AGENCY FOR INTERNATIONAL DEVELOPMENT
WASHINGTON, D. C. 20523

BIBLIOGRAPHIC INPUT SHEET

FOR AID USE ONLY
BATCH #14

1. SUBJECT CLASSIFICATION

A. PRIMARY
Agriculture AA00-0000-G214

B. SECONDARY
General--Nigeria

2. TITLE AND SUBTITLE
A programmer's description of NISIM,the executive program for a simulation model of the Nigerian agricultural economy

3. AUTHOR(S)
Page,Gloria; Carroll,T.W.

4. DOCUMENT DATE	5. NUMBER OF PAGES	6. ARC NUMBER
1969	75p.	ARC

7. REFERENCE ORGANIZATION NAME AND ADDRESS
Mich.State

8. SUPPLEMENTARY NOTES *(Sponsoring Organization, Publishers, Availability)*
(In Project working paper 69-2)

9. ABSTRACT

10. CONTROL NUMBER
PN-RAA-722

11. PRICE OF DOCUMENT

12. DESCRIPTORS
Computer programs
Decision making
Nigeria
Simulation

13. PROJECT NUMBER

14. CONTRACT NUMBER
CSD-1557 Res.

15. TYPE OF DOCUMENT

AID 590-1 (4-74)

A SIMULATION MODEL OF THE

NIGERIAN AGRICULTURAL ECONOMY

Contract No. AID/csd-1557

A Programmer's Description of NISIM --The Executive Program for a Simulation Model of the Nigerian Agricultural Economy

Project Working Paper 69-2

Prepared By:
Gloria Page
Thomas W. Carroll

December 31, 1969
East Lansing, Michigan

A PROGRAMMER'S DESCRIPTION OF NISIM
--THE EXECUTIVE PROGRAM FOR A SIMULATION MODEL
OF THE NIGERIAN AGRICULTURAL ECONOMY

Executive Program NISIM consists of two major parts. The first part--which can be called the initialization phase--sets initial conditions for variables and constant values for parameters of the various model sub-programs. This initialization phase of NISIM also sets control parameters and switches (which then help direct its own activities and those of the model's sub-programs).

The initialization section includes all of program NISIM up to the beginning of the "time loop" (DØ 1 ITER = 1,NTIME) in which T is incremented; i.e. the initialization phase includes all activities of NISIM during which T = 0 or has not yet been given any value.

The second part of NISIM--which can be called the simulation phase--directs calls to the model's subroutines, converts some output variables of one subroutine to a form usable as input to another subroutine, computes a few criterion variables, compares simulation output with actual data series (if desired) and controls a small part of the model's output printing. (Most of a job's output is already directed by various subroutines, eg. VARPLØT, CRTRN, SUMRYC.)

The second section of program NISIM begins with the time loop DØ 1 ITER = 1,NTIME and includes all the remainder of NISIM, i.e. the simulation phase includes all activities of NISIM during which

$DT \leq T \leq DT*NTIME$

where DT is our time increment and NTIME is the upper limit of the time loop index ITER. Hence NTIME is the integral number of times T is incremented during a simulation "run". So DT * NTIME is the "run" length or the value of T at the end of a simulation run.

The fortran listing of executive program NISIM to be found at the end of this description was made during job #CS46989 on December 31, 1969.

As model changes and additions affect program NISIM, this description will be updated.

TABLE OF CONTENTS

Part 1 -- Initialization Phase.....Page 7

I. Control Parameters and Switches of NISIM.....Page 7

A. DT.....Page 7

B. NTIME.....Page 7

C. NR.....Page 7

D. IDT, IPRINT.....Page 8

E. NPRTCY.....Page 10

F. ISSQR.....Page 11

G. ISSTIME.....Page 11

H. $INTERP_k$.....Page 12

I. ISENS.....Page 12

J. IPSENS.....Page 15

II. Data Statements of NISIM.....Page 16

II. Subroutine PØPDAT.....Page 16

IV. Subroutine PØPSET.....Page 17

V. Further Initialization in Program NISIM.....Page 17

A. Variables set to constant initial values in arithmetic assignment statements.....Page 17

B. First call to subroutine PØPØUT during each run.....Page 17

C. Computation of consistent initial values for input to LANDN.....Page 17

D. Initial Values for the BEEF Sector.....Page 18

Part 2 -- Simulation Phase.....Page 19

I. Interface Variables of Program NISIM .....Page 19

1. ATL_k.....Page 19

2. PM(3), PM(4) .....Page 20

3. PPRS(3), PPRS(4) .....Page 20

4. ATL_k (test for ≤ 0).....Page 21

5. $LABA_k$, $ALABA_k$.....Page 21

6. TDNRES .....Page 22

7. CALAN, CALAM .....Page 22

8. CRU_k.....Page 22

9. SUPCØT, DEMCØT, FØREXC, FØREXG .....Page 23

10. $SIMVAL_j$.....Page 23

11. Y1, Z1, Y2, Z2, Y3, Z3 arrays .....Page 24

12. PM(1), PM (2) .....Page 24

13. PP(1), PP(2) .....Page 25

14. SUPCA.....Page 26

II. Very Brief Notes on Subroutine Calls and Print Control .....Page 27

III. Conventional Documentation for Subroutine SENSINC.....Page 28

IV. Conventional Documentation for Subroutine SUMSQR.....Page 34

V. Population Component: Purpose of Each Subroutine.....Page 38

Flow Charts, Parameter Lists, Fortran Listings

Page

A non-detailed chart of Program NISIM A-1

--which refers to lists of parameters (pages C-1 to C-7) to supply details of the initialization phase of the program. The scope of the "time" loop DØ 1 ITER = 1,NTIME is omitted since it is given in complete detail in the flow chart found on pages B-1, B-2, B-3.

A conventional flow chart of the simulation phase of Program NISIM.....B-1
B-2
B-3

--a flow chart of all activities of NISIM after initialization occurs. This flow chart starts with the beginning of the "time" loop and continues to the end of the program.

A list of parameters preset in data statements in NISIM.....C-1

A list of parameters preset in subroutine PØPDAT.....C-2

A list of parameters set back each run in subroutine PØPSET.....C-3

A list of variables whose initial values are set in program NISIM.....C-4

A list of LANDN variables whose initial values are computed in program NISIM C-5

A list of BEEF parameters whose initial values are set in program NISIM C-6

A list of variables and data plotted.....C-7

A conventional flow chart of subroutine SUMSQR.....D-1
D-2

A Fortran listing of SUMSQR.....D-3

A conventional flow chart of subroutine SENSINC.....E-1

A Fortran listing of SENSINC.....E-2

A list of function and subroutine sub-programs called by program NISIM.....F-1

A Fortran listing of executive program NISIM.....G

Part 1 -- Initialization

I. Control Parameters and Switches of NISIM

Nearly all control parameters and switches are now set by a call to subroutine PARAMS. This call is the first executable statement of program NISIM.

A. DT: The time increment added to T at the beginning of each pass thru the time loop {DØ 1 ITER = 1,NTIME}. Until recently DT = .1 or 1/10 year. Presently DT = .25 or 1/4 year, but can be altered at will, i.e. no programming changes are made necessary by a change in DT.

B. NTIME: The upper limit on the index ITER of the time loop DØ 1 ITER = 1,NTIME. Since T is initialized at 0 and since the statement T = T + DT (inside the time loop) is executed NTIME times, the simulation run length (or the value of T at the end of a simulation run) = DT * NTIME.

The user having chosen a value for DT and decided how long a simulation run should last (which run length may of course be any integral multiple of DT) will choose NTIME = desired run length/DT. (eg. for DT = .25 and a desired run time of 20 years, NTIME = 80)

C. NR: The Number of Runs of a job. NR is the upper limit of the index M of the run loop DØ 22 M = 1,NR. (M is then the run number)

At the beginning of each run, i.e. just after the statement DØ 22 M = 1,NR, T is set back to 0. The time loop is nested within the run loop so for each run T increases from 0 to DT * NTIME. Two sets of parameters will also have values assigned near the beginning of <u>each</u> run:

1) variables which are required to have initial values, eg. SFL array for LANDN, and
2) parameters whose values are constant during a run but whose values may be changed from one run to the next, eg. for sensitivity tests so that each run has a specific set of initial conditions which affect the results of the passage thru the time loop for that run. (See page A-1)

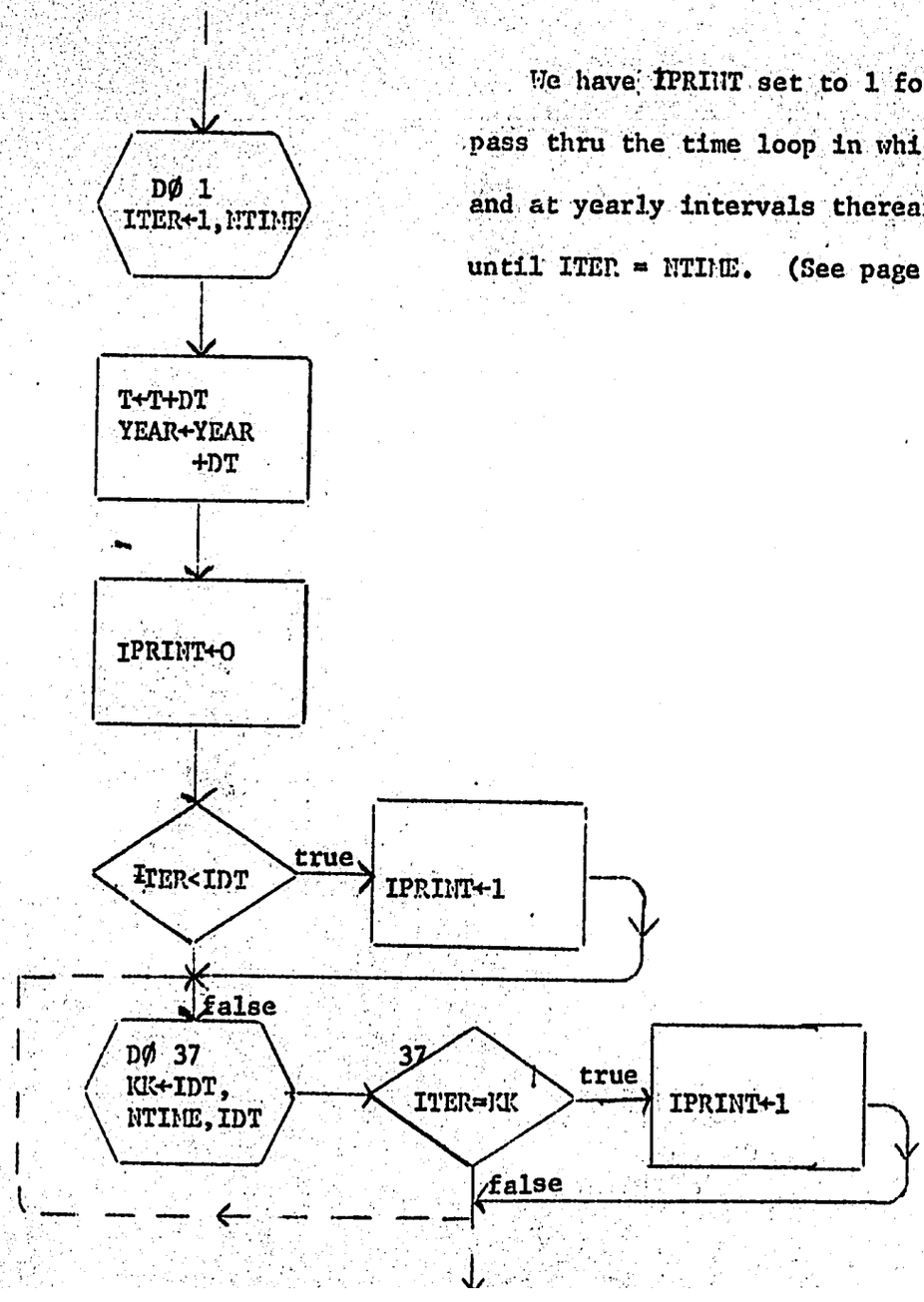
D. IDT, IPRINT: IDT = 1./DT + .00001

IDT is not set by the user.

IDT is a control parameter for the regulation of printing by program NISIM, subroutine CRTRN, (and eventually by all subprograms which compute values of variables which are to be printed.)

IDT, NTIME, and the value of ITER at each pass through the time loop together determine the value of IPRINT, the print "switch".

We have IPRINT set to 1 for each pass thru the time loop in which T $\leq$ 1 and at yearly intervals thereafter until ITER = NTIME. (See page B-1)



E. NPRTCY: NPRTCY regulates the printing of population output at regular intervals.

In subroutine PØPOUT we have

```
IF (MØD (ITIME, NPRTCY). EQ.0
        .AND. NPRTCY. LT. 350) CALL PØPPRT
```

The first test of the if statement is equivalent to asking: is ITER (ITIME is the local name for ITER in PØPØUT) an integral multiple of NPRTCY.

The test for NPRTCY <350 simply provides a way to turn off the printing of human population output. If this output is not desired, set NPRTCY to some value $\geq$ 350.

Hence population output occurs whenever NPRTCY < 350 and ITER = K * NPRTCY where K is some non-negative integer and ITER is the time loop index.

The user, having chosen a value for DT and having decided to print population output every Y years, (where Y <u>must</u> be an integral multiple of 3!) will choose NPRTCY = Y/DT. eg. if he wishes to print every 6 years and DT = .25 then he will set NPRTCY = 6/.25 = 24.

Many human population variables are updated only every 3 years--not every pass through the time loop. Hence meaningful printing can only be done every Y years where Y is an integral multiple of 3. So NPRTCY must always be some integral multiple of 3/DT or 3 * IDT.

F. ISSQR: The user will set ISSQR = 1 whenever he wishes to compare simulation output with actual data series.

If and only if ISSQR = 1 and ISSTIME = 1 (for ISSTIME, See below)

Subroutine SUMSQR is called and compares

computed PFN with SERIESA (NN,1)	
computed ØUTP(1) with SERIESA (NN,2)	NN = 1,13
computed ØUTP(2) with SERIESA (NN,3)	

Where	PFN is the market price for food in the North	f/lb
	ØUTP(1) is groundnut production	k lb/year
	ØUTP(2) is cotton production	k lb/year

and SERIESA array contains actual data 1953-1965

Note: A flow chart for Subroutine SUMSQR appears on pages D-1 & D-2. A Fortran listing appears on page D-3. Conventional documention of subroutine SUMSQR is found in Part 2, section IV.

G. ISSTIME: The user does not set ISSTIME. It is set to 0 first during every pass thru the time loop. Then it will be set to 1 <u>only if</u> data exists for that point in time, T (which corresponds to the present value of ITER--See page B-2 for computation of ISSTIME) <u>and only if</u> ISSQR = 1 for that job. (Presently ISSTIME = 1 at T = 1., 2., 3., ..., 12., 13. See SUMSQR description)

Note: Values of PFN, ØUTP(1) and ØUTP(2) are stored in dummy arrays for plotting comparisons with actual data only during passes thru the time loop for which ISSTIME = 1. (See Page B-3)

H. INTERP(K) K = 1,4:

If the user wishes to interpolate linearly for PP(1) and PP(2) from actual data for groundnut and cotton prices,

he will set INTERP(1) = 1

and INTERP(2) = 1

In this case, a FORDYN function sub-program TABEXE is referenced (which also extrapolates linearly when T is outside the time range of the data). The data is found in arrays VALG and VALC, each dimensioned to 13 for the yearly prices between 1953 and 1965. (See page B-1)

For INTERP(K) = 1 the usual computation of PM(1) and PM(2) in NISIM, dependent on world price levels, is bypassed. K=1,2 (See page B-1)

Instead, after entry to subroutine AMP, PM(1) and PM(2) are computed, dependent on the PP(1) and PP(2) from the data interpolation, and the usual computation of PP(1) and PP(2) in terms of PM(1) and PM(2) is also bypassed in subroutine AMP.

At present there is no similar set up for food prices, although INTERP(3) and INTERP(4) exist. They are defined and set to 0 so that the PM and PP computation decisions in subroutine AMP could be made general. So for now INTERP(3) and INTERP(4) must always have values ≠1.*

I. ISENS: If and only if ISENS = 1 for a job, subroutine SENSINC is called near the beginning of each run of that job. (See page A-1)** SENSINC permits the user to determine the sensitivity of the model to

* 12/31/69 INTERP(1) and INTERP(2) are both given the value 0 after YEAR = 1966. (YEAR = T + 1953.) Value assignments moved from PARAMS to inside the run loop.

** 12/31/69 SENSINC is called from NISIM if and only if ISENS = 1 and IPSENS ≤ 0. (See IPSENS below)

preassigned percentage changes in the initial or constant values of key model parameters. (flow chart page E-1, description page 28)

Sensinc replaces the values of each of from one to twenty elements of "dummy" array SVAR with values obtained by multiplying each (singly, i.e. only one incrementation per run) by an incrementation factor ZIN.

No incrementation is performed in run 1; it is a comparison run. Hence if K parameters are to be tested; $1 \leq K \leq 20$ then NR must be set to K + 1.

In general Sensinc increments

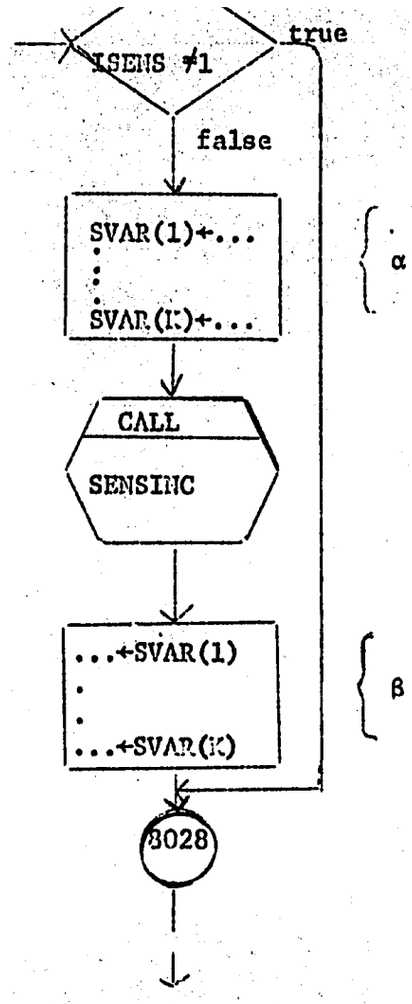
SVAR(J) in run number J + 1

Specifically, the user must, for a sensitivity job,

1. set ISENS = 1
2. choose K parameters $1 \leq K \leq 20$ for testing
3. set NR = K + 1
4. choose a value for ZIN. Eg. if a 25% increase is desired, set ZIN = 1.25.
5. give hollerith values to the Name array in a data statement (for output)

Suppose the parameters are B1, B2, TDNAD. Then the user will insert DATA (NAME = 2HB1, 2HB2, 5HTDNAD)

6. In general, the user will insert 2 blocks (α and β) of simple assignment statements into the usual NISIM program. These "blocks" simply will store values in and return values from the "dummy" array SVAR. (See page A-1)



Suppose the parameters are B1, B2, TDNAD, RLTT, B27. Then the user would insert

```
α { SVAR(1) = B1
    SVAR(2) = B2
    SVAR(3) = TDNAD
    SVAR(4) = RLTT
    SVAR(5) = B27
```

and

```
β { B1 = SVAR(1)
    B2 = SVAR(2)
    TDNAD = SVAR(3)
    RLTT = SVAR(4)
    B27 = SVAR(5)
```

To continue this example; the user would set IR = 6 and insert

```
DATA (NAME = 2HB1, 2HB2, 5HTDNAD,
  4HRLTT, 3HB27)
```

J. IPSENS:

IPSENS is a switch for the population "sensitivity" subroutine PØPSEN. PØPSEN is called by subroutine PØPSET.

Whenever IPSENS>0 a call to PØPSEN is executed from PØPSET. (This subroutine will be described later with the rest of the human population sector. However PØPSEN does in turn call subroutine SENSINC.)

Whenever IPSENS $\leq$ 0 no call to PØPSEN is executed.

(As will be seen in the documentation for PØPSEN, the user must allow IPSENS to take on <u>only</u> the values 0, 1, 2, or 3.)

II. Data Statements of NISIM

Data statements are non-executable Fortran statements which cause constant values to be assigned to simple variables and arrays at the time the program or sub-program is loaded, i.e. before execution begins. Although the values of these variables and arrays may change as execution occurs, they can never be set back to their original values (during the same job) by the use of data statements.

Hence data statements in NISIM are used to assign values to parameters which are constants and whose values (normally) do not change from run to run.

Data statements inside program NISIM set values for parameters used by subroutines SENSINC, SUMSQR, LANDN, AMP, BEEF, and for commodity price equations and for the interpolation of the groundnut and cotton producer prices.

A complete list of parameters set in data statements in Nisim is found on page C1.

III. Subroutine PØPDAT

Subroutine PØPDAT is called just once during a job; just after the call to PARAMS which sets control parameters and switches, and just before the beginning of the run loop DØ 22 M = 1,NR.

A complete list of initial and constant values set (mostly in data statements) by PØPDAT appears on page C-2.

PØPDAT also writes on a disc the initial or constant values which need to be reset <u>each run</u>. This list appears on page C-3.

IV. Subroutine PØPSET

Subroutine PØPSET is called once during each run of a job.

During run 1 POPSET merely assigns values to five parameters; ELASFP, ELASFC, ELASFY, CALPP, SPØIL.

During any run after the first, PØPSET reads the values written by subroutine PØPDAT (in effect, restoring their initial values of run 1) and assigns again values to ELASFP,..., SPØIL. (See page C-3.)

V. Further Initialization in Program NISIM

A. Between the call to PØPSET and the call to PØPSEN there is a list of simple variables and arrays set to constant values. Please see the list on page C-4.

Please note: PØPSEN will be described, together with the rest of the human demographic features of the model, elsewhere.

B. Call PØPØUT:

The functions of this demographic subroutine will be described elsewhere. What is relevant here is that this first call to PØPØUT each run (before the beginning of the "time" loop, see page A-1) provides initial values for $LABA_k$ and $DEMR_k$ K = 1,4 which are needed to compute certain initial values for LANDN variables.

C. Compute <u>consistent initial values</u> for inputs to subroutine LANDN. (A list appears C-5)

The user need be concerned with 3 points only concerning these com-computations:

1. The expressions for CALAN and CALAM are correct only so long as $ALABA_k = DEML_k$ for the production subroutine AMP. This equality will be true for the foreseeable future, but new expressions for CALAN and CALAM would need to be written if it were not.
2. CL20(J) is the proportion of AL(J) in cash food at T = 0.
   J = 1,2
3. CL21 is the proportion of AL(3) in cotton
   CL22 is the proportion of AL(3) in cash food
   } at T = 0.

The computed initial values for LANDN are printed, and comprise the first block of output for each run.

D. Set the constant and computed initial values for beef parameters. The computed values are printed, forming the second block of output for each run. (See C-6 for a list)

Part 2 -- Simulation Phase

I. Interface Variables of Program NISDM

The conversion of output variables of one subroutine into a form usable by another subroutine:

1. ATL array: During the Kth call to subroutine ANP (K = 1,4) in any pass thru the "time" loop, ATL(K) replaces the local variable LND of ANP. i.e. ATL(K) is the total cultivated acreage value for the Kth call to ANP.

   Output variables of subroutine LANDN are stored in the ATL array to be used as input to Subroutine ANP.

   ATL(1) = TGLN; total acreage in groundnuts in the North

   ATL(2) = TCLN; total acreage in cotton in the North

   ATL(3) = ZLC1 - TL(4); total food land in the North
   - total cultivated land in the middle belt

   (or total food land in the northern North,* considered to be in grain-food crops)

   ATL(4) = TL(4); total cultivated land in the middle belt (considered to be in root-food crops)

   where TGLN, TCLN, ZLC1, TL(4) are all output variables of LANDN.

   (See page B-1)

*By total food land in the northern North, we mean total land in the North where food is grown in competition with groundnuts and/or cotton.

2. PM(3), PM(4):

PM(3) and PM(4) are respectively the market prices for food in the northern North (grain-crops) and food in the middle belt (root-crops). They are input variables to AMT, replacing the local variable PM in that subroutine for the third and fourth calls at each time period.

PM(3) = PFN / (CP1*CP2 + CP3)

PM(4) = CP1*PM(3)

Where PFN is an "average" food price for the North computed by subroutine LANDN. (CP1, CP2, CP3 are simply constant-valued parameters set at the beginning of a job.) See page B-1)

3. PPRS(3), PPRS(4):

PPRS(3) and PPRS(4) are respectively the proportions of food sold in the northern North and the middle belt. They are input variables to subroutine AMT replacing the local variable PPRS in the third and fourth calls at each time period.

PPRS(3) = AMIN1 (1., DEMCFN / SUPCFN)

PPRS(4) = PPRS(3)

Where DEMCFN (demand cash food in the North) is an output variable of PØPØUT and SUPCFN (supply cash food in the North) is an output variable of LANDN.

(PPRS(1) and PPRS(2) are always = 1, i.e. assume that all of the groundnut and cotton crop can be sold.) (See page B-1)

4. The ATL array is given values from LANDN as described above in #1. But ATL(K) = 0. for any K = 1,4 would create chaos in subroutine AMP.

Hence inside the do-loop for the call to AMP, DØ36 L = 1,4 we have (before the call to AMP)

```
IF (ATL(L) .LE.0.) ATL(L) = .001
```

where .LE. is the fortran relation "less than or equal to".

5. LABA, ALABA arrays:

The LABA array values are computed by subroutine PØPØUT and used as input to subroutine LANDN.

These LABA values from PØPØUT are not appropriate for input to AMP. The values from PØPØUT are computed for the <u>ecological regions</u> of subroutine LANDN. The values required by subroutine AMP are the values for labor available to produce each "crop"; i.e. groundnuts, cotton, food (grain crops) in the northern North, and food (root crops) in the middle belt. So we define a new ALABA array for input to AMP.

Presently we require ALABA(L) $\geq$ DEML(L) L = 1,4. Inside the do-loop for the call to AMP we have ALABA(L) = ATL(L)*(C4(L) + C5(L)*PY(L)) where the ATL array is described above in #1 and #4 and the C4, C5, and PY arrays presently have constant values assigned at the beginning of each run. (Inside subroutine AMP we have

```
1  YLDD = LND*PY
2  DEML = (C4*LND + C5*YLDD) / CM
3  EMP = AMIN1 (LABA, DEML)
```

The CM(L) does not need to appear in the expression for ALABA(L) in NISIM because CM(L) $\geq$ 1. L = 1,4 and surplus ALABA (L) produces the same model output as ALABA (L) = DEMT (L).)

6. TDNRES:

TDNRES, the TDN available to beef from crop residues (rough forage) is an input variable to subroutine BEEF.

TDNRES = ØUT3(1) + ØUT3(2) + ØUT3(3) + ØUT3(4)

Where the ØUT3(K) K = 1,4 are output variables of subroutine AMP. (See page B-2)

7. CALAN, CALAM:

Calories produced on one acre of food land in the northern North and middle belt respectively; these are <u>input variables to subroutine LANDN</u>, used to compute the acreages needed to feed the farm population.

CALAN = C9(3) * ØUT1(3) / ATL(3)

CALAM = C9(4) * ØUT1(4) / ATL(4)

Where ATL(3) and ATL(4) are from LANDN as described in #1 and #4 above;

ØUT1(3) and ØUT1(4) are output variables of AMP, and C9(3) and C9(4) are constant valued parameters set at the beginning of a job. (See page B-2)

8. CRU array:

This array of unlagged cash returns is an important input to subroutine LANDN.

CRU(K) = RLRE(K) K = 1,4

Where RLRE(K) is an output variable of subroutine AMP.

8.5 YNFN: YNFN, total nonfarm income in the North, is an input variable to subroutine PØPØUT computed in NISIM itself. YNFN = YNFNI*EXP(.061*T)

9. SUPCØT, DEMCØT, FØREXC, FØREXG:

SUPCØT is used by program NISIM in the computation of FØREXC, which is an <u>input variable to subroutine CRTRN</u>.

SUPCØT = .345*ØUTP(2)

Where ØUTP(2) is computed by <u>subroutine amp</u>.

FØREXC = (SUPCØT-DEMCØT) * PCW

Where DEMCØT and PCW are variables computed in NISIM itself.

DEMCØT = 112000.*EXP((.025 + .00123*T)*T)

PCW = PCWØ*(1. + CW1*T)

FØREXG = ØUTP2(1) * PGW

PGW = PGWØ*(1. + CW2*T)

FØREXG is an input variable to subroutine CRTRN. ØUTP2(1) is computed by subroutine AMP. PGW - world price for groundnuts - is computed in NISIM itself. (See pages B-1, PCW, PGW, and B-2)

10. SIMVAL array:

This array of Simulation Values is the input array to subroutine SUMSQR which is compared to the array of actual data SERIESA.

SIMVAL(1) = PFN	£/lb.
SIMVAL(2) = ØUTP(1)	Klb./year
SIMVAL(3) = ØUTP(2)	Klb./year

Where PFN is an output variable of subroutine LANDN, an "average" food market price in the North; ØUTP(1) and ØUTP(2), cotton and groundnut production respectively, are outputs of subroutine AMP.

11. Y1, Z1, Y2, Z2, Y3, Z3 arrays:

These arrays are input arrays to subroutine VARPLØT,

Y1(NN) = PFN

Z1(NN) = SERIESA(NN,1)

Y2(NN) = ØUTP(1)

Z2(NN) = SERIESA(NN,2)

Y3(NN) = ØUTP(2)

Z3(NN) = SERIESA(NN,3)

Where PFN, ØUTP(1) and ØUTP(2) are as described in #10 above

and

NN = 1,13 →
- SERIESA (NN,1) contains actual data values for the northern food market price 1953-1965
- SERIESA (NN,2) contains actual data values for groundnut production 1953-1965
- SERIESA (NN,3) contains actual data values for cotton production 1953-1965

12. PM(1) and PM(2):

The computations for these two input variables to subroutine AMP are a special case.

PCW and PGW are computed in NISIM at every time period, for use in the equations for FØREXC and FØREXG.

PCW = PCWØ * (1. + CW1*T)

PGW = PGWØ * (1. + CW2*T)

But PM(1) and PM(2) are computed in NISIM only during jobs in which INTERP(1) and INTERP(2) ≠ 1. (See description of control parameters of NISIM) (When INTERP(1) = 1 and INTERP(2) = 1, PM(1) and PM(2) are computed inside AMP)

When INTERP(1) ≠ 1 and INTERP(2) ≠ 1 we have in NISIM:

PM(1) = PGW

PM(2) = .345*PCW + .0038

13. PP(1) and PP(2):

Conversely these two input variables to AMP are computed in NISIM if and only if INTERP(1) = 1 and INTERP(2) = 1

Then we have, at each time period,

PP(1) = TABEXE(VALG, 1., 1., 12, T)

PP(2) = TABEXE(VALC, 1., 1., 12, T)

Where TABEXE is a linear interpolation FØRDYN routine. (when T is outside the data time range, TABEXE extrapolates linearly)

VALG and VALC hold actual data for groundnut and cotton producer prices for 1953-1965.

Note: When INTERP(1) and INTERP(2) ≠ 1

PP(1) and PP(2) are computed inside AMP.

14. SUPCA

This variable is computed in program NIGSIM and printed with the output of subroutine LANDN, although it is not now an input variable to subroutine LANDN. (SUPCA may be used later in the calculation of per capita nutritional level.)

SUPCA = ØUT1(3) + ØUT1(4)

Where ØUT1(3) and ØUT1(4) are the total yearly outputs of K calories from the food land in the northern North and the middle belt respectively.

Additional Notes on NISIM'S Simulation Phase

1. Calls to model subroutines:

Preparations for and the order of NISIM'S calls to the model subroutines are detailed in the flow chart on pages B-1, B-2, B-3.

The model sub-programs themselves have been described in the current progress report.

The simulation phase of NISIM (during which DT $\leq$ T $\leq$ DT * NTIME) in its present form demands no choices by the user. Once initial values are chosen and control parameters and program switches are given values, the simulation phase proceeds by itself.

2. Print control:

If the user wishes to print at different intervals, he will have to re-program the IPRINT set up. Presently we print at each pass through the time loop until T = 1. and afterward at yearly intervals until T = DT * NTIME — regardless of the user's choice of DT and NTIME values.

Eg. if the user wishes to print every five years (as well as on each pass through the time loop until T = 1.) he will remove statement DØ 37 KK = IDT, NTIME, IDT and replace it with

```
IDT5 = 5 * IDT
DØ 37 KK = IDT5, NTIME, IDT5
```

III. Subroutine SENSINC (Flow chart page E-1, Fortran list page E-2)

SENSINC permits the user of executive program NISIM to "determine the sensitivity of the model" to preassigned changes in the initial or constant values of key model parameters. (A flow chart of subroutine SENSINC appears on page E-1)

SENSINC replaces the values of each of from one to twenty parameters with values obtained by multiplying each (singly, i.e. one and only one incrementation at the beginning of each run after the first run) by an incrementation factor ZIN.

The subroutine performs no incrementation in the first run -- this is the run against which all other runs are compared in the analysis of output. Note that as the subroutine is written now, if the user wishes to test K parameters (1.LE.K.LE.20) then NR (the number of runs) must be set to K + 1 in NISIM. Hence 2.LE.NR.LE.21 . (However the subroutine could be altered to permit non-sensitivity runs to be executed after run number K + 1. It is probably easier to insert statements in NISIM to bypass the SENSINC call after run K + 1.)

SENSINC increments the first parameter in the dummy array SVAR at the beginning of run number 2, the second parameter in SVAR at the beginning of run number 3, and so on.

SENSINC does no printing during the first run -- in which run no incrementation occurs. In each of the second to 21st possible runs SENSINC prints the run number, the parameter name and its unincremented value in the form:

Run number 4

Unincremented GRE = 8.000+000 .

SENSINC also prints the parameter name with its incremented value in the form:

Incremented GRE = 8.800+000 .

The subroutine has an alternate entry SENSDEC which restores its unincremented value to each parameter previously incremented. This entry is provided to enable the user to test for sensitivity parameters which are not reset by the executive program at the beginning of each run loop, i.e. parameters whose initial or constant values are set prior to the statement DØ 22 M = 1,NR. (An alternative to calling SENSDEC in such a case is to require that the parameters in question be reset in each run, i.e. that their initial value or constant value equations appear inside and at the beginning of the run DØ Loop DØ 22 M = 1,NR.)

PARAMETER LIST

Subroutine SENSINC (M, ZIN, OLAM, XLAM, SVAR, NAME)

M: Run number, integer. The index of the DØ Loop in which NISIM operates on a new set of data for each increment of the index variable M. When M=1 no incrementation occurs. When M=2 the parameter replacing SVAR(1) is incremented and so on until M=NR. Similarly if there is a 21st run, then of course the incrementation is performed on the parameter replacing SVAR(20).

ZIN: Multiplicative incrementation factor. The incremented value of each parameter = ZIN times the unincremented value of that parameter -- the value supplied to SENSINC by NISIM. eg. If the user wishes to increase the value of each parameter by 10 percent, then he will set ZIN to 1.1. Then EG. If GRE had the value 8. when SENSINC was called -- in the run that increments GRE its value 8. is replaced by 8.8 and control is returned to NISIM for execution of that run's computations.

If eg. the user wishes to increment each parameter in the call by 5 percent he will set ZIN to 1.05.

SENSDEC restores the unincremented value of each parameter at the end of the run in which it is incremented by dividing the incremented value by ZIN, so that this parameter again has the value it had in run 1. (ZIN floating point)

ØLAM: original lambda, or the unincremented value of SVAR (K-1) during run number K. floating point

XLAM: incremented lambda, or the incremented value of SVAR(K-1) after multiplication by ZIN during run number K. floating point

Note: both ØLAM and XLAM are input variables to subroutine SUMSQR.

SVAR: A dummy array of parameters to be incremented. floating point

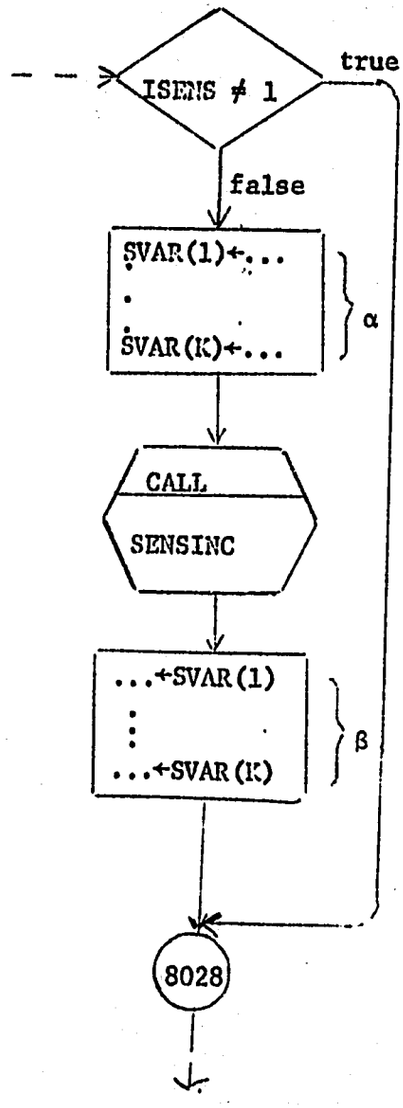
The actual values of the parameters are stored in SVAR just before the call to SENSINC.

The user must also provide for the incremented values to be stored in their proper locations after the call to SENSINC. (See the example below.)

NAME: An array of Hollerith constants used to print the names of incremented parameters. The entries of NAME are set in a Data Statement in the executive program. Name is dimensioned to 20.

EG. if the parameters chosen are B1, B2, RLTT, then the user will set DATA (NAME = 2HB1, 2HB2, 4HRLTT).

ISENS: Executive Program Switch for Subroutine SENSINC



(Isens is given its value in subroutine PARAMS.)

The user will set ISENS = 1 for a job which tests the sensitivity of the model to pre-assigned changes in parameter values

Whenever ISENS ≠ 1

the call to SENSINC is bypassed (along with "blocks" α and β of assignment statements for SVAR - if such statements are present).

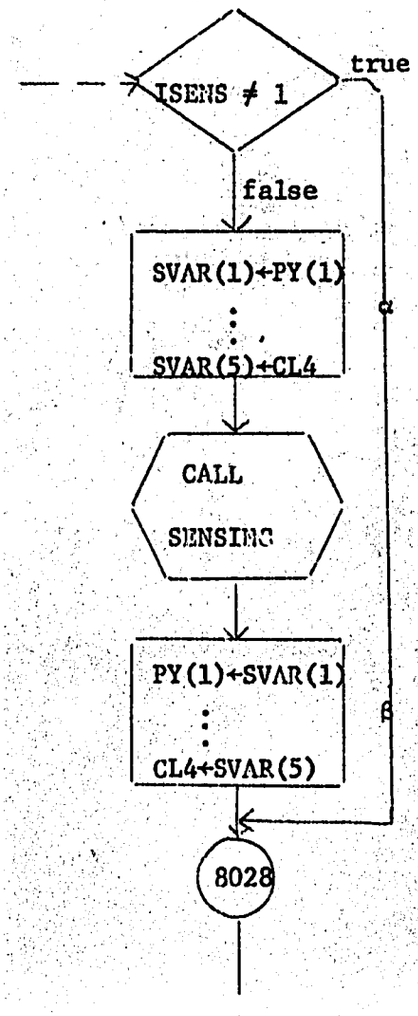
Suppose the user would like to test the sensitivity of the model to 25% increases in the values of PY(1), APLØ(1), CL1, CL2, CL4. He will set ISENS=1 and NR=6 (in subroutine PARAMS)

and DATA (ZIN = 1.25)

DATA (NAME = 5HPY(1), 7HAPLØ(1), 3HCL1, 3HCL2, 3HCL4)

in the executive program,

and he will insert blocks α and β



α { SVAR(1) = PY(1)
SVAR(2) = APLØ(2)
SVAR(3) = CL1
SVAR(4) = CL2
SVAR(5) = CL4

and β { PY(1) = SVAR(1)
APLØ(1) = SVAR(2)
CL1 = SVAR(3)
CL2 = SVAR(4)
CL4 = SVAR(5)

in the executive program.

IV. Subroutine SUMSQR

(Flow Chart pages D-1, D-2; Fortran listing page D-3)

Subroutine SUMSQR compares the values of three output variables of program NISIM with three series of actual data.

At present, PFN, ØUTP(1) and ØUTP(2) values are compared with the actual yearly data (1953-1965) for food market price in the North, groundnut production, and cotton production respectively.

IVa. Generally, the method of comparison is as follows; Subroutine SUMSQR computes:

A. the mean $AMEAN_j$ of each of the three data series

$j = 1,3$ --floating point

B. the sums SS_j $j = 1,3$

Where

$$SS_j = \sum_{IPØINT = 1}^{NPØINT} \left(\frac{\text{Simulation value}_{j,IPØINT} - \text{data value}_{j,IPØIN}}{AMEAN_j} \right)$$

(and NPØINT presently = 13) --floating point

C. the sum $TSS = \sum_{j=1}^{3} SS_j$ --floating point

D. $ØSS_j$ $j = 1,3$ and ØTSS (Ø denotes Original value)

Where

$ØSS_j = SS_j$ of run 1 $j = 1,3$

and ØTSS = TSS of run 1

to provide a basis of comparison with subsequent runs. --all floating point

E. For ISENS = 1 and M > 1

(Where M is the run number, the index of the loop DØ 22 M=1, NR)

SUMSQR computes

FIT_j $j = 1,3$ and FITT --all floating point

Where $FIT_j = \dfrac{SS_j - ØSS_j}{|XLAM - ØLAM|}$

and

$$FITT = \frac{TSS - ØTSS}{|XLAM - ØLAM|}$$

Vb. Parameter List

SUBROUTINE SUMSQR (M, ØLAM, XLAM, IFIT)

M: run number, index of the run loop

DO 22 M=1, NR in NISIM --integer

OLAM: Original Lambda, output of SENSINC, (see SENSINC description) unincremented model parameter--floating point

XLAM: incremented lambda, output of SENSINC--floating point

IFIT: local variable name for ISENS switch variable of NISIM. Only if ISENS (=IFIT) = 1 are DLAM = $|XLAM - ØLAM|$, FITT and FIT_j j=1,3 computed.

Vc. Common Blocks of SUMSQR

CØMMØN/SERS/ SERIESA, NPØINT, SIMVAL, FIT, FITT, IPØINT

CØMMØN/TEMP/ SUM, AMEAN, SS, ØSS, TSS, ØTSS, DLAM

SERIESA: floating point array dimensioned to (13,3) which holds the actual data. Presently we have

	SERIESA(K,1) ← food market price in the North	₵/lb
K=1,13	SERIESA(K,2) ← groundnut production	Klbs/yr
	SERIESA(K,3) ← cotton production	Klbs/yr

NPØINT: integer number of yearly points at which data exists - or the first dimension of SERIESA - (NPØINT presently = 13)

SIMVAL: floating point array dimensioned to (3) which holds the present Values of Simulation variables.

At present we have

SIMVAL (1) ← PFN

SIMVAL (2) ← ØUTP(1)

SIMVAL (3) ← ØUTP(2) at any time period

IPØINT: integer counter for SUMSQR. IPØINT is set to 0 at the beginning of each run and is incremented by 1 just after entry to SUMSQR.

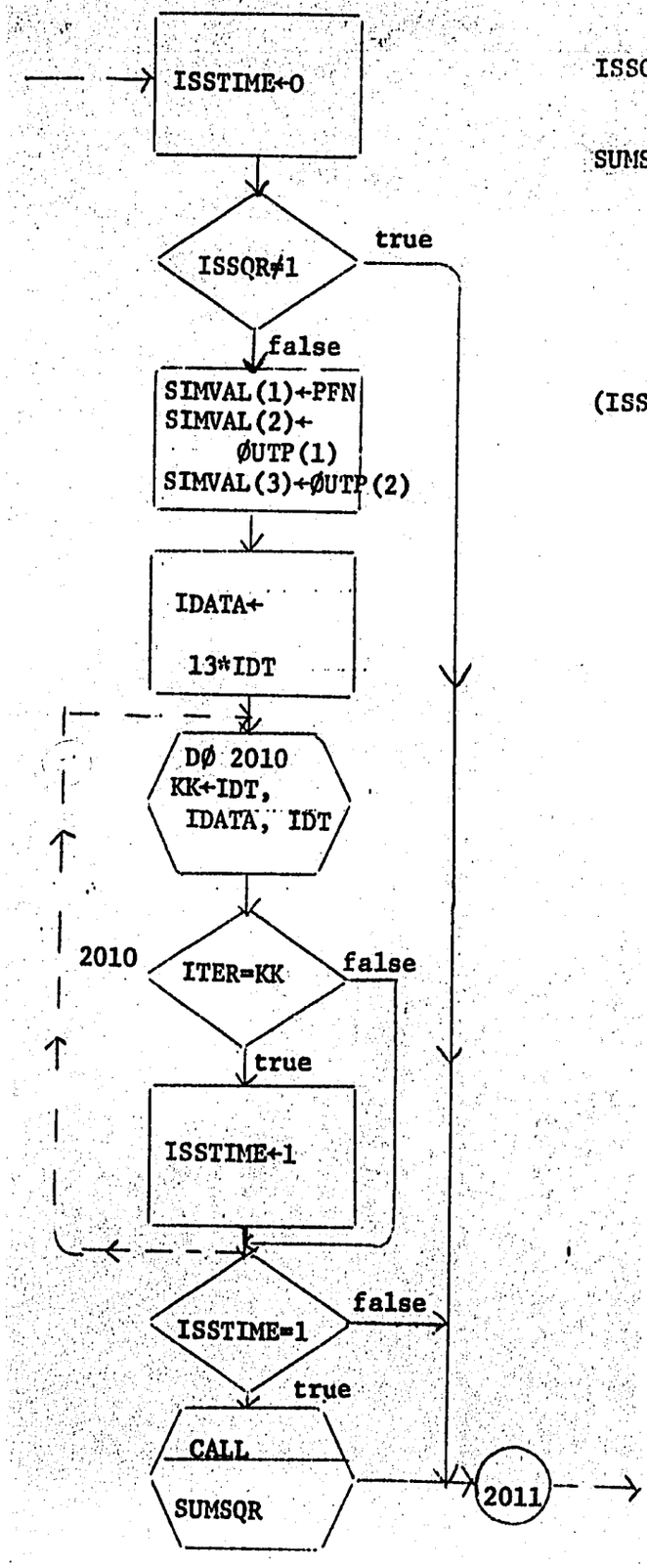
SUM: floating point array dimensioned to (3).

$$SUM_k = \sum_{j=1}^{NP\emptyset INT} SERIESA\ (j,K) \qquad K = 1,3$$

DLAM: floating point "delta lambda"

= | XLAM - ØLAM |

IVd. Call to SUMSQR in NISIM



ISSQR is given its value in subroutine PARAMS.

SUMSQR is called

if and only if

both ISSQR and ISSTIME

are equal to 1

(ISSTIME now = 1 at T = 1., 2., 3., ..., 13.)

V. Population component: Purpose of Each Subroutine

POPDAT This subroutine enters the arrays of population data by means of DATA statements. The birth and death rates are converted to fractions. The population arrays are saved in file LPSAVE on an auxiliary storage unit.

POPSET This subroutine reads the population data arrays from auxiliary storage into core memory at the beginning of each run. Certain variables and parameters which were not initialized by the data arrays are initialized at this point. If sensitivity analyses are to be made on the variables and parameters of the population component, then subroutine POPSEN is called to perform the necessary initializations. Finally, the initial values of certain population summary variables are computed at this point through a call to subroutine POPSUM.

POPSEN This subroutine performs the initializations required for the various sensitivity runs of the population component. The necessary calls to subroutine SENSINC are made at this point rather than from the main executive program.

Sensitivity analysis is performed on the following variables and parameters of the population component for various values of IPSENS:

IPSENS = 1	IPSENS = 2	IPSENS = 3
DECDTH	*POPN	DLABOR(G)
DECINM	*BIRTHR	DLABOR(C)
INMOR	*DEATHR	DLABOR(CG)
CALPP	*TLF	DLABOR(FN)
SPOIL	*AGR	RUM(M,N)
ELASFC	*EQVDAY	RUM(F,N)
ELASFP	*EQVMAN	
ELASFY		
	*arrays	

POPOUT This subroutine as presently utilized in the simulations of the agricultural sector of the Northern Region accepts from the executive routine the total nonfarm income YNFN and the price of food PFN and then returns the demand for subsistence food in each crop area $DEMR_{karea}$, the demand for food staples in the cash market DEMCFN, and the number of man-units of labor available in each crop area $LABA_{karea}$.

POPUP This subroutine updates the number of people in each cohort at the end of the next three-year major cycle (both before and after rural-urban migration has been taken into account) and computes the rate of change of nonagricultural man-units of labor, the rate of change of the total population in the agricultural sector, and the rate of change of the total population in the nonagricultural sector.

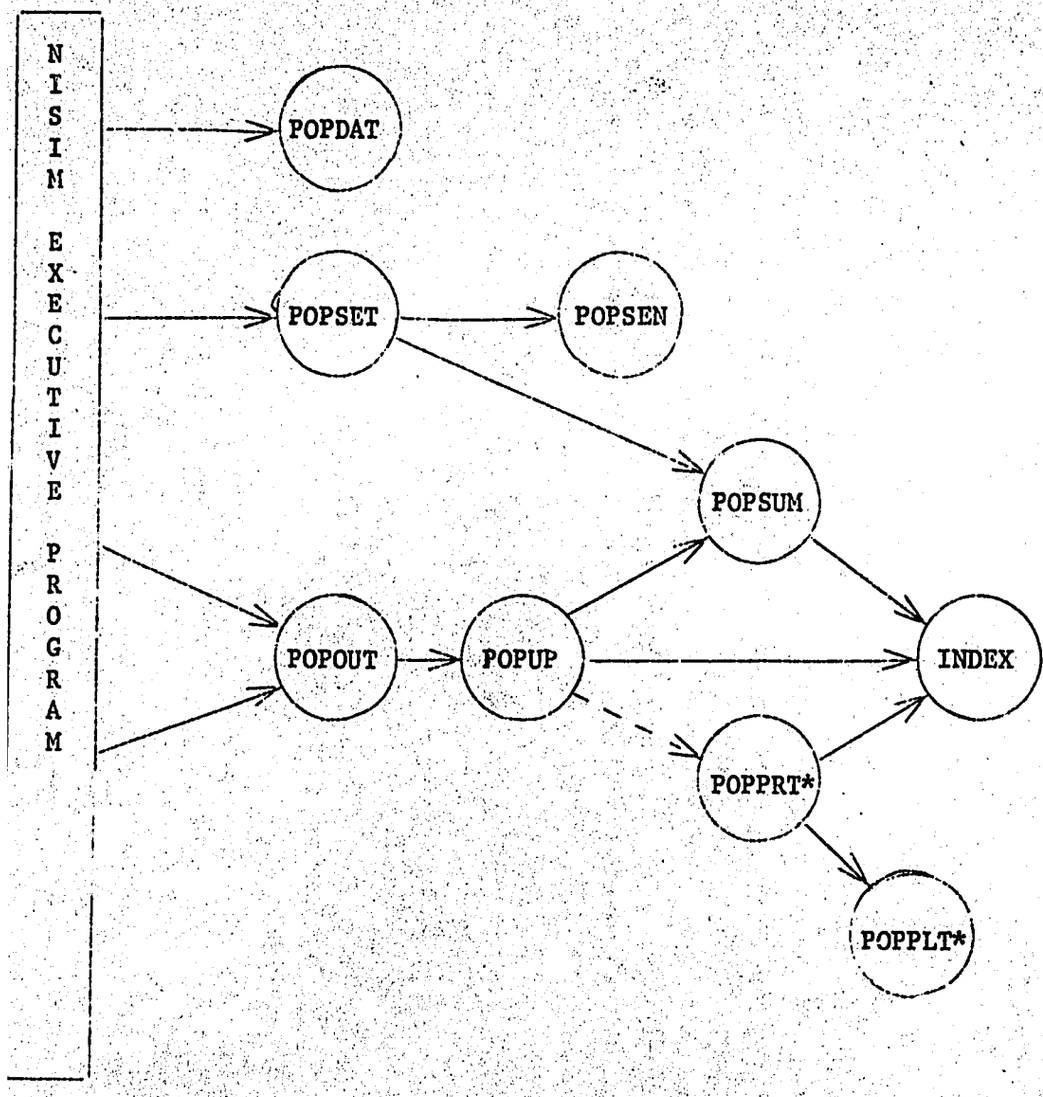
POPSUM This subroutine computes by taking into account rural-urban migration the total agricultural man-units of labor, the total nonagricultural man-units of labor, the total population in the agricultural sector, and the total population in the nonagricultural sector at the end of the next three-year major cycle.

POPPRT This subroutine prints a detailed summary of the number of people in the various cohorts of the population component.

POPPLT This subroutine plots on the line printer the age-sex distribution of the population in each region at the beginning of the simulation run and at various times during the simulation run.

INDEX This function computes the index of the cell in a single dimensioned population array as a function of subindices along the age, sex, occupation, and region dimensions.

Subroutine Linkage within Population Component



*Utilized during test runs of population component.

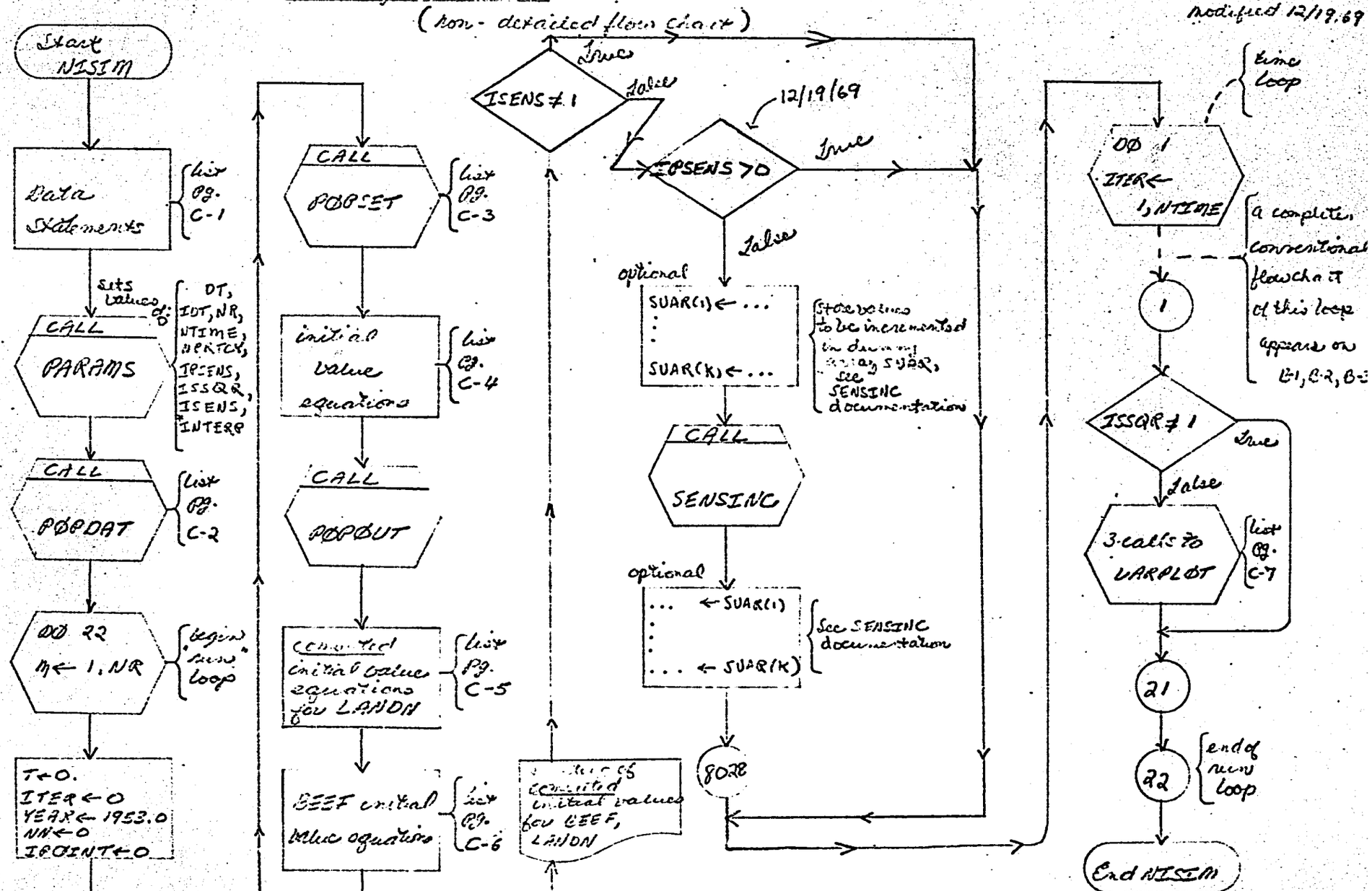
NISIM

Line Reference Chart

(non-detailed flow chart)

11/17/69

Glossary page modified 12/19/69

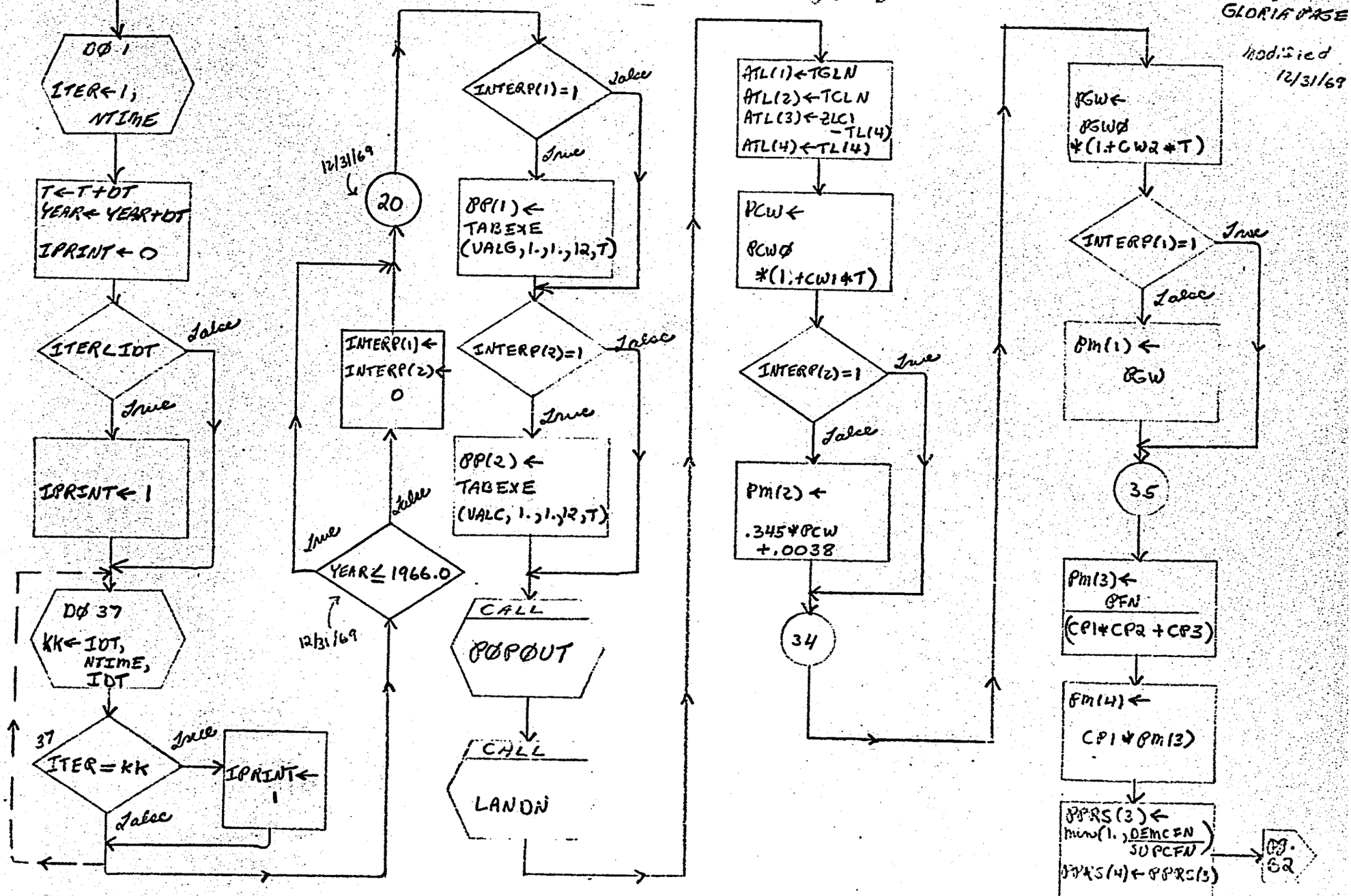


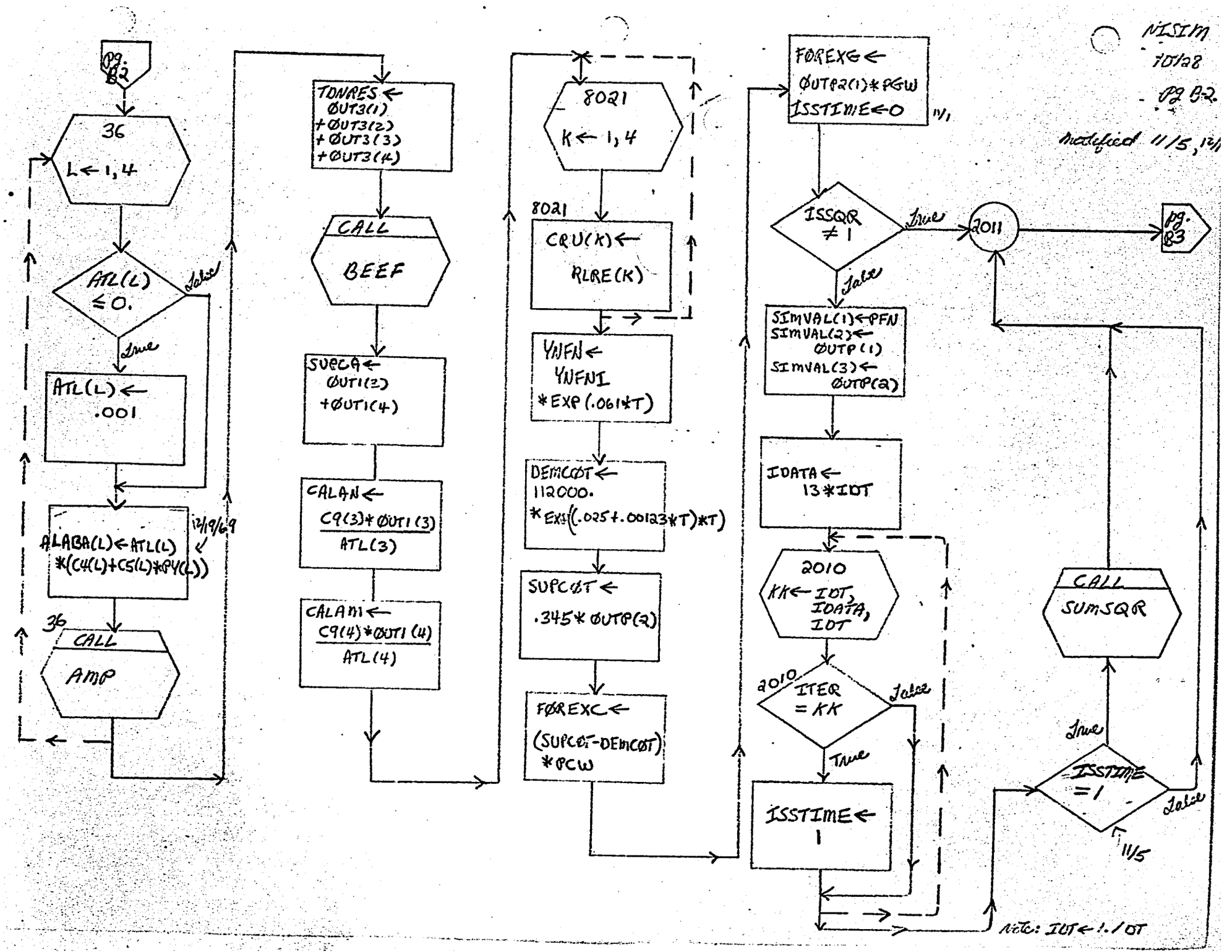
NISI.n "SIMULATION PHASE" -- from the beginning of the "time loop" to the end of program NISIM

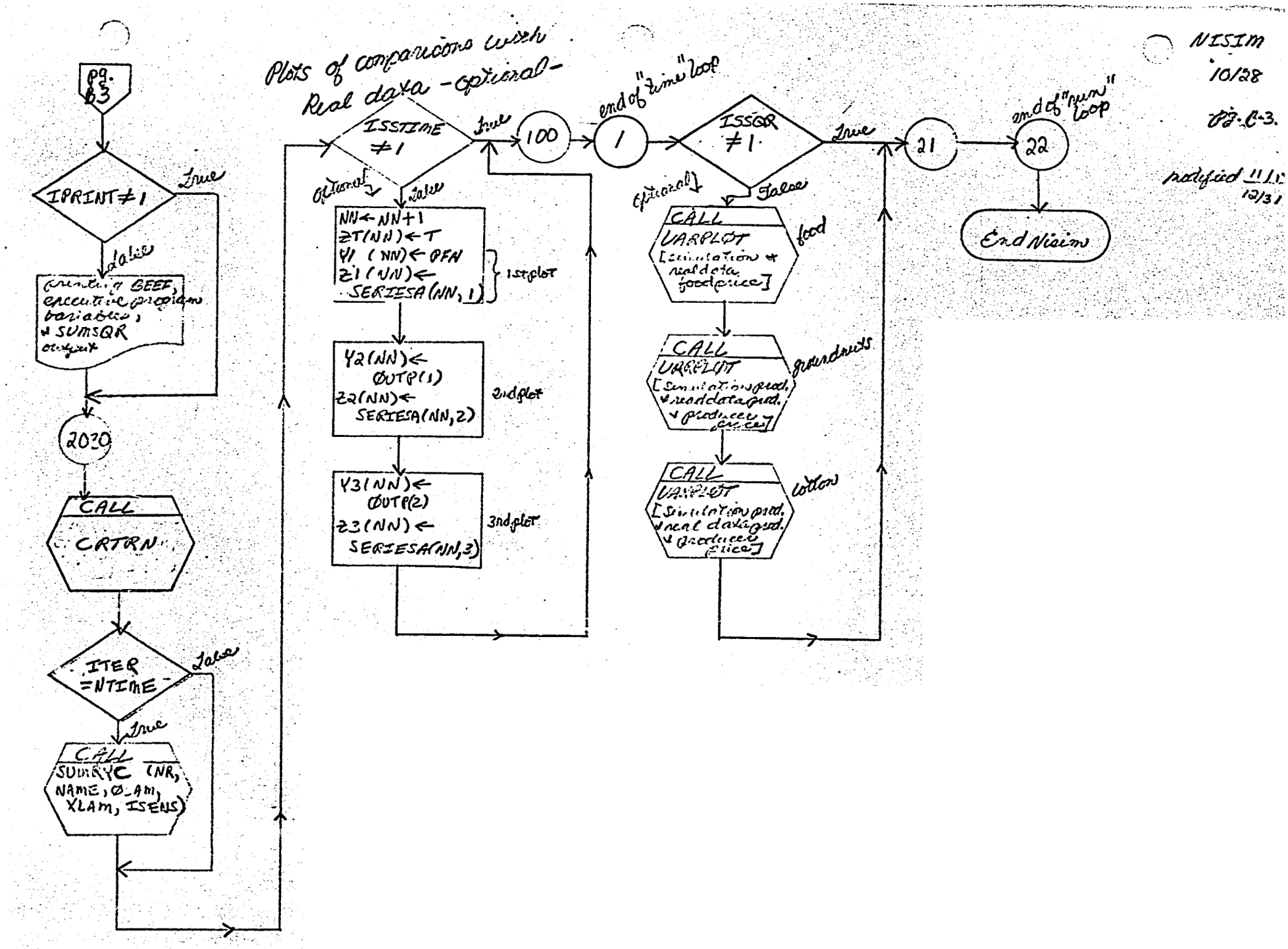
11/17/69

GLORIA PAGE

modified 12/31/69




NISIM
10/28
pg B-2
modified 11/5, 12/1
pg. B2
36
L ← 1, 4
ATL(L) ≤ 0.
false
true
ATL(L) ← .001
ALABA(L) ← ATL(L) *(C4(L)+C5(L)*PY(L))
12/9/69
36
CALL
AMP
TONRES ← ØUT3(1) + ØUT3(2) + ØUT3(3) + ØUT3(4)
CALL
BEEF
SUPCA ← ØUT1(2) + ØUT1(4)
CALAN ← C9(3)*ØUT1(3) / ATL(3)
CALAM ← C9(4)*ØUT1(4) / ATL(4)
8021
K ← 1, 4
8021
CRU(K) ← RLRE(K)
YNFN ← YNFNI *EXP(.061*T)
DEMCØT ← 112000. *EXP((.025+.00123*T)*T)
SUPCØT ← .345* ØUTP(2)
FØREXC ← (SUPCØT-DEMCØT) *PCW
FØREXG ← ØUTP2(1)*PGW
ISSTIME ← 0
11/1
ISSQR ≠ 1
true
false
2011
pg. B3
SIMVAL(1) ← PFN
SIMVAL(2) ← ØUTP(1)
SIMVAL(3) ← ØUTP(2)
IDATA ← 13*IDT
2010
KK ← IDT, IDATA, IDT
2010
ITER = KK
false
true
ISSTIME ← 1
CALL
SUMSQR
true
ISSTIME = 1
false
11/5
Note: IDT ← 1./DT


NISIM
10/28
pg. B-3.
modified 11/1
12/31
pg. B3
IPRINT ≠ 1
True
false
printing BEEF, executive program variables, & SUMSQR output
2020
CALL
CRTRN
ITER = NTIME
false
True
CALL
SUMRYC (NR, NAME, Ø_AM, XLAM, ISENS)
Plots of comparisons with Real data - optional -
ISSTIME ≠ 1
True
optional
false
NN ← NN+1
ZT(NN) ← T
Y1 (NN) ← PFN
Z1 (NN) ←
SERIESA (NN, 1)
1st plot
Y2(NN) ←
ØUTP(1)
Z2(NN) ←
SERIESA(NN, 2)
2nd plot
Y3(NN) ←
ØUTP(2)
Z3(NN) ←
SERIESA(NN, 3)
3rd plot
100
end of "time" loop
1
ISSQR ≠ 1
True
optional
false
CALL
VARPLOT
[simulation & real data food price]
food
CALL
VARPLOT
[Simulation prod. & real data prod. & producer price]
groundnuts
CALL
VARPLOT
[Simulation prod. & real data prod. & producer price]
cotton
21
end of "run" loop
22
End Nisim

List of Parameters Pre-set in Data Statements in NISIM

*denotes an array name

LUNPØP	C1		*VAL1	B21	PAØ	CP1
LPSAVE	C2		*VAL2	B22	PAD	CP2
	C3		*VAL3	B23	GRGDEL	CW1
NPØINT	C4		*VAL4	B24	FEGDEL	PCWØ
ZIN	C5	all arrays	*VAL5	B25	PFCAT	CW2
DLAM	C6		B1	B26	PFCAM	PGWØ
FITT	C7		B2		PRMT	
*FIT	C8		B3	B28	PRMM	
TSS	C9		B4	B29	YMAT	
*SS	C10			B30	YMAM	
ØTSS	WR				LGØ	
*ØSS	TAXRP		B7	B32	LGF	
	TAXRM		B8	B33	GRDEPD	
	C11		B9	B34	TDNAD	
*VALG	C12		B10	B35	D1	
*VALC	C13		B11		D2	
*SERIESA	C14		B12	B37	D3	
	C16		B13	B38	D4	
*TLM			B14	B39	D5	
	WRM			B40M	ELAS1	
*NAMEM	PPRS(1)		B16	B40T	ELAS2	
	PPRS(2)		B17		AL1	
*NAME			B18		AL3	
	*C17		B19		DI	
	*C18		B20		TAXJ	
					PCNTMA	

List of Parameters pre-set in POPDAT Subroutine

*PØPN

*PØPS

*BIRTHR

*INMØR

*DEATHR

*TLF

*AGR

*EQVDAY

*EQVMAN

*RUM

*DLABOR

*SRATIØ

DECDTH

DECINM

NØCCUP

AGRIC

NØNAG

UNEMP

MALE

FEMALE

MREGNA

MREGNZ

NAGE

DTY

NPØPCY

List of Parameters Set Back to Original Values (from T = 0. Run 1)

by Subroutine POPSET

*PØPN

*PØPS

*INMØR

*DEATHR

*BIRTHR

*CRATIO

DECDTH

DECINM

*EQUDAY

*EQUMAN

*AGR

*TLF

*RUM

*DLABOR

Values for these parameters were originally set by the call to POPDAT in run 1 (and written on a disc) and are set back each run after the first by subroutine POPSET

ELASFC

ELASFP

ELASFY

CALPP

SPØIL

Values for these parameters are set each run by sub. POPSET

List of Variables set to Initial Values in Program NISIM
(between the call to PØPSET and the first call to PØPØUT
of each run)

YFN	
YNFNI	
YNFN	*EAP
	*APLØ
*PMAR	*CL7
	*PF
*PY	*CL20
	CL21
*CM	CL22
	*CRU
CL1	*CR
CL2	*BPF
CL4	SUPCA
CL5	ZLC1
CL6	PFN
	PFNU
S3	DEMCFN
C1(3)	
C1(4)	

* denotes an array name

List of LANDN Variables Whose Initial Values are Computed in NISIM

CALAN

CALAM

*SFL

*APL

*AL

*TLCF

*TLNF

TLJC
TLJG
TLJF

CP3

* denotes an array name

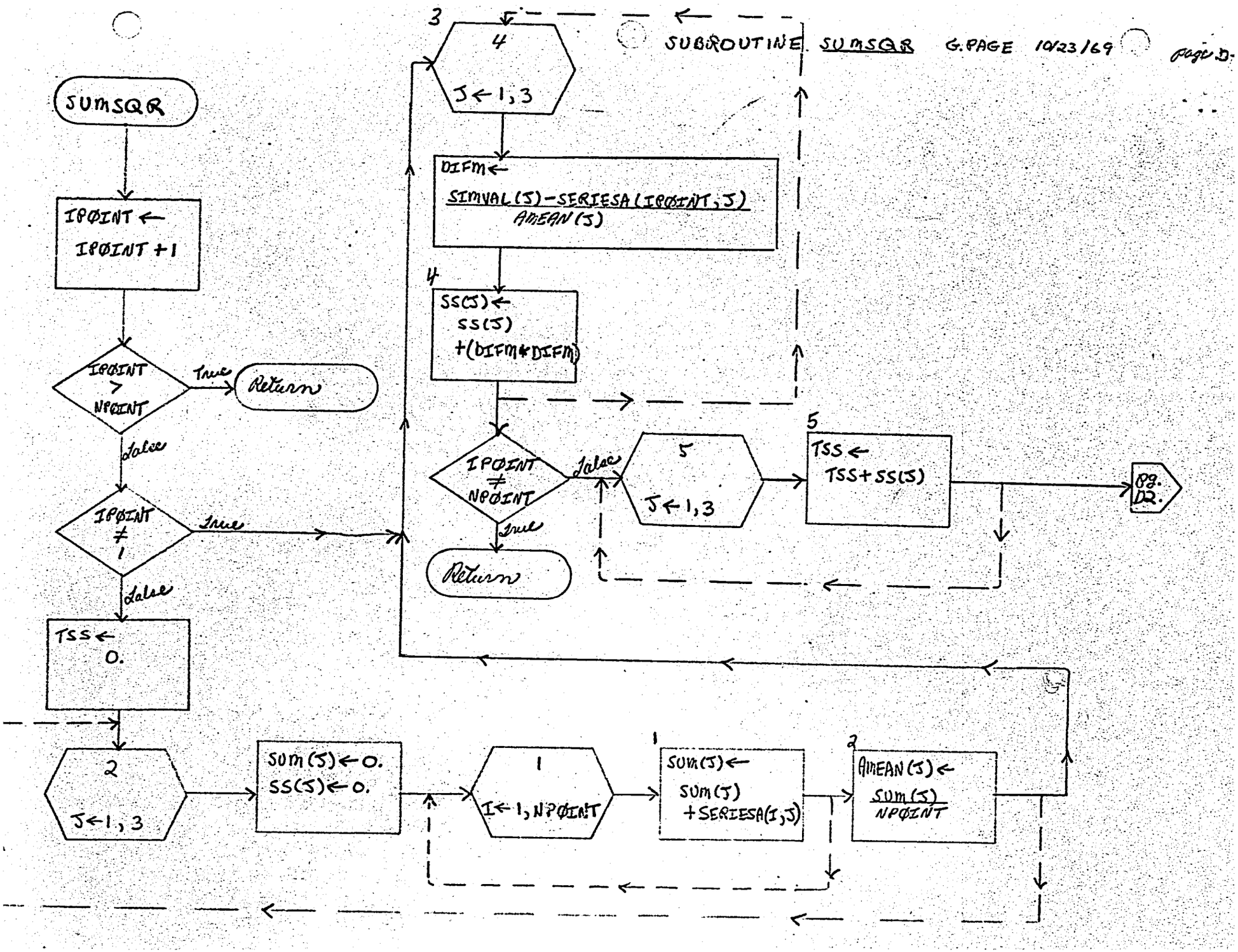
List of Beef Parameters whose Initial Values are set
(some by simple computations) in NISIM

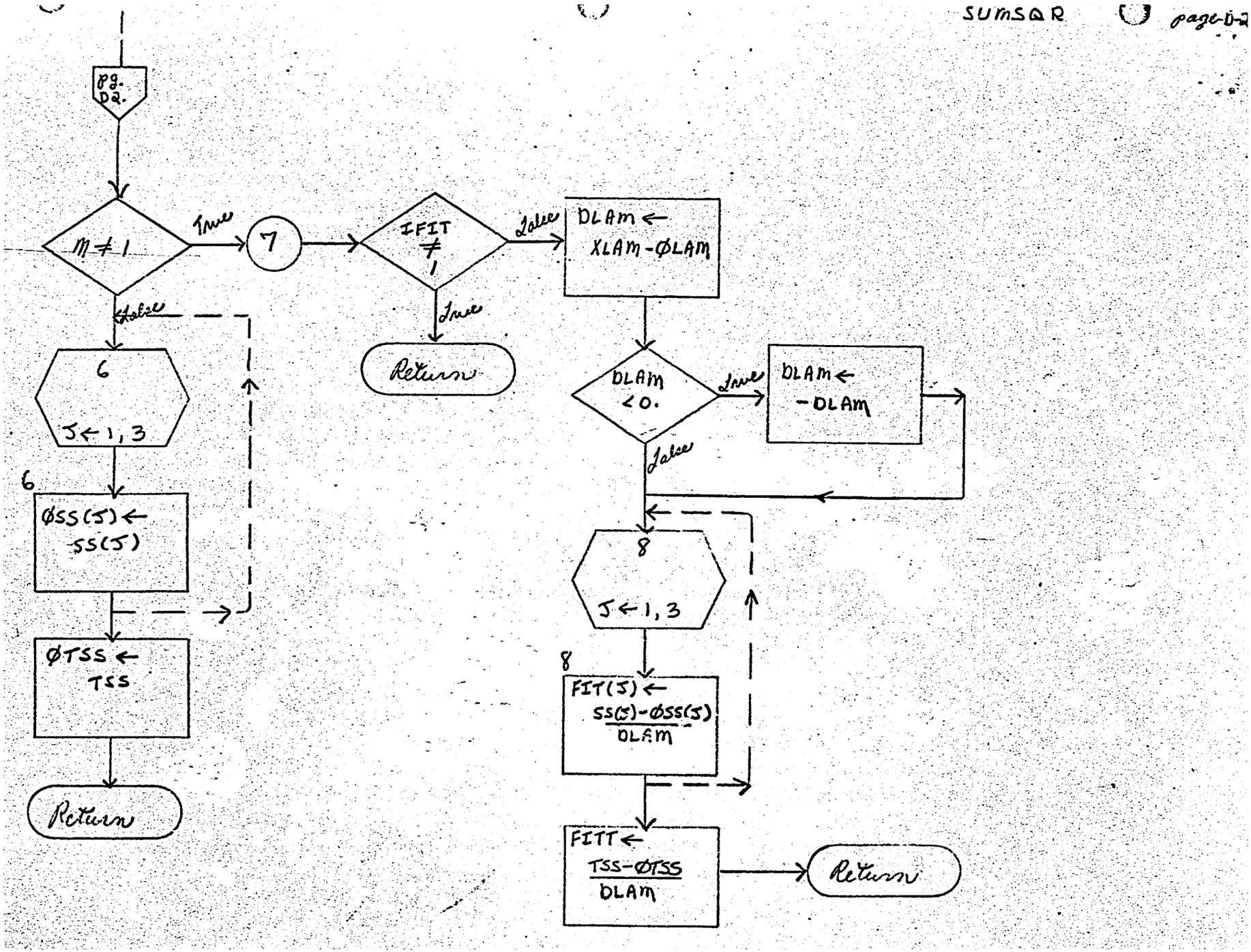
Assigned numeric values	computed
PFT	
PMT	
PFM	DFT
PMM	DMT
TDNAM	DFM
LGM	DMM
*CRØUT1	A2T
*CRØUT2	A2M
*CRØUT3	SFT
PA	TDNM
RCØN	
CF	
CF1	
A1PT	
A1PM	
ERT	
ERM	
LC3	
FARMIA	
FØREXBA	
SFM	
GRCAP	
RLTT	
GRE	

* denotes an array name

List of Plot Variables (T plotted on horizontal axis)
NN ← 1,13

plot one	PFN	
	SERIESA(NN,1)	actual data
plot two	ØUTP(1)	
	SERIESA(NN,2)	actual data
	VALG x 10^8 before plotting	
plot three	ØUTP(2)	
	SERIESA(NN,3)	actual data
	VALC x 10^7 before plotting	


SUMSQR
IPØINT ← IPØINT + 1
IPØINT > NPØINT
True
Return
False
IPØINT ≠ 1
True
False
TSS ← 0.
2
J ← 1, 3
SUM(J) ← 0.
SS(J) ← 0.
1
I ← 1, NPØINT
SUM(J) ← SUM(J) + SERIESA(I,J)
AMEAN(J) ← SUM(J) / NPØINT
3
4
J ← 1, 3
DIFM ← SIMVAL(J) − SERIESA(IPØINT, J) / AMEAN(J)
4
SS(J) ← SS(J) + (DIFM * DIFM)
IPØINT ≠ NPØINT
False
True
Return
5
J ← 1, 3
5
TSS ← TSS + SS(J)
pg. D2.


pg. D2.
M ≠ 1
True
False
7
IFIT ≠ 1
False
True
Return
DLAM ← XLAM - ØLAM
DLAM < 0.
True
DLAM ← -DLAM
False
6
J ← 1, 3
6
ØSS(J) ← SS(J)
ØTSS ← TSS
Return
8
J ← 1, 3
8
FIT(J) ← (SS(J) - ØSS(J)) / DLAM
FITT ← (TSS - ØTSS) / DLAM
Return

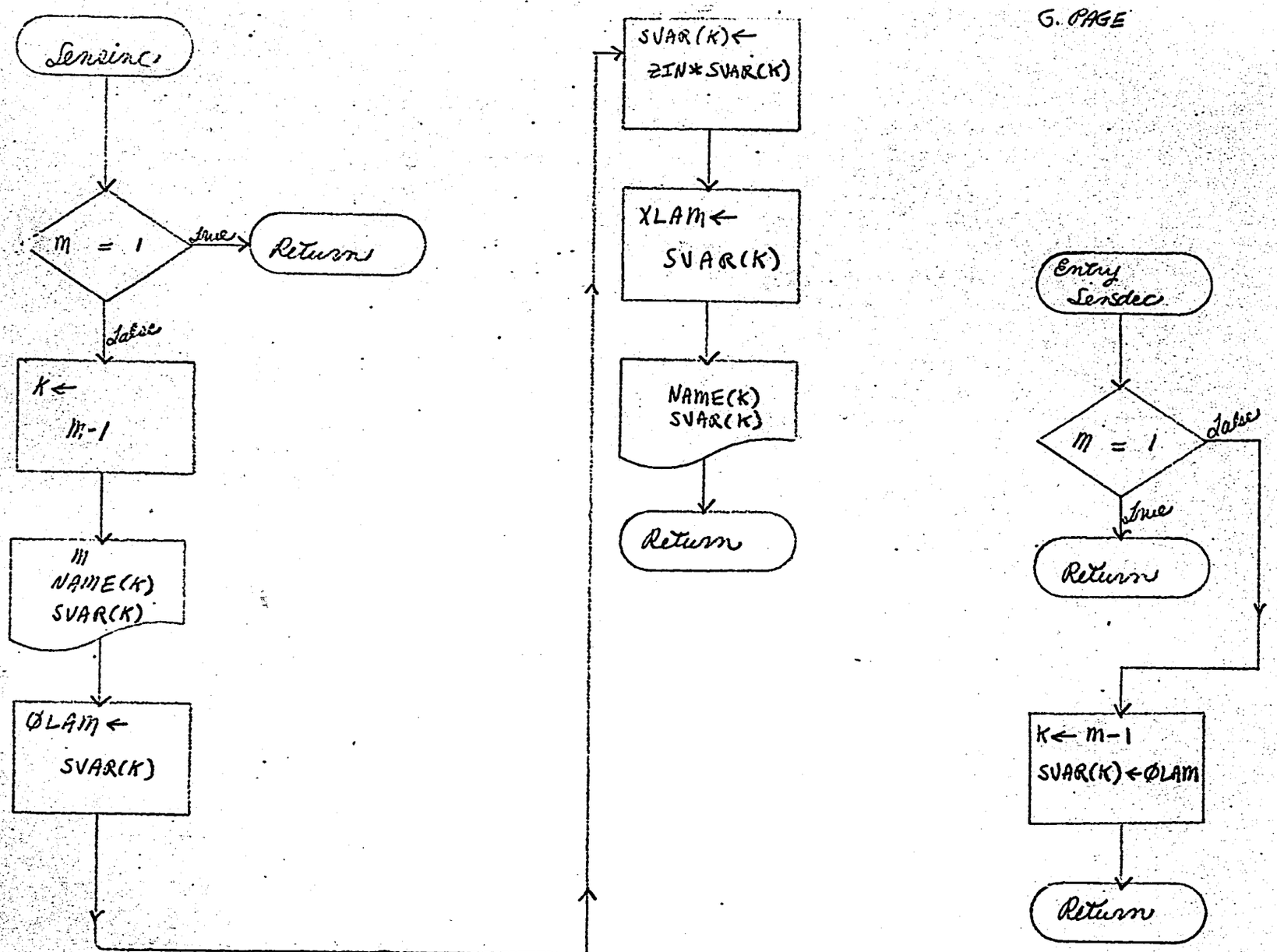
```
      SUBROUTINE SUMSQR (M,OLAM,XLAM,IFIT)
      COMMON /SERS/ SERIESA,NPOINT,SIMVAL,FIT,FITT,IPOINT
      COMMON /TEMP/ SUM,AMEAN,SS,OSS,TSS,OTSS,DLAM
      DIMENSION SUM(3),SS(3),SERIESA(13,3),AMEAN(3),SIMVAL(3),
     1 OSS(3),FIT(3)
      IPOINT=IPOINT+1
      IF(IPOINT.GT.NPOINT) RETURN
          IF(IPOINT.NE.1) GO TO 3
           TSS=0.
           DO 2 J=1,3
              SUM(J)=0.
              SS(J)=0.
              DO 1 I=1,NPOINT
1                SUM(J)=SUM(J)+SERIESA(I,J)
2             AMEAN(J)=SUM(J)/NPOINT
3     DO 4 J=1,3
        DIFM=(SIMVAL(J)-SERIESA(IPOINT,J))/AMEAN(J)
4       SS(J)=SS(J)+DIFM*DIFM
      IF(IPOINT.NE.NPOINT) RETURN
      DO 5 J=1,3
5     TSS=TSS+SS(J)
      IF(M.NE.1) GO TO 7
        OTSS=TSS
        DO 6 J=1,3
6          OSS(J)=SS(J)
           RETURN
7     CONTINUE
      IF (IFIT.NE.1) RETURN
      DLAM=XLAM-OLAM
      IF (DLAM.LT.0.) DLAM=-DLAM
      DO 8 J=1,3
8     FIT(J)=(SS(J)-OSS(J))/DLAM
      FITT=(TSS-OTSS)/DLAM
      RETURN
      END
```

10/9

10/20

Subroutine SENSINC 10/24

G. PAGE



```
      SUBROUTINE SENSINC (M,ZIN,OLAM,XLAM,SVAR,NAME)
      DIMENSION SVAR(20), NAME(20)
   60 FORMAT(*0 INCREMENTED *,A8,*=*,E10.3)
   61 FORMAT(*0 RUN NUMBER*,I3/*0 UNINCREMENTED *,A8,*=*,E10.3)
      IF(M.EQ.1) RETURN
      K=M-1
      PRINT 61, M, NAME(K), SVAR(K)
       OLAM=SVAR(K)
       SVAR(K)=ZIN*SVAR(K)
       XLAM=SVAR(K)
      PRINT 60, NAME(K), SVAR(K)
      RETURN
      ENTRY SENSDEC
      IF(M.EQ.1) RETURN
      K=M-1
      SVAR(K)=OLAM
      RETURN
      END
```

Subprogram and Function Subroutines called by NISIM

PARAMS	SUMSQR
PØPDAT	CRTRN
PØPSET	SUMRYC
PØPØUT	VARPLØT
SENSINC	
TABEXE	
LANDN	
AMP	
BEEF	

```
      PROGRAM NISIM (OUTPUT, TAPE1=OUTPUT,TAPE2=129,TAPE50=129)          NSM
C
C
C 12/23  NEW CRITERION VARIABLES  TDCRDT,  TDFERT, TNDINV, PCNNA
C 12/23  REVISED SUMMARY TABLE
C 12/19/69
C AFTER YEAR 1966  THE MODEL NO LONGER EXTRAPOLATES TO FIND PP(1) AND PP(2)
C
C
C
C  PROGRAM NISIM IS THE EXECUTIVE PROGRAM FOR A SIMULATION MODEL OF THE
C                        NIGERIAN AGRICULTURAL ECONOMY
C
C
C  DURING THE INITIALIZATION PHASE OF PROGRAM NISIM
C      I.E. BEFORE THE LOOP  DO 1 ITER=1,NTIME  BEGINS
C  THE EXECUTIVE PROGRAM CALLS
C
C            PARAMS
C            POPDAT
C            POPSET
C            POPOUT
C            SENSINC (OPTIONAL)
C
C
C  AFTER THE INITIALIZATION PHASE IS COMPLETED,
C THE EXECUTIVE PROGRAM CALLS -- ONCE DURING EACH TIME PERIOD
C
C            POPOUT
C            LANDN
C            AMP - ONCE FOR EACH OF GROUNDNUTS, COTTON, FOOD IN THE NORTH
C                  COMPETING WITH GROUNDNUTS AND/OR COTTON AND FOOD IN THE
C                  MIDDLE BELT
C            BEEF
C            SUMSOR  (OPTIONAL)
C            CRTRN
C            SUMRYCS (OPTIONAL)
C
C
C FOR SENSITIVITY JOBS  SENSINC IS CALLED ONCE AT THE BEGINNING OF EACH RUN
C
C FOR COMPARISONS  WITH REAL DATA, SUMSOR IS CALLED AFTER THE CALL TO BEEF
C AT EACH T FOR WHICH DATA EXISTS
C
C  A SUMMARY TABLE (OF CRITERION VARIABLES AND RESULTS OF COMPARISONS WITH
C    REAL DATA) IS PRINTED AT THE END OF EACH RUN OF A JOB FOR WHICH ISSOR=1
C
C     PLOTS OF REAL DATA COMPARISONS WITH SIMULATION OUTPUT
C
C  GRAPH ONE PLOTS PFN AND REAL FOOD PRICE DATA
C  GRAPH TWO PLOTS OUTP(1), REAL DATA GROUNDNUT PRODUCTION AND REAL PP(1)
C  GRAPH THREE PLOTS OUTP(2), REAL DATA COTTON PRODUCTION AND REAL PP(2)
C
```

```
C
C PP(1) AND PP(2) ARE DETERMINED BY INTERPOLATION ON YEAR Y DATA
C     (AND BY EXTRAPOLATION OUTSIDE THE DATA TIME RANGE)
C
C ARRAYS FOR SUBROUTINE LANDN (LAND ALLOCATION FOR NORTHERN NIGERIA)
C        SUBSCRIPT
C            1 REFERENCES GROUNDNUT BELT VARIABLES
C            2 REFERENCES COTTON BELT VARIABLES
C            3 REFERENCES  GROUNDNUT-COTTON JOINT AREA VARIABLES
C            4 REFERENCES MIDDLE BELT VARIABLES
C
C ARRAYS FOR SUBROUTINE AMP (PRODUCTION AND MARKETING ) FOR NORTHERN NIGERIA
C        SUBSCRIPT
C            1 REFERENCES GROUNDNUT VARIABLES
C            2 REFERENCES COTTON VARIABLES
C            3 REFERENCES VARIABLES FOR FOOD -- GRAIN CROPS - IN COMPETITION WITH
C                  GROUNDNUTS AND COTTON
C            4 REFERENCES VARIABLES FOR FOOD -- ROOT CROPS - MIDDLE BELT
C
C
C
C DECLARATIONS, GENERAL
C
      COMMON /LUNIT/ LPSAVE, LUNPOP
      COMMON /RUN/ M,NR,IPSENS
      COMMON /TIME/ T,DT,ITER ,NTIME,YEAR,NPOPCY,NPRTCY,DTY
C
C DECLARATIONS, POPOUT
C
      TYPE REAL INMOR,NAGMU
      TYPE INTEGER AGRIC,UNEMP,FEMALE
      COMMON /PSIZE/ NAGE,NOCCUP,MREGNA,MREGNZ,NAREA
      COMMON /PINDEX/ AGRIC,NONAG,UNEMP,MALE,FEMALE,MREGN
      COMMON /POPD/ POP(324),BIRTHR(27,2),SRATIO(2),INMOR(2),
     1  DEATHR(27,2,2),DECINM,DECDTH,TLF(27,2,2),AGR(27,2,2),
     2  EQVDAY(27,2),EQVMAN(27,2),DLABOR(5,2,2),RUM(2,2)
      COMMON /POPVAR/ AGMU(2,2),NAGMU(2,2),TPOPAG(2,2),TPOPNA(2,2),
     1  RAGMU(2),RNAGMU(2),RTPAG(2),RTPNA(2),TBTH(2),TDTH(2)
      COMMON /DEMAND/ ELASFC,ELASFP,ELASFY,CALPP,SPOIL
C
C DECLARATIONS, LANDN
C
      COMMON/LAND/FFLND
      COMMON /ALN1/ TL,AL,ALP,TLNF,TLCF,CR,      TLJG,TLJC,TLJF,TLM,
     1 ZLC1,SFL,DAL,XTL,R,                       RJ3,RJ2,
     2 TGLN,TCLN,SUPCFN,PFNU,PFN,DEMCFN,CALAN,CALAM
      COMMON/ALN2/ RSOM,RSUM,PF,BPF, SUPCA,DEMR,IRANK,LABA,APL,B,CM,
     1 CL1,CL2,    CL4,CL5,CL6,     S3,      CRU,      ADAL,RESID
     2 ,EAP,CL7,APLO                                                  10/22
      COMMON /ALN3/ TLMBCF,TCFLN
      DIMENSION TL(4),        APL(4),B(4),CM(4),TLM(4),AL(4),ALP(4),
     1 RSOM(4),RSUM(4),PF(4),TLNF(2),TLCF(2),CR(4),SFL(4),DEMR(4),
     2 DAL(3),XTL(2),R(2),CRU(4),ADAL(3),LABA(4),BPF(2,4),CL20(2)
```

```
      DIMENSION ZAP(4),CL7(4),APLO(4)                                   10122
C
C  DECLARATIONS FOR THE MODERNIZATION PROGRAM
C
      COMMON/MODEXT/NAMEM(14), TCAM,  JEXTN(7)
C
C DECLARATIONS, AMP
C
      COMMON /AMP1/ L, C17(4),C18(4)                                    12/17/6
      REAL LABA,INCP,INCA,INMY
      DIMENSION C1(4),C2(4),C3(4),C4(4),C5(4),C7(4),C8(4),C9(4),C6(4),
     1    C10(4),          PY(4),WR(4),TAXRP(4),TAXRM(4),PMAR(4),PM(4),
     2    C11(4),C12(4),C13(4),C14(4),                ALABA(4),PFTP(4)
      DIMENSION VAP(4),YLDD(4),EMP(4),DEML(4),WAGES(4),YLD(4),YLD1(4),
     1    YLD2(4),YLD3(4),VAM(4),INCP(4),TAXP(4),OUTP(4),OUT1(4),PP(4),
     2    OUT2(4),OUT3(4),INCA(4),TAXM(4),CNFI(4),EMPM(4),WAGESM(4),
     3    INMY(4),RLRE (4),OUTP1(4),OUTP2(4),OUTP3(4),PFTM(4),ATL(4)
      DIMENSION YLDW(4),SUBI(4),        TAXG(4),   FEGRI(4),FFCI(4),
     1 RLND(4),C16(4),PPRS(4)
C
C  DECLARATIONS FOR THE BEEF SECTOR
C
      REAL LGF,LG,LGO,     LC3,LGM,LGT
      DIMENSION VAL1(9),VAL2(9),VAL3(9),VAL4(9),CROUT1(3),CROUT2(3),
     1  VAL5(9),CROUT3(3)
      COMMON /BF1/PFT,PFM,SUP,RCON,          FOREXBA,  EFE,EXGR,CF,
     1  YM,        PMT,PMM,DEM, ERM,     RLC3, LGO,     SFT,SFM,YA,
     2          FARMIA,SRT,SRM,ERT,LGM,LG,RLTT,SMT,SMM,
     3            TDNAT,       TDNAM,LGT,COGR,ANPROT,VALCAP
      COMMON /BF2/ B1,B2,B3,B4,        B7,B8,B9,B10,B11,B12,B13,B14,      M
     1      B16,         B28,B29,B30,         B33,B34,B35,B36,B37/,      M
     2  B40M,B40T,D1,D2,D3,D4,D5,AL1,      AL3,     ELAS1,ELAS2          M
      COMMON /BF3/ CROUT1,CROUT2,CROUT3,VAL1,VAL2,VAL3,VAL4,VAL5,
     1        GRGDEL,FEGDEL,            PAD,GRDEPD,DI,          LC3,LGF,
     2        PA,PAO,TDNAD,DFM,DFT,GRE,PFCAT,PFCAM,YMAT,YMAM,
     3  PRMT,PRMM,CF1,GRCAP
      COMMON/BF4/  BRT,DRT,DMT,ERPT,A1PT,A1T,A2T,A3T,A4T,BRDELT,BRM,DRM,  C1
     1  DMM,ERPM,A1PM,A1M,A2M,A3M,A4M,BRDELM,CIMP,AUX1,AUX2,AUX3,AUX4,    C2
     2      GRT,TDNGT,TDNRES,TDNT,TDNGM,TDNFC,TDNM,B17,B18,B19,B20,B21,
     3  B22,B23,B24,B25,B26,TAXJ,PCNTMA
C
C  PRINT AND PLOT DECLARATIONS
      COMMON/PRNT/ IPRINT
      DIMENSION IGRAPH(101,56)
      DIMENSION Y1(13), Y2(13),Y3(13), Z1(13),Z2(13),ZT(13),Z3(13)
C
C DECLARATIONS FOR THE CRITERION SUBROUTINE,  CRTRN
      COMMON /BFCRT/ TAXA,FARMI,FOREXB,FOREXC,FOREXG
C
C DECLARATIONS FOR SUMSQR, SENSINC, AND PP INTERPOLATION FOR GR. AND COT.
      COMMON /SERS/ SERIESA,NPOINT,SIMVAL,FIT,FITT,IPOINT
      COMMON /TEMP/ SUM,AMEAN,SS,OSS,TSS,OTSS,DLAM
      COMMON /SENSIT/ NAME(20),OLAM,XLAM
```

```
      DIMENSION SUM(3),SS(3),SERIESA(13,3),AMEAN(3),SIMVAL(3),
     1 OSS(3),FIT(3)
      DIMENSION SVAR(20)
      DIMENSION VALG(13), VALC(13)
      DIMENSION INTERP(4)
C
C          INITIALIZATION                INITIALIZATION                INITIALIZATION
C          INITIALIZATION                INITIALIZATION                INITIALIZATION
C
C  DATA STATEMENTS  *      *      *      *      *
C
      DATA (LUNPOP=1)
      DATA (LPSAVE=2)
C
      DATA (NPOINT=13), (ZIN=1,20)
      DATA (DLAM=0.), (FITT=0.), (FIT=0.,0.,0.)
      DATA (TSS=0.), (SS=0.,0.,0.),(OTSS=0.),(OSS=0.,0.,0.)
C
C  ACTUAL DATA SERIES FOR 1953-1965
C
      DATA (VALG=.01673,.01668,.01532,.01560,.01532,.01682,.01577,
     1 .01636,.01618,.01495,.01295,.01336,.01300)
      DATA(VALC=.02583,.02426,.02466,.02547,.02546,.02434,.02531,
     1 .02624,.02438,.02311,.02046,.02081,.02047)
      DATA (SERIESA=.007291,.010000,.007691,.008123,.008214,.008900,
     1 .008541,.007400,.007804,.009527,.008654,.006886,.007691,
     2    947531.,947802.,820107.,1166482.,787450.,1572336.,1173379.,
     3 979970.,1361912.,1513569.,1917340.,1728890.,1486945.,
     4   112207.,169033.,217059.,177740.,160492.,272593.,192223.,
     5 188962.,342654.,186483.,322318.,286251.,286117.)
C
C LANDN DATA
      DATA (TLM=18000., 10000., 18108., 36000.)                    10/21
C
C OUTPUT LABELS FOR THE MODERNIZATION PROGRAM
      DATA (NAMEM =        20HGROUNDNUTS          ,
     1                     20HCOTTON              ,
     2                     20HFOOD  NORTHERN NORTH,
     3                     20HFOOD MIDDLE BELT    ,
     4                     20HMECHANIZATION       ,
     5                     20HBEEF .?.            ,
     6                     20HBEEF .?.            )
C
C
C TEMPORARY          TEMPORARY          TEMPORARY          TEMPORARY
C TEMPORARY          TEMPORARY          TEMPORARY          TEMPORARY
C TEMPORARY UNTIL MODERNIZATION PROGRAM IS OPERATING
      DATA(JEXTN=0,0,0,0,0,0,0)
C
C DATA FOR AMP NORTH -- VALUES GIVEN ARE FOR GROUNDNUTS, COTION,
C        FOOD IN FLY-FREE NORTH, AND FOOD IN THE MIDDLE BELT - IN THAT ORDER
C
      DATA (C1=300.,0.,1870., 546.),(C2=.035, .67, .14, .14),      8/28
```

```
     1 (C3=183.,183.,393.,393.), (C4=.11,.08,.18,.18),                  8/18
     2 (C5=.00016,.0003,.000052,.000052),(C6=.3,.7,.7,.3),              8/19
     3 (C7=.90, .90, .90, .90), (C8=.9, 0., .9, .9)
      DATA (C9=.85,.85,.85,.85),(C10=.000016,.000067,.00022,.00022)
     1,(WR=60.,60.,60.,60.), (TAXRP=.2, .2, .2, .2),
     2 (TAXRM=.2,.2,.2,.2),        (C11=0.,0.,1.,1.)
      DATA(C12=.7, .1, 0., 0.), (C13=.3, .9, 0., 0.),
     1 (C14=630.,0.,860.,860.),(C16=0.,0.,0.,0.),(WRM=60.)              8/28
      DATA (PPRS=1.,1.)
      DATA (C17=.00001,.00001,.00001,.00001),(C18=.17,.17,.17,.17)  12/17/6
C
C
C  BEEF DATA   3600 AND 6500
C
      DATA (VAL1=.06,.19,.27,.33), (VAL2=.08,.29,.44,.54),
     1 (VAL3=.55,.3,.22,.2,.14,.1,.1),                                  8/19
     2 (VAL4=.5,.17,.08 ,.06, .05, .05, .05), (VAL5=1.,1.)              8/19
      DATA (B1=4.25),(B2=5.75),(B11=1.15),(B14=.036),
     1 (B16=.000333),              (B28=1.5), (B29=60.), (B30=.035),
     2              (B32=.0056),(B33=0.),(B34=0.), (B36=.1), (B37=.285),
     3 (B38=.1), (B39=0.), (B40M=.6), (B40T=.6)
      DATA (D1=1.),(D2=3.), (D3=1.),(D4=1.), (D5=3.), (ELAS1=1.),
     1 (ELAS2=1.), (AL1=.029),              (AL3=.06),
     2 (DI=850.),(PAD=17.9),(PAD=37),(GRGDEL=3.),(FEGDEL=3.),
     3 (PFCAT=.5 ), (PFCAM=.65), (PRMT=.01), (PRMM=.01), (YMAT=950.),   8/19
     4 (YMAM=1200.),(LGO=47000.),                (LGF=74000.),          8/18
     5 (GRDEPD=10.), (TDNAD=3000.)
      DATA (B3=156.),(B4=.003),(B7=241.),(B8=.2),(B9=500.),
     1  (B10=3700.), (B12=0.),  (B13=0.),  (B17=0.),  (B18=0.),
     2 (B19=0.),(B20=.023),(B21=.29),(B22=.47),(B23=.3),
     3  (B24=.6), (B25=.57), (B26=.7), (B35=.208)
      DATA (TAXJ=.4333),(PCNTMA=.77)
C
C PRICE DATA
      DATA (CP1=.74), (CP2=.535)                                        8/28
      DATA (CW1=0.),(PCWO=.086),(CW2=0.),(PGWO=.023)                    10/20
C
C
C
C  HERE BEGINS THE EXECUTABLE PORTION OF PROGRAM NISIM
C
C
C
C          INITIALIZATION              INITIALIZATION              INITIALIZATION
C          INITIALIZATION              INITIALIZATION              INITIALIZATION
C
C
CCC  SET RUN PARAMETERS
      CALL PARAMS        (DT,NR,NTIME,NPRTCY,ISENS,ISSQR,INTERP,
     1  IPSENS,IDT)
C
C POPULATION ARRAY INITIALIZATION
      CALL POPDAT
```

```
C
C
C RUN LOOP
C
      DO 22 M=1,NR
C
C
      T=0.
      ITER=0
      YEAR=1953.0
      NN=0
      IPOINT=0
C
C RESET POPULATION ARRAYS AND PARAMETERS FOR EACH RUN
      CALL POPSET
C
C TEMPORARY FARM INCOME
      YFN=200000.
C
C INITAL NONFARM INCOME
      YNFNI=97250.
      YNFN=YNFNI
C
C POLICY VARIABLES
      PMAR(1)=.3
      PMAR(2)=.2
      PMAR(3)=.1
      PMAR(4)=.1
C
C TEMPORARY UNTIL WE HAVE AVERAGE PY FROM THE MODERNIZATION SECTOR
      PY(1)=700.
      PY(2)=260.
      PY(3)=700.
      PY(4)=5320.
C
C FOLLOWING PARAMETERS SET BACK EACH RUN FOR SENSITIVITY JOBS
      CM(1)=CM(2)=CM(3)=  1.
      CM(4)=1.
      CL1=.5
      CL2=.5
      CL4=1.
      CL5=1.
      CL6=.5
      S3=1.
      C1(3)=1870.
      C1(4)=546.
C
C  INITIAL AND CONSTANT VALUES, SUBROUTINE LANDN
C
      EAP(1)=.75
      EAP(2)=.5
      EAP(3)=.75
      EAP(4)=.6
```

```
      APLO(1)=6.0
      APLO(2)=6.0
      APLO(3)=6.0
      APLO(4)=5.3
      CL7(1)=.05
      CL7(2)=.1
      CL7(3)=.05
      CL7(4)=.03
      PF(1)=1.
      PF(2)=1.
      PF(3)=1.
      PF(4)=1.
      CL20(1)=0.
      CL20(2)=.25
      CL21=0.
      CL22=0.
      CRU(1)=44.3
      CRU(2)=35.5
      CRU(3)=17.1
      CRU(4)=43.
C
      DO 8022 K=1,4
        BPF(1,K)=1.5
        BPF(2,K)=.5
 8022 CR(K)=CRU(K)
      SUPCA=18393940.
      ZLC1= 15240.
      PFN=.0073
      PFNU=PFN
      DEMCFN=2800000000.
C
C
C
      CALL POPOUT (YNFN,     PFN,DEMR,DEMCFN,LABA)
C
C
C
C  THE FOLLOWING INITIAL VALUE EQUATIONS FOR CALAN, CALAM ARE CORRECT
C    AS LONG AS CM(K).GE.1.  AND LARA(K)=DEML(K)    K=1,4
      CALAN=C9(3)*C8(3)*C1(3)*PY(3)*CM(3)
      CALAM=C9(4)*C8(4)*C1(4)*PY(4)*CM(4)
C
      DO 8026  K=1,3
 8026 SFL(K)=DEMR(K)/CALAN
      SFL(4)=DEMR(4)/CALAM
      DO 8027  K=1,4
      APL(K)=AMAX1(APLO(K)*CM(K) -SFL(K)/LABA(K), 0.)
 8027 AL(K)=LABA(K)*EAP(K)*APL(K)
      DO 8025  K=1,2
       TLCF(K)=CL20(K)*AL(K)
 8025  TLNF(K)=(1.-CL20(K))*AL(K)
C
      TLJC=CL21*AL(3)
```

```
      TLJF=CL22*AL(3)
      TLJG=(1.-CL21-CL22)*AL(3)
C PRINT LANDN INITIAL VALUES (INDIRECTLY SET) WHICH ARE ALTERED
C     BEFORE THE FIRST OUTPUT AT T=DT
      PRINT 8012,M
      PRINT 8013, AL,SFL,TLJG,TLJC
      PRINT 8014, APL,TLNF,TLCF,TLJF
C
        CP3=1.-CP2
C
C INITIAL CONDITIONS, BEEF
      PFT=5100.
      PMT=2400.
      PFM=36.5
      PMM=13.5
      TDNAM=5260.
      LGM=1000.
      DO 6010    LL=1,3
        CROUT1(LL)=0.
        CROUT2(LL)=0.
 6010   CROUT3(LL)=0.
      PA=17.9
      RCON=1.
      CF=0.
      CF1=0.
      A1PT=0.
      A1PM=0.
      ERT=.08
      ERM=.2
      LC3=0.
      FARMIA=0.
      FOREXBA=0.
      SFM=0.
      GRCAP=0.
      RLTT=0.
      GRE=8.
C
C INITIAL VALUES FOR THE BEEF SECTOR WHICH ARE NOT SET DIRECTLY, AND WHICH
C   THEREFORE WILL BE PRINTED BEFORE THE TIME LOOPS BEGIN
C
      DFT=.1*PFT
      DMT=.1*B40T*PMT
      DFM=.07*PFM
      DMM=.07*PMM
      A2T=.25*PFT
      A2M=.41*PFM
      SFT=B20*PFT
      TDNM=B9*LGM
C
      PRINT 6007
      PRINT 6008, DFT,DMT,DFM,DMM,A2T,A2M,SFT,TDNM
C
C
```

```
C
C ISENS IS THE SWITCH FOR THE CALL TO SENSINC
C    IF ISENS=1 THE CALL WILL BE EXECUTED
C      EXCEPT WHEN PERFORMING SENSITIVITY ANALYSES ON THE POPULATION
C      COMPONENT (IPSENS.GT.0), IN WHICH CASE SENSINC IS CALLED FROM
C      POPSET.
C
      IF (ISENS.NE.1) GO TO 8028
      IF (IPSENS.GT.0) GO TO 8028
C SET PARAMETERS FOR SENSINC
                        CALL SENSINC (M,ZIN,OLAM,XLAM,SVAR,NAME)
C
 8028 CONTINUE
C
C
C BEGIN TIME DO-LOOP IN WHICH T ACCUMULATES TO THE RUN LENGTH (=DT*NTIME
C  BEGIN TIME DO-LOOP IN WHICH T ACCUMULATES TO THE RUN LENGTH (=DT*NTIME
C
C
      DO 1 ITER=1,NTIME
C
      T=T+DT
      YEAR=YEAR+DT
C
C SET PRINT SWITCH
      IPRINT=0
       IF(ITER.LT.IDT) IPRINT=1
         DO 37 KK=IDT,NTIME,IDT
   37        IF(ITER.EQ.KK) IPRINT=1
C
C
C AFTER YEAR 1966 THE MODEL NO LONGER EXTRAPOLATES TO FIND PP(1) AND PP(
      IF (YEAR.LE. 1966.) GO TO 20
         INTERP(1)= INTERP(2)= 0
   20 CONTINUE
C
C INTERPOLATE  TO GET GROUNDNUT AND COTTON PRODUCER PRICES AT EACH POINT
C    IN TIME FROM THE YEARLY PRICES GIVEN
C BYPASS PP COMPUTATION IN AMP FOR GROUNDNUTS AND COTTON WHEN WE
C  INTERPOLATE FROM ACTUAL DATA  (INTERP=1)
      IF(INTERP(1).EQ.1) PP(1)=TABEXE(VALG,1.,1.,12,T)
      IF(INTERP(2).EQ.1) PP(2)=TABEXE(VALC,1.,1.,12,T)
C
C PERFORM POPULATION CALCULATIONS FOR NORTH ONLY
      CALL POPOUT (YNFN,    PFN,DEMR,DEMCFN,LABA)
C
C
 8020 CALL LANDN
C
      IF (IPRINT.EQ.1) CALL LANDNP
C
C ATL CONTAINS- TOTAL GROUNDNUT LAND IN NORTHERN NIGERIA          = ATL(1)
C (IN K ACRES)          TOTAL COTTON LAND IN NORTHERN NIGERIA     = ATL(2)
```

```
C                          TOTAL FOOD LAND NOT IN THE MIDDLE BELT     = ATL(3)
C                          TOTAL FOOD LAND IN THE MIDDLE BELT         = ATL(4)
      ATL(1)=TGLN
      ATL(2)=TCLN
      ATL(3)=ZLC1-TL(4)
      ATL(4)=TL(4)
C
C GENERATE COTTON PRICE
C
      PCW=PCWO*(1.+CW1*T)
      IF(INTERP(2).EQ.1) GO TO 34
      PM(2)=.345*PCW + .0038
   34 CONTINUE
C
C GENERATE GROUNDNUT PRICE
C
      PGW=PGWO*(1.+CW2*T)
      IF(INTERP(1).EQ.1) GO TO 35
      PM(1)=PGW
   35 CONTINUE
C
C COMPUTE PM(3) --PRICE FOR GRAIN FOOD AND PM(4) -- PRICE FOR ROOT CROP FOOD
C
      PM(3)=PFN/(CP1*CP2 + CP3)                                          8/28
      PM(4)=CP1*PM(3)                                                    8/28
C
C COMPUTE THE PROPORTION OF FOOD SOLD
      PPRS(3)=AMIN1(DEMCFN/SUPCFN, 1.)
      PPRS(4)=PPRS(3)
C
C
C DO LOOP FOR THE FOUR NORTHERN AMP CALLS
      DO 36 L=1,4
C
C
C  PREVENT CHAOS IN AMP RESULTING FROM ZERO CROP LAND
      IF( ATL(L).LE. 0.)  ATL(L)=.001
C
C SET ALABA(K)=THE EXPRESSION FOR DEMI(K)  (FROM AMP)      K=1,4
            ALABA(L)= ATL(L)    *(C4(L)+C5(L)*PY(L))                   12/18/69
C
   36 CALL AMP (ATL(L),PY(L),ALABA(L),WR(L),TAXRP(L),TAXRM(L),PMAR(L), 12/18/69
     1 PM(L),                 VAP(L),C1(L),C2(L),C3(L),C4(L),C5(L),
     2 C6(L),C7(L),C8(L),C9(L),YLDD(L),EMP(L),DEML(L),WAGES(L),YLD(L), 82
     3 YLD1(L),VAM(L),YLD2(L),YLD3(L),INCP(L),TAXP(L),OUTP(L),OUT1(L), 83
     4 OUT2(L),OUT3(L),INCA(L),TAXM(L),C10(L),CM (L),CNFI(L),EMPM(L),  84
     5 WAGESM(L),WRM,    INMY(L),RLRE (L),OUTP1(L),OUTP2(L),OUTP3(L),  85
     6 C11(L),C12(L),C13(L),C14(L),PP (L),PETP(L),PETM(L),C16(L),
     7 FFGRI(L),FFCI(L),RLND(L),TAXG(L),YLDW(L),SUBI(L),INTERP(L),
     8 PPRS(L))
C
C SUPPLY TDN FROM CROP RESIDUES IN FLY-FREE AREA TO SUB. BEEF
C
```

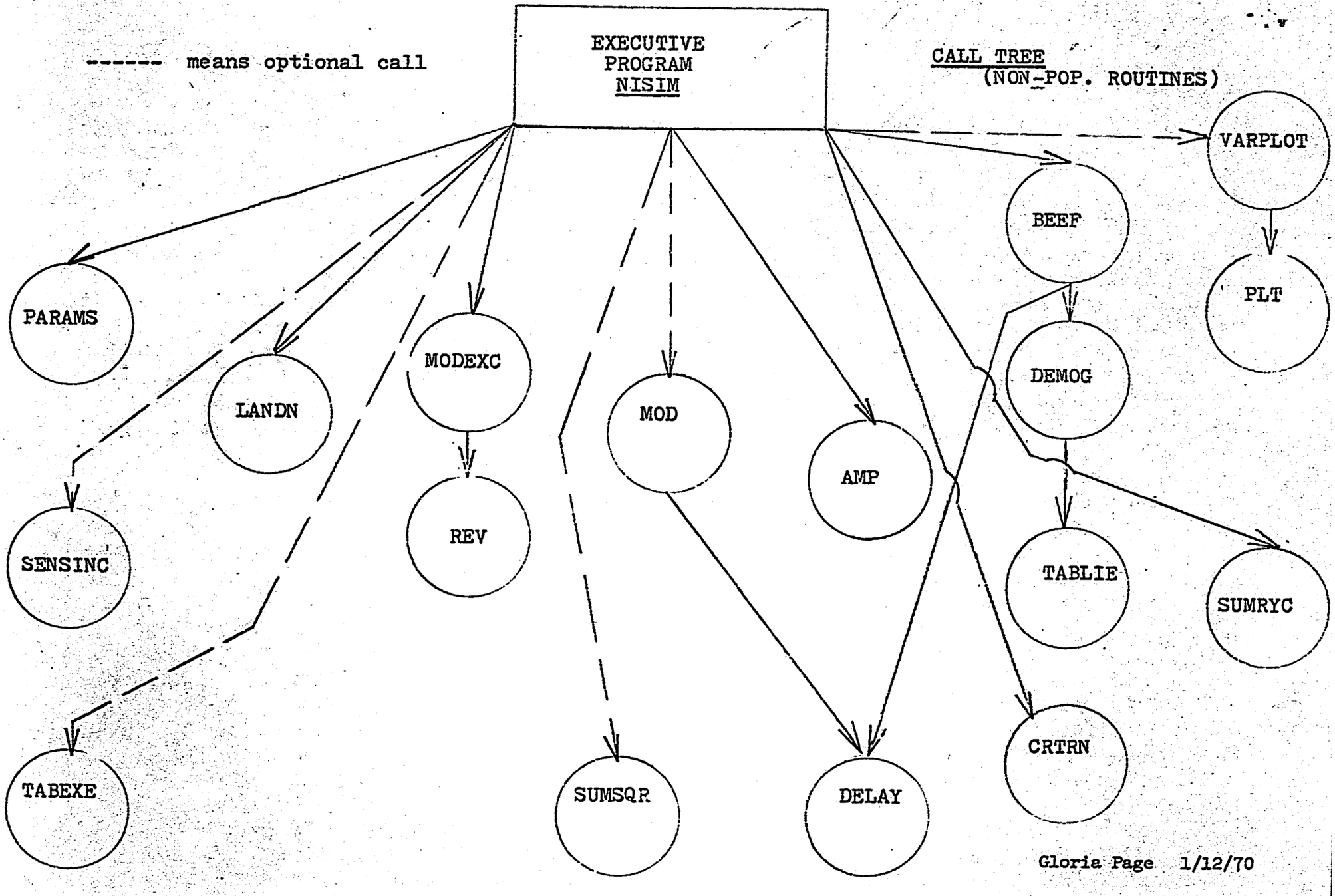
```
      TDNRES=OUT3(1)+OUT3(2)+OUT3(3)+OUT3(4)                            8/19
C
      CALL BEEF
C
C DETERMINE THE SUPPLY OF CALORIES FROM LAND IN THE FLY-FREE NORTH PLUS
C                                LAND IN THE MIDDLE BELT
      SUPCA=OUT1(3)+OUT1(4)
C
C COMPUTE CALORIES PER ACRE, FLY-FREE AREA AND MIDDLE BELT - AFTER LOSSES  8/28
C
      CALAN=C9(3)*OUT1(3)/ATL(3)                                        8/28
      CALAM=C9(4)*OUT1(4)/ATL(4)                                        8/28
C
C SET CASH RETURNS FROM AMP FOR NEXT LANDN CALL
C
      DO 8021 K=1,4                                                     7/27
 8021 CRU(K)=RLRE(K)                                                    7/27
C
C NORTHERN NON-FARM INCOME, INPUT FOR POP. SECTOR
      YNFN=  YNFNI*EXP(.061*T)
C
C GENERATE CRITERION VARIABLES
C
C NIGERIAN DEMAND FOR COTTON  T=0. AT 1963
      DEMCOT=112000.*EXP((.025+.00123*T)*T)
C COTTON NET EXPORT BALANCE
      SUPCOT=.345*OUTP(2)
      FOREXC=(SUPCOT-DEMCOT)*PCW
C GROUNDNUT NET EXPORT BALANCE
      FOREXG=OUTP2(1)*PGW
C
C
C IF ISSQR=1  , THIS JOB  PARES SIMULATION VALUES WITH ACTUAL DATA SERIES
C IF ISSTIME=1  , ACTUAL DATA VALUES EXIST AT THIS POINT IN TIME AND
C   SUMSQR WILL BE CALLED DURING THIS PASS THRU THE MODEL
C
      ISSTIME=0
      IF(ISSQR.NE.1) GO TO 2011
      SIMVAL(1)=PFN
      SIMVAL(2)=OUTP(1)
      SIMVAL(3)=OUTP(2)
      IDATA=13*IDT
      DO 2010 KK=IDT,IDATA,IDT
 2010 IF (ITER.EQ.KK) ISSTIME=1
                        IF (ISSTIME.EQ.1) CALL SUMSQR(M,OLAM,XLAM,ISENS)
 2011 CONTINUE
C
C
      IF(IPRINT.NE.1) GO TO 2030
C
C BEEF PRINTING
      CALL BEEFP
C
```

```
C PRINT EXECUTIVE PROGRAM OUTPUT
      PRINT 8499
      PRINT 8500, FOREXC,FOREXG,YNFN,DEMCOT,SUPCOT
C
C  REGULATE SUMSQR PRINTING
      IF(ISSQR.NE.1) GO TO 2030
      CALL  SUMSQP  (M,OLAM,XLAM,ISENS)
C
 2030 CONTINUE
C
        CALL    CRTRN (TAXP,FFCI,CNFI,VAP,VAM,TPOPAG,TAXM,IDT,
     1  TPOPNA,SUPCFN)
C
      IF (ITER.EQ.NTIME) CALL SUMRYC (NAME,OLAM,XLAM,ISENS)
C
C
C STORE PLOT VALUES WHENEVER ISSTIME=1
C
       IF(ISSTIME.NE.1) GO TO 100
      NN=NN+1
      ZT(NN)=T
      Y1(NN)=PFN
      Z1(NN)=SERIESA(NN,1)
      Y2(NN)=OUTP(1)
      Z2(NN)=SERIESA(NN,2)
      Y3(NN)=OUTP(2)
      Z3(NN)=SERIESA(NN,3)
  100 CONTINUE
C
    1 CONTINUE
C
C  HERE ENDS THE TIME LOOP  --  DO 1 ITER=1,NTIME
C
      IF (ISSQR.NE.1) GO TO 21
       CALL VARPLOT (2, Y1, Z1, ZT, ZT, ZT, ZT, ZT, ZT,ZT, ZT,
     1                 1HP,1HD,1HH,1HH,1HH,1HH,1HH,1HH,1HH,1HH,
     2 14., 0., .012,.007, 13,  IGRAPH, ZT,3HPFN, 8HDATA PFN,
     3                         1HH,1HH,1HH,1HH,1HH,1HH,1HH,1HH,
     4       1.,1.,1.,1.,1.,1.,1.,1.,1.,)
       CALL VARPLOT (3, Y2, Z2,VALG,ZT, ZT, ZT, ZT, ZT,ZT, ZT,
     1              1HG,1HD,1HP,  1HH,1HH,1HH,1HH,1HH,1HH,1HH,
     2 14.,0.,1920000., 780000., 13, IGRAPH, ZT,7HOUTP(1),
     3 7HDATA GR, 4HVALG,          1HH,1HH,1HH,1HH,1HH,1HH,1HH,
     4 1.,1.,100000000.,1.,1.,1.,1.,1.,1.)
       CALL VARPLOT (3,Y3,Z3,VALC,  ZT, ZT, ZT, ZT, ZT,ZT, ZT,
     1                1HC,1HD,1HP,  1HH,1HH,1HH,1HH,1HH,1HH,1HH,
     2 14.,0.,343000.,112000.,13,    IGRAPH, ZT,7HOUTP(2),
     3 8HDATA COT, 4HVALC,          1HH,1HH,1HH,1HH,1HH,1HH,1HH,
     4 1.,1.,10000000., 1.,1.,1.,1.,1.,1.)
C
   21 CONTINUE
   22 CONTINUE
C
```

```
C  HERE ENDS THE RUN LOOP  --  DO 22 M=1,NR
C
C
 8012 FORMAT(1H1,*INITIAL VALUES FOR LANDN -- INDIRECTLY SET*,
     1  5X*RUN NUMBER  *,I2)
 8013 FORMAT(1H0,9X*AL(1)*,7X*AL(2)*,7X*AL(3)*,7X*AL(4)*,
     1  7X*SFL(1)*,6X*SFL(2)*,6X*SFL(3)*,6X*SFL(4)*,6X
     2  *TLJG*,8X*TLJC*/
     3 9X10(E11.4,1X))
 8014 FORMAT(1H0,9X*APL(1)*,6X*APL(2)*,6X*APL(3)*,6X*APL(4)*,
     1 6X*TLNF(1)*,5X*TLNF(2)*,5X*TLCF(1)*,5X*TLCF(2)*,5X*TLJF*/
     2 9X10(E11.4,1X))
C
 6007 FORMAT(1H0,*INITIAL VALUES FOR REEF SECTOR -- INDIRECTLY SET*)
 6008 FORMAT(1H0,9X*DFT*,9X*DMT*,9X*DFM*,9X*DMM*,9X*A2T*,
     1 9X*A2M*,9X*SFT*,9X*TDNM*/
     2 9X10(E11.4,1X))
C
 8499 FORMAT(1H0,* EXECUTIVE PROGRAM OUTPUT*)
 8500 FORMAT(1H0,9X*FOREXC*,6X*FOREXG*,6X*YNFN*,8X*DEMCOT*,6X*SUPCOT*/
     1 9X10(E11.4,1X))
 2020 FORMAT(30X3(E11.4,1X))
      END
```

```
CORE MAP .15.25.03.  NORMAL                    CONTROL
     ---TIME---LOAD MODE --L1--L2-----TYPE-------------USER--+---CALL----
     FWA LOADER  073717  FWA TABLES  066701
     -PROGRAM----ADDRESS-                        --LABELED---COMMON--
     CRTRN       000212                                   TIME      000100
                                                          SMCRT     000110
                                                          BFCRT     000131
                                                          PRNT      000136
                                                          MODEXT    000137
                                                          MODCRT    000165
     SUMRYC      000672                                   SMCRT     000110
                                                          MODCRT    000165
                                                          RUN       000540
                                                          TIME      000100
                                                          BFCRT     000131
                                                          TEMP      000543
                                                          MODEXR    000562
                                                          MODEXI    000573
     AMP         001564                                   MODEXT    000137
                                                          AMP1      001551
                                                          ALN3      001562
                                                          PRNT      000136
     LANDN       002445                                   TIME      000100
                                                          RUN       000540
                                                          LAND      002257
                                                          ALN1      002260
                                                          ALN2      002341
                                                          ALN3      001562
     BEEF        003741                                   TIME      000100
                                                          BFCRT     000131
                                                          BF1       003451
                                                          BF2       003512
                                                          BF3       003552
                                                          BF4       003665
                                                          LAND      002257
     SUMSQR      005027                                   SERS      004747
                                                          TEMP      000543
     VARPLCT     005240
     PLT         007360
     TABLIE      010043
     DELAY       010130
     TABEXE      010207
     DEMOG       010265
     SENSINC     010441
     POPDAT      012320
                                                          LUNIT     010573
                                                          RUN       000540
                                                          TIME      000100
                                                          PSIZE     010575
                                                          PINDEX    010602
                                                          POPD      010610
     POPSET      012454                                   LUNIT     010573
                                                          RUN       000540
                                                          PSIZE     010575
                                                          PINDEX    010602
                                                          POPD      010610
                                                          POPVAR    012413
                                                          DEMAND    012447
     POPOUT      012525                                   RUN       000540
                                                          TIME      000100
                                                          PSIZE     010575
                                                          PINDEX    010602
```

		POPVAR	012413
		DEMAND	012447
POPUP	012701	TIME	000100
		PSIZE	010575
		PINDEX	010602
		POPD	010610
		POPVAR	012413
POPSUM	013176	PSIZE	010575
		PINDEX	010602
		POPD	010610
POPPRT	013433	LUNIT	010573
		RUN	000540
		TIME	000100
		PSIZE	010575
		PINDEX	010602
		POPD	010610
		POPVAR	012413
		RGNNAM	013430
POPPLT	015675	LUNIT	010573
		TIME	000100
		PSIZE	010575
		PINDEX	010602
		RGNNAM	013430
INDEX	034053	PSIZE	010575
PARAMS	034074		
NISIM	034210	LUNIT	010573
		RUN	000540
		TIME	000100
		PSIZE	010575
		PINDEX	010602
		POPD	010610
		POPVAR	012413
		DEMAND	012447
		LAND	002257
		ALN1	002260
		ALN2	002341
		ALN3	001562
		MODEXT	000137
		AMP1	001551
		BF1	003451
		BF2	003512
		BF3	003552
		BF4	003665
		PRNT	000136
		BFCRT	000131
		SERS	004747
		TEMP	000543
		SENSIT	034162



Gloria Page 1/12/70