAC = MINPAR
      DO 350 I = 1, NPAR
      IF ((IY(I).NE.0).OR.(JPAR(I).LE.0)) GO TO 350
      IF (MINPAC - JPAR(I)) 350, 330, 340
  330 JPAR(I) = 0
      GO TO 350
  340 MINPA = JPAR(I)
      MINADD = I
  350 CONTINUE
      IF (MINADD.LE.0) GO TO 380
  360 K = K + 1
      NOPAR(K) = MINPAC
  370 JPAR(MINADD) = 0
      GO TO 320
  380 NSTATE = K
      GO TO 730
C-----------------------------------------------------------------------
C     AT THE BEGINNING OF EACH SUCCESSIVE RUN, THE INITIAL VALUES OF
C     ALL PARAMETERS ARE RESTORED, AND THEN MODIFIED.
C-----------------------------------------------------------------------
  390 REWIND 8
      READ(8) STATE,P
      JRUN = IRUN
      INTXYY = 0
      IF (NPAR.LE.0) GO TO 480
      IF(IRUN.GT.N[illegible])GOTO420

C...........SINGLE PARAMETERS ARE CHANGED.
      IRA=IDPAR(IRUN)
      ID = IRA(IRUN)
      IYY = IY(ID)
      GO TO(400,410), IYY
  400 P(IDA) = VPAR(IRUN)
      GO TO 730
  410 STATE(IDA) = VPAR(IRUN)
      GO TO 730
  420 IF (IRUN.GT.[illegible]) GO TO 48[illegible]

C...........[illegible] PARAMETERS ARE CHANGED
      [illegible]
      [illegible] UN)
      [illegible] UN)
      [illegible]
      IYY = IY( )
      GO TO (4[illegible],440), IYY
  430 P(IDA) = VPAR(IRUN)
      GO TO [illegible]
  440 STATE([illegible]) = VPAR([illegible])
  450 IYY = IY(I[illegible])
      GO TO (460,470), IYY
```

```
      460 P(IDE)=VPAS(IPUN)                                             SENS351C
          GO TO 730                                                     SENS352C
      470 STATE(IDP) = VPAS(IPUN)                                       SENS353C
          GO TO 730                                                     SENS354C
      480 IF (IRUN.GT.MARUN) GO TO 570                                  SENS355C
      490 K = JRUN                                                      SENS356C
      500 K1 = NPAR                                                     SENS357C
                                                                        SENS358C
C..........SETS OF PARAMETERS ARE CHANGED TO NEW EMPIRICAL SETS.        SENS359C
          DO 510 I = 1, NSETS1                                          SENS360C
          IF (K.LE.MADDP(I+1)) GO TO 530                                SENS361C
          K = K - NDIFF1(I)                                             SENS362C
          K1 = K1 + NALT1(I)                                            SENS363C
      510 CONTINUE                                                      SENS364C
          WRITE (6,520)                                                 SENS365C
      520 FORMAT (' ERROR IN NSETS1')                                   SENS366C
          STOP                                                          SENS367C
      530 K2 = NALT1(I)                                                 SENS368C
          K = K - MRUN                                                  SENS369C
          DO 560 J1 = 1, K2                                             SENS370C
          K1 = K1 + 1                                                   SENS371C
          IP = IPAR(K1)                                                 SENS372C
          K3 = K1 - NPAR                                                SENS373C
          B = ANEW(K3,K)                                                SENS374C
          IYY = IY(K1)                                                  SENS375C
          GO TO (540,550), IYY                                          SENS376C
      540 P(IP) = B                                                     SENS377C
          GO TO 560                                                     SENS378C
      550 STATE(IP) = B                                                 SENS379C
      560 CONTINUE                                                      SENS380C
          IF (INTXYY) 730,730,720                                       SENS381C
      570 IF (IRUN.GT.NRUNN) GO TO 670                                  SENS382C
                                                                        SENS383C
C..........SETS OF PARAMETERS ARE MULTIPLIED BY A COMMON FACTOR.        SENS384C
      580 K = JRUN - MARUN                                              SENS385C
      590 K1 = MAPAR                                                    SENS386C
          DO 600 J = NSETS4, NSETS3                                     SENS387C
          IDIFF = MADDR(J+1) - MADDR(J)                                 SENS388C
          IF (K.LE.IDIFF) GO TO 620                                     SENS389C
          K = K - IDIFF                                                 SENS390C
          K1 = K1 + NALT1(J)                                            SENS391C
      600 CONTINUE                                                      SENS392C
          WRITE (6,610)                                                 SENS393C
      610 FORMAT (' ERROR IN NSETS2')                                   SENS394C
          STOP                                                          SENS395C
      620 A = FACTOR(J)                                                 SENS396C
          J11 = J + 1                                                   SENS397C
          IF (IDIFF.EQ.1) GO TO 630                                     SENS398C
          I22 = (K+1)/2                                                 SENS399C
          I21 = I22*2 - K                                               SENS400C
          IF (I21.LE.0) I22 = -I22                                      SENS401C
          A = A**I22                                                    SENS402C
          K2 = NALT1(J)                                                 SENS403C
      630 DO 660 I = 1, K2                                              SENS404C
          K1 = K1 + 1                                                   SENS405C
          IYY = IY(K1)                                                  SENS406C
          IP = IP1(J,I)                                                 SENS407C
```

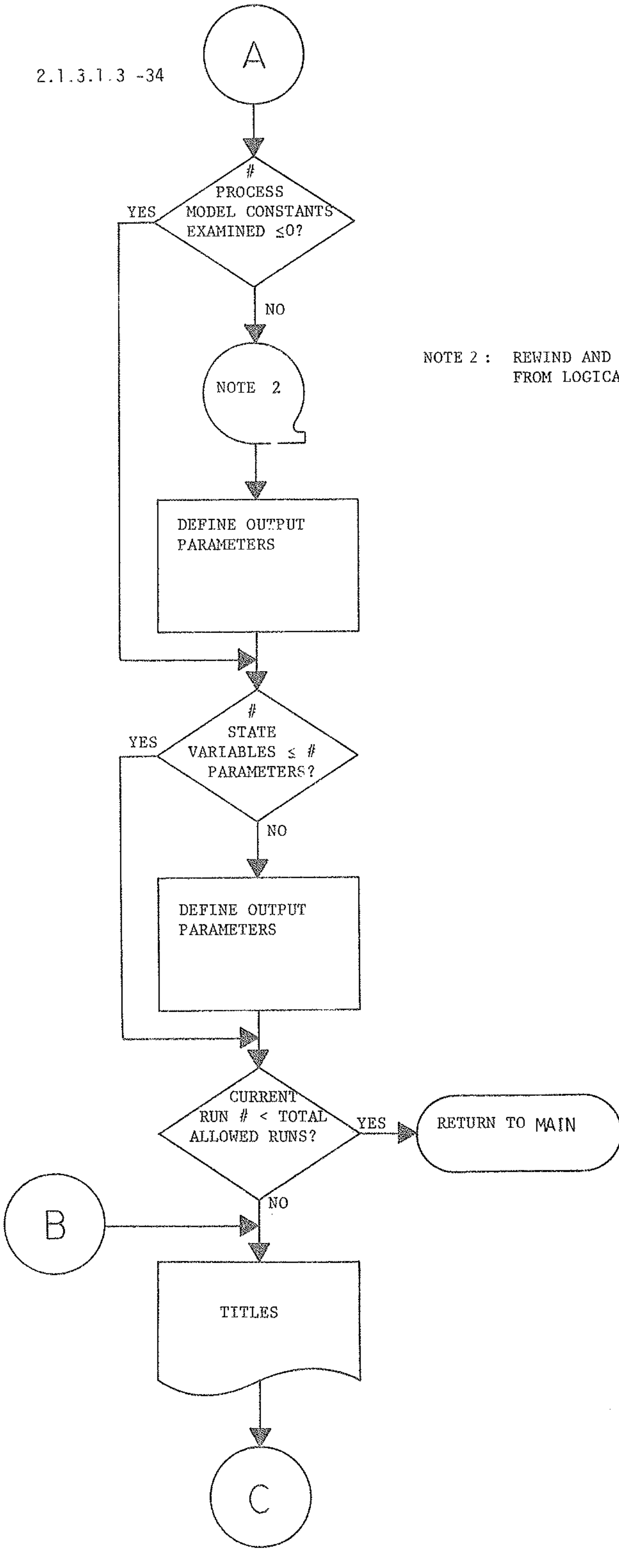
```
          (F4 , 5 ), XXY                                                 SENS4080
   34     (IF) = F(I ) * A                                               SENS4090
            TO 60                                                        SENS4100
   5      STATE (I9) = STATE(II) * A                                     SENS4110
   660 CONTINUE                                                          SENS4120
       IF (INTYXY) 700,700,720                                           SENS4130
   670 CONTINUE                                                          SENS4140
   680 IF (INTSET.LE.0) GO TO 730                                        SENS4150
       INTXXX = 1                                                        SENS4160
   690 IF (TRUN.LT.ISTORE(I91)) GO TO 700                                SENS4170
       I91 = I91 + 1                                                     SENS4180
       IF (I91.LE.INTSET) GO TO 690                                      SENS4190
       RETURN                                                            SENS4200
                                                                         SENS4210
C..........PARAMETER VALUES ARE CHANGED FOR SET INTERACTIONS.             SENS4220
   700 K20 = TRUN - NRUNN                                                SENS4230
       IF (I91.GT.1) K20 = TRUN - ISTORE(I91 - 1)                        SENS4240
       J21 = INSETS(I91,1)                                               SENS4250
       J22 = INSETS(I91,2)                                               SENS4260
       K22 = MADDR(J22+1) - MADDR(J22)                                   SENS4270
       K31 = (K20 - 1)/K22 + 1                                           SENS4280
       K32 = K20 - (K31 - 1)*K22                                         SENS4290
       K41 = K31 + MADDR(J21)                                            SENS4300
       K42 = K32 + MADDR(J22)                                            SENS4310
       JRUN = K41                                                        SENS4320
   710 CONTINUE                                                          SENS4330
       IF (J21 - NSETS1) 430,430,580                                     SENS4340
   720 INTXXX = 0                                                        SENS4350
       JRUN = K42                                                        SENS4360
       IF (J22 - NSETS1) 430,430,580                                     SENS4370
   730 CONTINUE                                                          SENS4380
C------------------------------------------------------------------------SENS4390
C      INITIAL VALUES OF THE STATE VARIABLES FOR THE CURRENT RUN ARE     SENS4400
C      RECORDED FOR USE BY THE SENOUT SUBROUTINE.                        SENS4410
C------------------------------------------------------------------------SENS4420
       REWIND 8                                                          SENS4430
       WRITE (8) STATE                                                   SENS4440
       RETURN                                                            SENS4450
       END                                                               SENS4460
```

FLOW CHART FOR SENOUT

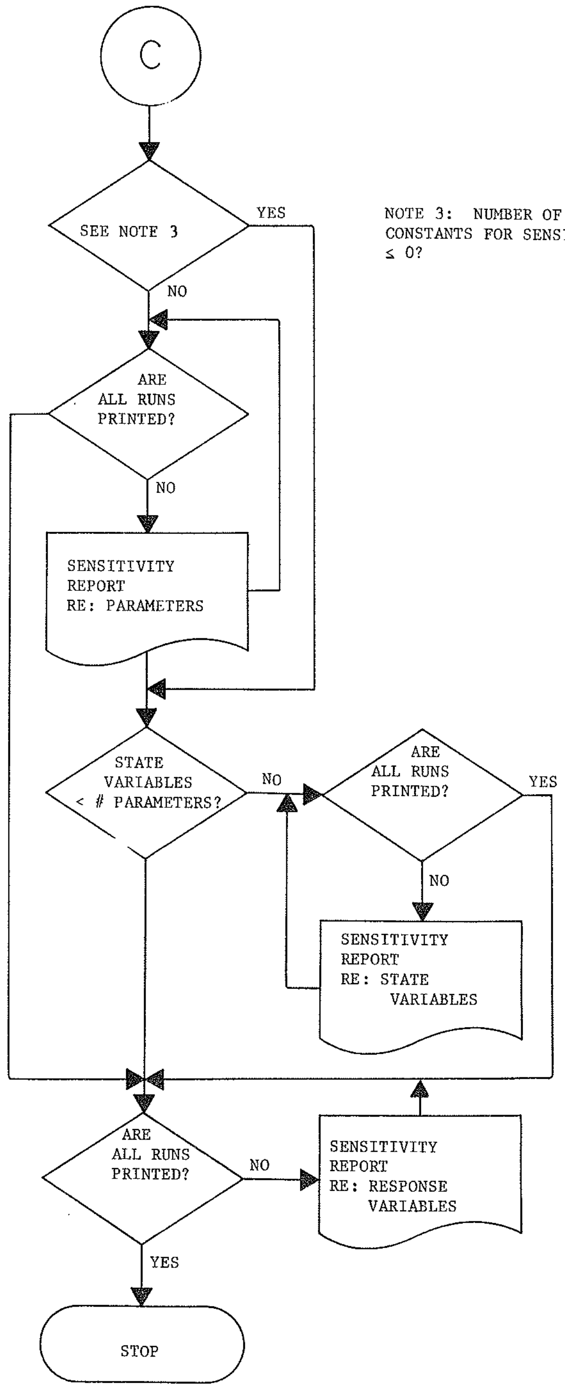


NOTE 1: DATE FOR CALCULATING RESPONSE VARIABLES SAME AS CURRENT SIMULATION DATE?

2.1.3.1.3 -34



NOTE 2 : REWIND AND READ STATE VARIABLE FROM LOGICAL (8).



NOTE 3: NUMBER OF PROCESS MODEL CONSTANTS FOR SENSITIVITY ANALYSIS ≤ 0?

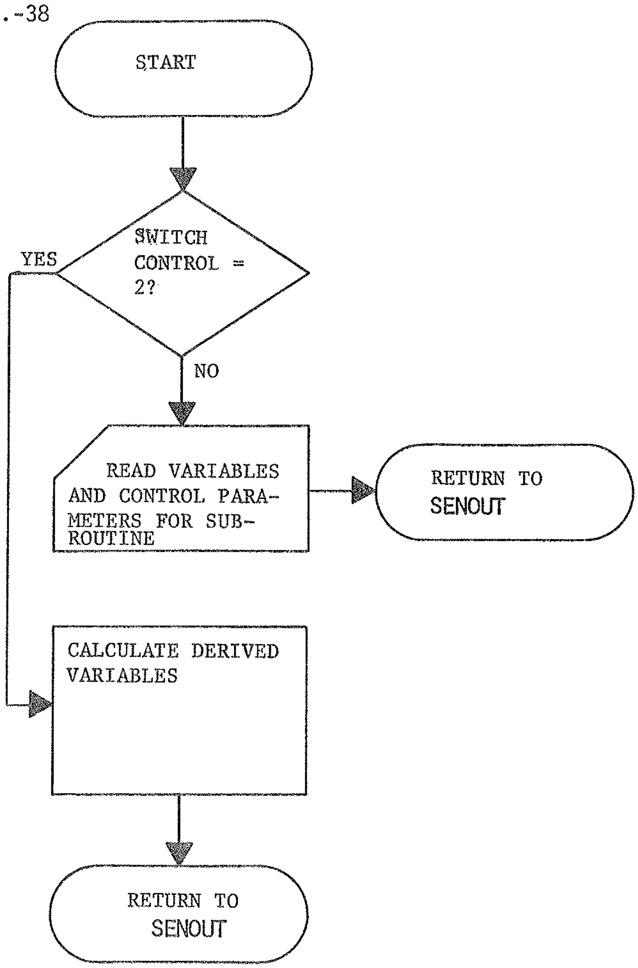
SENOUT

PROGRAM LISTING

```
                                                                         SOUT0010
C     SUBROUTINE SENOUT STORES VALUES OF RESPONSE VARIABLES SPECIFIED    SOUT0020
C     FOR THE SENSITIVITY CALCULATIONS AND PRINTS THE RESULTS OF THE SENSITIVITY SOUT0030
C     TESTS AT THE CONCLUSION OF THE JOB.                                SOUT0040
                                                                         SOUT0050
      SUBROUTINE SENOUT (ISW,IDAY,IRUN)                                  SOUT0060
      COMMON /PARAM/ P(1020C)                                            SOUT0070
C------------------------------------------------------------------------SOUT0080
C     THE COMMON BLOCK /NEWVAR/ IS REQUIRED FOR RESPONSE VALUES          SOUT0090
C     CALCULATED BY THE SUBROUTINE DERIVS.                               SOUT0100
C------------------------------------------------------------------------SOUT0110
      COMMON/NEWVAR/VNEW(10)                                             SOUT0120
      COMMON/STAT/STATE(1670)                                            SOUT0130
      COMMON/TOTALS/SUMS(1459)                                           SOUT0140
      COMMON/ACC/ STNG(19)                                               SOUT0150
      COMMON/RESP/ ITYP(20), INUM(20), IDAYT(20), NRESP,                 SOUT0160
     1 NPAR, NRUN, NOPAR(100),NPARAM,NPARA1, NSTATE                      SOUT0170
      DIMENSION PARXX(30,30), RX(30,30)                                  SOUT0180
      DATA LIMRUN/70/                                                    SOUT0190
      IF (ISW.GT.3) ISW = 1                                              SOUT0200
      GO TO (10, 110, 160, 10, 10), ISW                                  SOUT0210
   10 IF  (NRUN.LE.LIMRUN) GO TO 30                                      SOUT0220
      WRITE (6,20)                                                       SOUT0230
   20 FORMAT ('0TOO MANY RUNS')                                          SOUT0240
      STOP                                                               SOUT0250
C------------------------------------------------------------------------SOUT0260
C     IF RESPONSES ARE REQUIRED ON THE CURRENT DAY OF THE SIMULATION,    SOUT0270
C     THEY ARE STORED, AFTER CALCULATION IF NECESSARY.                   SOUT0280
C------------------------------------------------------------------------SOUT0290
   30 DO 100 I = 1, NRESP                                                SOUT0300
      IF (IDAY.NE.IDAYT(I)) GO TO 100                                    SOUT0310
      INA = INUM(I)                                                      SOUT0320
      IT = ITYP(I)                                                       SOUT0330
      ISV=2                                                              SOUT0340
      IF(IT.EQ.5)CALL DERIVO(INA,ISV)                                    SOUT0350
      GOTO(40,50,60,70,80),IT                                            SOUT0360
   40 RX(I,IRUN) = STATE (INA)                                           SOUT0370
      GO TO 90                                                           SOUT0380
   50 RX(I,IRUN) = SUMS  (INA)                                           SOUT0390
      GO TO 90                                                           SOUT0400
   60 CONTINUE                                                           SOUT0410
      GO TO 90                                                           SOUT0420
   70 RX(I,IRUN) = STNG  (INA)                                           SOUT0430
      GOTO 90                                                            SOUT0440
   80 RX(I,IRUN) = VNEW  (INA)                                           SOUT0450
   90 CONTINUE                                                           SOUT0460
  100 CONTINUE                                                           SOUT0470
      GO TO 330                                                          SOUT0480
  110 IF (NPARAM.LE.0) GO TO 150                                         SOUT0490
C------------------------------------------------------------------------SOUT0500
C     AT THE END OF EACH RUN, THE VALUES OF THE PARAMETERS SUBJECT TO    SOUT0510
C     CHANGE ARE RECORDED.                                               SOUT0520
C------------------------------------------------------------------------SOUT0530
      REWIND 2                                                           SOUT0540
      READ (2) STATE                                                     SOUT0550
      DO 120 I = 1, NPARAM                                               SOUT0560
```

```
      J = NOPAR(I)
  120 PARXX(I,IRUN) = P(J)
  130 IF(NSTATE.LE.NPARAM) GO TO 150
      DO 140 I = NPARA1, NSTATE
      J = NOPAR(I)
  140 PARXX(I,IRUN) = STAT (J)
  150 IF (IRUN.LT.NRUN)GO TO 330
  160 CONTINUE
C-----------------------------------------------------------------------
C     WHEN THE WHOLE JOB IS COMPLETED, THE RESULTS ARE PRINTED.
C-----------------------------------------------------------------------
      WRITE (6,170)
  170 FORMAT ('1   SUMMARY    OF    RUNS')
  180 FORMAT (1H , 5X, I5, 8F15.4)
      IF (NPARAM.LE.0) GO TO 230
      M1 = 1
      M2 = 8
  190 IF (M1.GT.NRUN) GO TO 240
      M = MINO(M2, NRUN)
      WRITE (6,200)           (I, I=M1, M)
  200 FORMAT ('0    IRUN', I13, 7I15)
      WRITE (6,210)
  210 FORMAT (' PARAMETER')
      DO 220 I = 1,NPARAM
  220 WRITE (6,180) NOPAR(I), (PARXX(I,J), J=M1, M)
  230 IF (NSTATE.LE.NPARAM) GO TO 280
      M1 = M  + 1
      M2 = M + 8
      GO TO 190
  240 IF (NSTATE.LE.NPARAM) GO TO 290
      M1 = 1
      M2 = 8
  250 IF (M1.GT.NRUN) GO TO 290
      M = MINO(M2, NRUN)
      WRITE (6,200) (I, I = M1, M)
      WRITE (6,260)
  260 FORMAT ('0STATE VARIABLE')
      DO 270 I = NPARA1, NSTATE
  270 WRITE (6,180) NOPAR(I), (PARXX(I,J), J=M1,M)
  280 M1 = M + 1
      M2 = M + 8
      GO TO 250
  290 M1 = 1
      M2 = 8
  300 IF (M1.GT.NRUN) STOP
      M = MINO(M2, NRUN)
      WRITE (6,200) (I, I=M1,M)
      WRITE (6,310)
  310 FORMAT (' RESPONSE ')
      DO 320 I = 1, NRESP
  320 WRITE (6,180) I, (RX(I,J), J=M1, M)
      M1 = M + 1
      M2 = M + 8
      GO TO 300
  330 CONTINUE
      RETURN
      END
```

FLOW CHART FOR DERIVD



FLT 005-03/09-14:37

```
C THIS SUBROUTINE CALCULATES WEIGHTED SUMS OF STATE VARIABLES AS
C REQUIRED BY THE SUBROUTINE SENOUT.

      SUBROUTINE DERIVD (INA,ISV)
      COMMON/STAT/STATE(1670)
      COMMON/TOTALS/SUMS(1459)
      COMMON/ACC/ STNG(18)
      COMMON/NEWVAR/VNEW(10)
      DIMENSION ITYP(10),NVAR(10),IVAR(10,20),WVAR(10,20)
      IF(ISV.EQ.2)GOTO30
C-----------------------------------------------------------------------
C     PARAMETER ADDRESSES AND MULTIPLIERS ARE READ IN.
C-----------------------------------------------------------------------
      READ10,ITYP(INA),NVAR(INA)
      NV=NVAR(INA)
      READ10, (IVAR(INA,J),J=1,NV)
      READ20,(WVAR(INA,J),J=1,NV)
   10 FORMAT(16I5)
   20 FORMAT(8F10.4)
      RETURN
   30 CONTINUE
C-----------------------------------------------------------------------
C     THE RESPONSE VARIABLE IS CALCULATED.
C-----------------------------------------------------------------------
      VNEW(INA)=0.
      NV=NVAR(INA)
      DO90J=1,NV
      IV=IVAR(INA,J)
      IT=ITYP(INA)
      GOTO(40,50,50,70),IT
   40 Q=STATE(IV)
      GOTO80
   50 Q=SUMS(IV)
      GOTO80
   60 CONTINUE
      GOTO80
   70 Q=STNG(IV)
   80 VNEW(INA)=VNEW(INA)+Q*WVAR(INA,J)
   90 CONTINUE
      RETURN
      END
```

INPUT/OUTPUT EXAMPLES

These examples are based on data from Rock Valley, using Version III of each process subroutine. Only input and output specific to these subroutines is reproduced.

EXAMPLE I

In this example, the parameters varied are the constants controlling

399 *Ephreda* rate of photosynthesis

1739 *Ephreda* rate of translocation from young stems

One alternative value for each of these parameters is used separately in runs 2 and 3, in combination in run 4, and the responses on two dates of two state variables are tested.

These response variables are:

1 Total carbon in young stems of *Ephreda* (address 256) on April 30.

2 Total carbon in young stems of *Ephreda* (address 256) on May 15.

3 Total carbon in seeds of *Ephreda* (address 301) on April 30.

4 Total carbon in seeds of *Ephreda* (address 301) on May 15.

EXAMPLE I - INPUT

						Input Statement Number
2	1	4	0	0	0	SENSIT 138
1	1739	1				SENSIT 150
	.04					SENSIT 151
1	399	1				SENSIT 150
	.005					SENSIT 151
1	2					SENSIT 170
2	256	120				SENSIT 264
2	256	135				SENSIT 264
2	301	120				SENSIT 264
2	301	135				SENSIT 264

EXAMPLE I - OUTPUT

SUMMARY OF RUNS

IRUN	1	2	3	4
PARAMETER				
399	.0010	.0010	.0050	.0050
1739	.0300	.0400	.0300	.0400

IRUN	1	2	3	4
RESPONSE				
1	2314.8548	2405.1715	2281.7584	2371.3781
2	2577.7919	2760.5218	2509.6163	2689.2925
3	5766.2277	6309.3925	5796.4032	6339.7650
4	7442.7401	8544.1537	7499.4282	8600.5635

EXAMPLE II

The second example varies three sets of parameters separately, viz.

(a) Photosynthesis rates of *Eurotia* and of the annuals are changed to a new set of values, not increased by a common factor.

(b) The initial weight of *Eurotia* leaves (all constituents) is doubled, and then halved.

(c) The initial weight of seed reserves of the annuals (all constituents) is doubled, and then halved.

Then interaction between set (a) and set (c) is tested.

The response to these variations which is tested is the value of a weighted sum of certain of the ecosystem components most valuable as livestock forage.

The quantities varied are:

Parameter 1743 Photosynthesis rate of *Eurotia*

Parameter 1745 Photosynthesis rate of annuals during phenological stage 3

Parameter 1960 Photosynthesis rate of annuals during phenological stage 4

Initial values of State Variables:

8 Content of nitrogen in *Eurotia* leaves
158 Content of ash elements in *Eurotia* leaves
308 Content of protein carbon in *Eurotia* leaves
458 Content of reserve carbon in *Eurotia* leaves
608 Content of structural carbon in *Eurotia* leaves

910 Content of nitrogen in shed seeds of annuals
970 Content of ash elements in shed seeds of annuals
1030 Content of protein carbon in shed seeds of annuals
1090 Content of reserve carbon in shed seeds of annuals
1150 Content of structural carbon in shed seeds of annuals

The three parameters (photosynthesis rates) were varied to a new arbitrary set of (larger) values in run 2; in runs 3 and 4 the state variables in the first set (composition of *Eurotia* leaves) were doubled, and then halved; in runs 5 and 6 the same was done with the second set of state variables (composition os shed seeds of annuals); and runs 7 and 8 repeated runs 5 and 6, with the alternative set of values for the three photosynthesis rates.

The response variables are taken on three dates:

1 January 20
2 February 9
3 March 1

The variables included, in each of these response variables and the weights applied to them are as follows:

Address	Variable	Weight
245	Total carbon in *Eurotia* leaves	1.00
247	Total carbon in annual leaves	1.00
260	Total carbon in *Eurotia* young stems	0.60
262	Total carbon in annual stems	0.80
292	Total carbon in annual inflorescences	1.00

EXAMPLE II - INPUT

									Input Statement Number
0	1	3	1	2	1				SENSIT 138
1	3	1	1704	1745	1760				SENSIT 204
	.4								SENSIT 216
	.2								SENSIT 216
	1.75								SENSIT 216
2	5	1	8	158	308	458	608		SENSIT 229
	2.0								SENSIT 239
2	5	1	910	970	1030	1090	1150		SENSIT 229
	2.0								SENSIT 239
1	3								SENSIT 250
5	1	385							SENSIT 264
2	5								DERIVD 15
245	247	260	262	292					DERIVD 17
	1.0		1.0		0.6		0.8	1.0	DERIVD 18
5	2	405							SENSIT 264
2	5								DERIVD 15
245	247	260	262	292					DERIVD 17
	1.0		1.0		0.6		0.8	1.0	DERIVD 18
5	3	425							SENSIT 264
2	5								DERIVD 15
245	247	260	262	292					DERIVD 17
	1.0		1.0		0.6		0.8	1.0	DERIVD 18

EXAMPLE II - OUTPUT

JMMARY OF RUNS

:RUN	1	2	3	4	5	6	7	8
\METER								
1743	.3000	.4000	.3000	.3000	.3000	.3000	.4000	.4000
1745	.1600	.2000	.1600	.1600	.1600	.1600	.2000	.2000
1760	1.2200	1.3500	1.2200	1.2200	1.2200	1.2200	1.3500	1.3500
:RUN	1	2	3	4	5	6	7	8
'E VARIABLE								
8	.5000	.5000	1.0000	.2500	.5000	.5000	.5000	.5000
158	1.6000	1.6000	3.2000	.8000	1.6000	1.6000	1.6000	1.6000
308	1.5000	1.5000	3.0000	.7500	1.5000	1.5000	1.5000	1.5000
458	4.0000	4.0000	8.0000	2.0000	4.0000	4.0000	4.0000	4.0000
608	3.0000	3.0000	6.0000	1.5000	3.0000	3.0000	3.0000	3.0000
910	10.0000	10.0000	10.0000	10.0000	20.0000	5.0000	20.0000	5.0000
970	10.0000	10.0000	10.0000	10.0000	20.0000	5.0000	20.0000	5.0000
1030	30.0000	30.0000	30.0000	30.0000	60.0000	15.0000	60.0000	15.0000
1090	48.0000	48.0000	48.0000	48.0000	96.0000	24.0000	96.0000	24.0000
1150	10.9100	10.9100	10.9100	10.9100	21.8200	5.4550	21.8200	5.4550
RUN	1	2	3	4	5	6	7	8
ONSE								
1	17.6728	18.5279	25.8964	13.5528	27.1055	12.9564	28.8074	13.3839
2	33.7464	38.7004	40.3207	29.6264	56.8902	22.1745	62.9037	24.6514
3	115.9820	155.7815	122.3951	109.3559	211.0813	68.4324	270.5533	90.0415