*The World's Largest Open Access Agricultural & Applied Economics Digital Library*

**This document is discoverable and free to researchers across the globe due to the work of AgEcon Search.**

**Help ensure our sustainability.**

Give to AgEcon Search

AgEcon Search
http://ageconsearch.umn.edu
aesearch@umn.edu

*Papers downloaded from **AgEcon Search** may be used for non-commercial purposes and personal study only. No other use, including posting to another Internet site, is permitted without permission from the copyright owner (not AgEcon Search), or as allowed under the provisions of Fair Use, U.S. Copyright Act, Title 17 U.S.C.*

ANNUAL SHELF

NOT FOR QUOTATION

GIANNINI FOUNDATION OF
AGRICULTURAL ECONOMICS
LIBRARY
WITHDRAWN
OCT 8 1975

# A User's Guide to Anchovy

# A Simulation Model of the Northern Anchovy Fishery

Appendix A

M. F. Tillman

June 23, 1972

University of Washington

File Manuscript No. 166-3

U.S NATIONAL MARINE FISHERIES SERVICE
ECONOMIC RESEARCH DIVISION

# A User's Guide to ANCHOVY
# A Simulation Model of the Northern Anchovy Fishery

## Appendix A

M. F. Tillman
6-23-72

## The Basic Model: SIMPLE ANCHOVY

### Purpose

ANCHOVY is a FORTRAN simulation model of the fishery for northern anchovy, Engraulis mordax Girard. This model provides the means of determining the biological and economic consequences of changing the regulations governing areas and seasons in the reduction fishery. Furthermore, through this model may be examined the sensitivity of optimal harvesting-processing systems to changes in biological parameters and models. Essentially, each sensitivity experiment will simulate the decision-making problem wherein a manager assumes that a given biological situation is true and plans to meet it but then encounters a completely different situation. This unexpected biological situation would then yield catches and economic returns either greater or smaller than anticipated.

SIMPLE ANCHOVY simulates that fishery for which the following simplifying assumptions have been made: Only a single stock of anchovy, the central stock, contributes biomass to the Southern California fishing area. The availability of anchovy in the fishing area varies monthly according to a strictly deterministic pattern. A single harvesting group, the Wetfish fleet, operates upon anchovy. Harvesters and processors do not change capacities during the course of simulation experiments but maintain optimal levels throughout.

General Description

SIMPLE ANCHOVY is formulated in terms of GAMES (see Appendix 1), the general-purpose simulator of resource use systems developed by Gales (1972). This Fortran IV program has been designed to simulate the activities of major sectors involved in the harvesting and marketing of renewable resources. The sectors modelled by GAMES include locations, stocks, harvesters, processors, regulators, products, and markets.

The anchovy fishery (Figure 2) is modelled by indicating through appropriate inputs the number of entities in each sector and their logical linkages. The user must also provide the values of parameters used to define system processes and structures and the initial values of variables used to describe systems behavior (see Appendix 2 for a list). Through appropriate control values the user may specify that certain built-in decision routines be used or else may provide algorithms of his own design by adding subroutines or by modifying existing ones.

As currently modified, the main SIMPLE ANCHOVY program resembles the partial listing given in Figure 3. The Labelled COMMON Blocks reserve sections of memory for storage

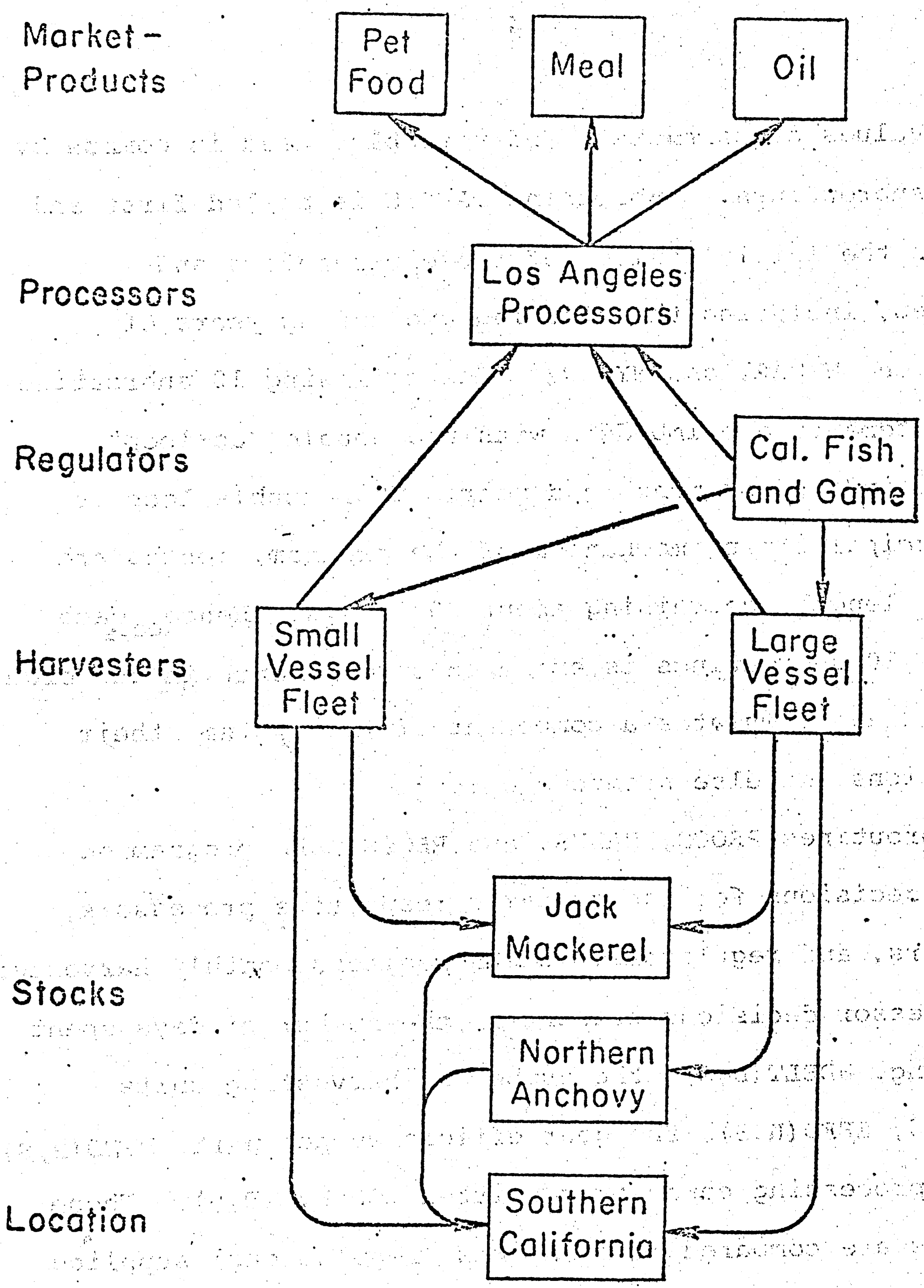


Figure 2 - Graphic representation of the logical relationships between sectors of the simplified northern anchovy system.

of the values of parameters and variables used in common by the 11 subroutines. Subroutine TAPEIN is called first and reads in the initial values of these parameters and variables, including the starting and ending years of simulation, NYEAR1 and NYEAR2. The following 10 subroutine call statements are imbedded within a double "do-loop" which is indexed by month and year. This double loop is the principal timing mechanism of the program; months are of equal length, containing about 30.5 days. Hence, each of these 10 subroutines is executed once a month in the order indicated and simulates a component of the system, their interactions, or else produces output.

Subroutines PROCS, HARVS, and REGLS make programmed monthly decisions for the system's respective processors, harvesters, and regulators. REGLS monitors monthly harvester and processor decisions concerning the number of days spent harvesting, HDELT(H,S), the number of harvesting units committed, EFFD(H,S), the gear efficiency per unit, FPWD(H,S), and the processing capacity committed, OPLEVL(P,S). These decisions are compared to standards (regulations) supplied by the user or else determined by the routine. If regulations are "broken", the subroutine then makes appropriate adjustments of the values of these variables.

Subroutine STOCKS simulates the monthly growth of the exploited anchovy resource and apportions mortalities to

```
      PROGRAM MAIN
      (Labelled COMMON Blocks)
      CALL TAPEIN
      DO 110 YEAR=NYEAR1, NYEAR2
      DO 100 MONTH = 1,12
      CALL PROCS
      CALL HARVS
      CALL REGLS
      CALL STOCKS
      CALL HRVST
      CALL RMARKT
      CALL PRCES
      CALL CMARKT
      CALL STATS
      CALL SMSTAT
  100 CONTINUE
  110 CONTINUE
      (Coding for Subroutines)
```

Figure 3. Partial listing of the main ANCHOVY program.

their appropriate sources. Described further in an ensuing section, it feeds catch values to HRVST, the subroutine which then simulates the monthly harvesting process. HRVST determines the catch of each stock by a harvester, QUANTH(H,S), his harvest proportional costs, HVCOST(H), and the cumulative catch taken from each stock, CCATCH(S,L).

RMARKT then simulates the sale of the harvesters' catches to the processors. Quantities not purchased are assumed to be dumped. Subroutine PRCES then transforms these newly purchased raw materials into finished goods, which are added to the processors' inventories, INVNT(P,D). Subroutine CMARKT then simulates the sale of these products on the open market to final consumers. The quantities demanded are determined from a user supplied demand curve and sales price.

STATS then computes and outputs financial statements for the processors and harvesters. It also provides cumulative physical reports describing through key variables the activities of the harvester, processor, and stock sectors. Subroutine SMSTAT provides an alternative means of obtaining cumulative physical reports. Although all reporst may be provided at monthly intervals, printout typically is suppressed until the year's end.

Subroutine STOCKS

STOCKS main job is to solve the catch equation and pass the result on to subroutine HRVST. The catch in numbers of each age group is first calculated for each harvester and stock:

$$CN(S,A,H) = \frac{F(S,A,H)}{Z(S,A)} * CENSUS(S,A,L) * EXP$$

where $EXP = 1 - e^{-Z(S,A)*(DT/NCYCL)}$

$F(S,A,H)$ = instantaneous rate of fishing mortality for a given stock, age, and harvester

$Z(S,A)$ = instantaneous rate of total mortality

$CENSUS(S,A,L)$ = total numbers alive at start of month

$DT = 1$ month

The accuracy of this solution depends on the parameter NCYCL; typically NCYCL is set at 4, yielding an effective DT of about 1 week. Catch in weight of each age group is then simply determined as follows:

$$CW(S,A,H) = CN(S,A,H)*WEIGHT(S,A)$$

where WEIGHT(S,A) = weight at age of an individual. CW(S,A,H) is the value ultimately passed to HRVST.

Within each month, then, the order of operations is as follows: Following the combined adjustments of PROCS, HARVS, and REGLS, the harvesting and capacity variables for the various fleets are passed to STOCKS:

EFFD(H,S) = fishing effort (number of boats)

FPWD(H,S) = fishing power

HCAPAC(H,S) = capacity (MT/boat/day)

HDELT(H,S) = fraction of a month fished.

These variables are used to calculate equivalent standard effort, F(H,S), in terms of boats fishing the entire month instead of a fraction of it, and the total harvesting capacity per harvesting interval, CAPAC(H,S).

Next, the age structure is updated. Fish in each age group have some probability of transferring to the next age group in each month, MARK(S,M). These transition fractions are applied to each age group. Recruits due to enter in the current month are added to the first age group, and fish leaving the last age group disappear. Within each age group, the size of individual fish is computed as a weighted average of the sizes of newly entered and residual fish. From these adjusted sizes and numbers at age, the biomass of the population is then computed, BMASS(S,L).

Contribution to spawning is then calculated for the current month. The number of females eligible to spawn is determined by the proportion of females in the population, FRFEM(S), and by a maturity at age schedule, FRMATF(S,A). Of these mature females the fraction which actually spawns during the month is given by FRSPAWN(S,M). Their egg production is computed by a fecundity at age schedule, EFEM(S,A). The results of this procedure are additions to the number of eggs deposited on the stock's spawning ground, EGG(G).

Instantaneous total mortality rates are then computed for each age group in the stock. Each age group may be subjected to a different total mortality, depending on natural mortality rate, NMORT(S,A), catchability coefficient, QCATCH(S,A), and availability factor, AVFR(S,L,M). In this model it has been assumed that catchability is age specific, that natural mortality may be constant or age specific, and that availability fluctuates monthly. Letting FF denote total units of standard effort operating in the stock's location during the month, instantaneous total mortality rates are calculated for each age group as

Z(A) = QCATCH(S,A)*AVFR(S,L,M)*FF+NMORT(S,A).

Given these mortality rates, the catch of anchovy by the harvester may then be computed subject to the constraint that the total catch may not exceed the total or assigned capacity of this harvester during the interval, CAPAC(H,S). As previously indicated, the harvester and natural mortality at first compete exponentially to determine the number of fish each would take if harvesting capacity were unlimited.

The harvester's catch in weight, CW(S,H), is then temporarily computed as the product of the number of fish taken and their weight at the beginning of the harvesting interval.

If this catch in weight exceeds the interval harvesting capacity, i.e.

CW(S,H)>CAPAC(H,S),

it and the harvester's fishing time are reduced accordingly. Firstly, an adjustment factor is calculated,

RC = CAPAC(H,S)/CW(S,H),

and applied to catch in numbers and to fishing time:

CN(S,H)' = RC*CN(S,H)

HDELT(H,S)' = RC*HDELT(H,S).

The harvester's catch in weight is then recalculated from this adjusted catch in numbers and his fleet is made inactive for the remainder of the fishing interval. Fish credited to the harvester in excess of capacity,

XTRA(S) = (1.0-RC)*CN(S,H),

are then subjected to natural mortality and returned to the population. Once the catch cycle has been completed, the number of fish remaining, CENSUS(S,A,L), is determined from the numbers caught and the numbers taken by natural mortality.

Growth in length which has occurred during the interval is then computed by a Von Bertalanffy equation. It has been assumed that growth is not density dependent. New individual weights at age are then computed by a cubic weight-length relationship.

Finally, future recruitment is calculated from the number of eggs deposited on the stock's spawning ground and the egg to recruitment survival rate. Recruitment is assumed to be density dependent and the model, therefore, adjusts egg survival in a compensatory manner according to user-supplied survival multipliers. The proper multiplier to be applied during the month is determined by the ratio of present biomass, BMASS(S,L), to the average or equilibrium biomass, WINF(S,L). This procedure determines the number of new recruits, RECRT(S,G,M), which will be available to the population after a delay of a number of months specified by RECINT(S,G,M).

Input

As indicated in Appendix 2, input cards are prepared in the English language free-field format described by Gales (ibid.) for GAMES. In general, an identifying keyword, usually the variable name, must occur in the first 8 columns of each card. This is followed by any combination of words and numbers which describe the variable and its subscripts; subscripts must occur in their proper order from left to right. Lastly, the value of the variable (if required) is indicated in the remaining card columns. For example, the data card indicating the maximum age of stock 1 would be prepared as follows:

```
col 1       9

    AGEMAX  MAXIMUM AGE FOR STOCK 1 = 7
```

All 80 columns of the data card may be used. Decimal points must be punched only to indicate decimal fractions; otherwise they are not required.

In addition to single values, users may also specify ranges or sequences of values for each of several parameters (subscripts or variable values) on a card. Ranges are specified by the keyword *TO* and sequences, by the keyword ***(.....). In effect when input routines encounter either of these 2 keywords, the same data card is read several times, and on each succeeding read a new parameter value is assigned from among those delimited by a given range or sequence.

The *TO* keyword assigns an initial value to a parameter and then increments it by one for each succeeding read of the card until it equals or exceeds the terminal value. The initial and terminal values immediately precede and follow, respectively, the keyword *TO*. For example, the data card specifying the fraction of the population which is female for stock 1 would be prepared as follows when that value remains constant over all ages:

FRFEM FRACTION FEMALE S 1, AGE 1 *TO* 7 = .56

In this manner, a single card replaces the seven normally required to define FRFEM(1,1) through FRFEM(1,7).

The ***(....) keyword encloses a sequence or list of values. Each read of the card assigns a new element of the list, from left to right, to the same parameter until the list is exhausted. For example, the data card specifying initial population size for ages 1-3 of stock 1 would be prepared as follows:

CENSUS NO. FISH S 1, AGE 1 *TO* 3 = ***( 0,228.85,76.18 )

Note that the left and right parentheses, respectively, must be followed and preceded by a single space. This single card is equivalent to the following 3 cards:

CENSUS NO. OF FISH S 1, AGE 1 = 0

CENSUS NO. OF FISH S 1, AGE 2 = 228.85

CENSUS NO. OF FISH S 1, AGE 3 = 76.18

Data cards prepared in this manner then are organized into the 10 data blocks indicated in Appendix 2. All of the variables listed must be included to run experiments with SIMPLE ANCHOVY. The 10 data blocks must occur in the order given and each, including the last one, must end with a a blank card. The variables within a given block may occur in any order desired by the user. Cards beginning with the letter C are comment cards which may be inserted to provide additional information about the input; they are ignored by the input routines.

Before examining these data blocks in detail, the following comments should be noted. The derivation of values for all listed variables is given in Tillman's (1972) analysis of the northern anchovy fishery. The following subscript abbreviations have been used:

S - Stock
A - Age
L - Location
M - Month
G - Spawning Ground
K - Table Value

Stock subscript 1 refers to the central stock of northern anchovy, and age subscripts 1-7 correspond to real anchovy ages 0-6.

Data block 0 controls the printout of input values. Due to the length of this listing and since only a few values will be changed from one experiment to the next, this printout has been suppressed.

Data block 1 controls the length of simulation and the form and frequency of financial and physical output reports (which are described in an ensuing section). The statements listed in Appendix 2 request that financial statistics be accumulated on an annual basis over a 20 year period. Printout of such annual reports is suppressed until the last year simulated, year 20. Physical reports for the stock, harvester, and processor sectors are to be printed out at the end of month 12 of year 20. If the user desires more frequent reports, say, every 5 years, the following alternative statements might be used:

```
CUMULATIVE FINAN REPORTS PRINTED YEARS ***( 5,10,15,20 )
PHYSICAL REPORTS ARE PRINTED YEARS ***( 5,10,15,20 )
```

Biannual stock reports for the above years could be obtained by using the following statement:

```
STOCK REPORTS PRINTED MONTHS ***( 6,12 )
```

Data block 2 is a verbal description of the system portrayed in Figure 2. The number of entities in each sector of the fishery and the logical linkages between entities in the different sectors are indicated. Note that in SIMPLE ANCHOVY the small vessel fleet, pet food, and jack mackerel have been removed from the description to simplify the model. The statement

```
PRODUCT 1 *TO* 2 SOLD ON MARKET 1 *TO* 2
```

may be interpreted as product 1 is sold on market 1, product 2 on market 2.

In data block 3 are defined, for financial report purposes, the sets to which harvesters and processors belong. Variables NHSET and NPSET are used to define the elements respectively comprising each harvester or processor set; e.g.

```
NHSET HARV SET 2 ENTRIES 1 *TO* 3, CONTAINS HARVESTERS
    ***( 2,3,5 )
```

The financial statements for all elements in a given set are summed together and printed out as a single report. These summed harvester and processor reports themselves may be summed together and printed out as a single financial report. The variable NHPSET is used to define these harvester-processor entities. The statements in Appendix 2 request that the financial reports of the harvester and processor in the simple anchovy system be amalgamated into a single annual report. The last statement of data block 3 requests that both the combined report and a separate report for the harvester and processor be printed out. To get only the combined report or only separate reports, the user should use, respectively, the following statements:

```
CBONLY PRINT OUT COMBINED REPORT ONLY
SPONLY PRINT OUT SEPARATE REPORTS ONLY
```

In data block 4, values are assigned to the following marketing parameters:

ALPHA, BETA, GAMMA - See Appendix 1 for a description of these. Since marketing is not examined and since perfectly elastic demand curves are used, the values assigned these parameters are not vital.

TRANS - These define the conversion factors indicating how efficiently raw materials (anchovy) are transformed into finished goods (fishmeal and oil) by processors.

PDC,QDC - These define the price and quantity axes of the perfectly elastic demand curves for fishmeal and oil (quantities in terms of MT/month).

Variables in data block 5 describe the initial state of processors in the fishery:

MSHARE - Since in the simple model processors are amalgamated together into one processing entity, the user need not worry about market shares since that entity controls all of the market.

PCOSTU - The cost of a new plant, excluding the cost of land.

PTOTPE - The total value of all plants initially available (number X PCOSTU).

PUCAP - The processing capacity per dollar of equipment (capacity per month in MT/PTOTPE) is used to calculate the new operating level when capacity of a certain value is added to the present total plant value.

PFXDXP - The total fixed cost of operating all plants in terms of cost per month. Includes an opportunity return on total investment.

POPERX - The variable costs of operation depend on the kind and amount of raw material processed. Value in terms of cost per MT of anchovy.

PLOCX - The x and y coordinates of the processor's location.

Variables in data block 6 describe the initial state of harvesters in the fishery:

HCOSTU - The cost of a new vessel.

HTOTPE - The total value of the fleet (number x HCOSTU).

HFXDXP - The fixed cost of operating a single vessel. Unlike processors' fixed costs, the total fleet cost is calculated internally by the program.

HVRCPE - The day-proportional cost of operating a vessel (cost per day of fishing).

HRCOST - The harvest-proportional cost of operating a vessel (cost per MT of anchovy captured).

HPLOCX - The x and y coordinates of the fleets initial location, relative to the processor's location.

In data block 7 values are assigned to variables used by the stock sector of the model. The variable names listed in Appendix 2 correspond to those used in the previous section describing subroutine STOCKS. For further description of these variables, the user should refer to pages 10-11 of Appendix 1, the writeup for GAMES. The values assigned in Appendix 2 are taken from Tillman (ibid.).

In most cases Appendix 2 is self-explanatory, and a detailed description of variables in data block 7 thus will not be made. It should be noted, however, that variables FFISH, FEGGS, FRECR, and FCOEFF act as multipliers and are used to scale values of the variables immediately following. Hence, they must precede the variables that they multiply: respectively CENSUS, EFEM, RECRT, and QCATCH. Further explanation also is required for the following variables:

WINF - This is the equilibrium biomass of the virgin stock, not the maximum biomass.

SMULT - The values listed define a Beverton and Holt stock-recruit relationship.

LMULT - This is set equal to one since growth in length is not density dependent.

MARK - These values define the fraction of the population which ages (has a birthdate) in the given month.

WF - This vector converts catch by age into total catch.

Data block 8 reads in processor decisions. These decisions, as well as those for harvesters and processors, remain in force throughout the length of a simulation experiment:

OPLEVI - The operating level or total plant capacity to be devoted to processing anchovy, in MT/month.

PURPRI - The purchase price offered to harvesters for their catches (price per MT of anchovy).

PRICES - The selling price desired for finished products (price per MT of meal or oil).

Data block 9 reads in the following decisions for harvesters:

EFFDI - The number of vessels operating on anchovy.

HCAPAC - The capacity in MT of an average vessel.

FPWDI - The harvesting efficiency or relative fishing power of the average vessel.

DYSTRP - The average trip length in days.

HLOCX - The x and y coordinates of the fishing location which indicate the distance in miles from the initial harvester location.

Data block 10 reads in the following decisions for regulators:

XI - A control variable indicating which of the built-in seasons is to be used:

1 - 8 month season, summer closure (15 Sept. - 15 May).

2 - 8 month season, winter closure (1 May - 31 Dec).

3 - 12 month season.

NAUT - A control variable indicating whether or not quotas are enforced:

1) Not enforced

   NAUT 1 = 1

   NAUT 2 = 1

2) Annual quotas enforced

   NAUT 1 = 0

   NAUT 2 = 1

   Set the quota as follows:

   CQUOTA S 1 L 1 = # MT/year

3) Monthly quotas enforced

   NAUT 1 = 1

   NAUT 2 = 0

   Set the quota as follows:

   CMQ S 1 L 1 = # MT/month

OPLL - The maximum allowable operating level or processing capacity. If not a constraint, set an arbitrarily high value. In terms of MT/month.

EFFL - The maximum allowable number of vessels.

RDELT - The maximum allowable number of days of fishing per month, up to 30.6 days.

FPWL - The maximum allowable fishing power or harvesting efficiency.

## Using SIMPLE ANCHOVY

To use SIMPLE ANCHOVY most expeditiously, the program deck, a compiled version of the program, and the previously defined data deck are all placed on a tape (#625 at the Center for Quantitative Science in Forestry, Fisheries and Wildlife, University of Washington). These three entities or decks comprise three sequential files on that tape (6-8 on tape 625). By utilizing a tape, the user avoids the necessity of carrying around several boxes of cards and, as will be seen, facilitates the manipulation of values in the data deck. To execute a simple experiment with the data as listed in Appendix 2, the following setup (using files 7 and 8 of tape 625) is required:

```
Job card (Set core requirement 55000 and time 60)
REQUEST,TAP,VRN=625,FILES=7-8,IN,HI,DI.
MAP(OFF)
COPYBF(TAP,PROG)
COPYBF(TAP,DATA)
REWIND(PROG,DATA)
SETCORE.
PROG(DATA)
6-7-8-9- card
```

All cards begin in the first column.

More complex experiments involving the manipulation of values in the data deck make use of the UPDATE system provided by the CDC 6400. UPDATE is a general system for manipulating or editing files of information placed on a tape. By means of a special control card and a special data deck, the user may indicate which file is to be edited, which statements or cards are to be changed and what specific changes are to be made. To change values in the SIMPLE ANCHOVY data deck (file 8 of tape 625) and then run an experiment, the following setup is required:

```
Job card (CM55000,T60)
REQUEST,TAP,VRN=625,FILES=7-8,IN,HI,DI.
MAP(OFF)
COPYBF(TAP,PROG)
COPYBF(TAP,PROG)
REWIND(PROG,DATA)
UPDATE(P=DATA,W,D,8,Q)
SETCORE
PROG(COMPILE)
7-8-9 card
     Text Manipulation Cards
6-7-8-9 Card
```

The UPDATE card requests that the data file be changed temporarily as indicated by the deck of text manipulation cards and that this new version be placed on a file called COMPILE. The experiment then is executed using COMPILE as the data source. Since the changes made are not permanent, the data file effectively is restored to its original version after the experiment is run.

Before setting up the text manipulation cards, the user should first observe that every data card in Appendix 2 is followed by a name, DATA1, and a number. When the UPDATE system initially is used to place this data file on tape, these identifying tags are appended to the right hand side of every card. Thus, in any manipulation, a specific card in this data file may be referenced by its identifying tag, using the format DATA1.m where m is the specific card number.

The following statements comprise a Text Manipulation Deck used to change the enforcement of annual quotes and the frequency of reports:

```
*IDENT CHD1
*COMPILE DATA1
*DELETE DATA1.138
NAUT 1 = 0
NAUT 2 = 1
CQUOTA S 1 L 1 = 1
*DELETE DATA 1.13, DATA 1.14
CUMULATIVE FINANCIAL REPORTS ARE PRINTED YEARS ***( 5,10,15 )
PHYSICAL REPORTS ARE PRINTED YEARS ***( 10,20 )
```

Note that several changes may be made at one time, that they may occur out of order in the manipulation deck, and that all statements begin in column 1. The *IDENT statement provides an identifying tag for the new data cards included in the manipulation deck. Tags other than CHD1 may be used but cannot exceed 7 characters in length, and DATA1 cannot be used again. The *COMPILE statement indicates that the new version of DATA1 is to be written onto the COMPILE file.

The *DELETE statements in effect cause the named data card or range of cards to be erased. If any data cards follow the *DELETE statements in the manipulation deck, then these new cards are inserted in place of the erased card(s). Thus, in the above example, data card 138 is erased and 3 new cards are inserted in its stead; then data cards 13 and 14 are erased and 2 new cards inserted.

To simply insert new data cards without any erasures, the *INSERT statement is used:

```
*INSERT DATA1.136
SLIMIT S 1 L 1 = 10 cm.
```

The above example would insert a size limit regulation into data block 10 immediately after data card 136. (The programming mechanisms to handle such a regulation are available but not used in the simple model.)

In the simulation experiments Tillman (ibid.) conducted with SIMPLE ANCHOVY, the following manipulations were required:

1) Changing stock-recruit curves for the M=1.10 model:

*DELETE DATA1.96

SMULT - use either of the 2 alternative tables provided by Tillman (ibid.).

2) Changing mortality models from M=1.10:

*DELETE DATA1.79, DATA1,80

CENSUS - initial population size by age, 2 cards

*DELETE DATA1.95

SURVN - egg-recruit survival

*DELETE DATA1.96

SMULT - egg-recruit survival multipliers

*DELETE DATA1.99

RECRT - initial numbers of recruits

*DELETE DATA1.107

NMORT - natural mortality

*DELETE DATA1.106

QCATCH - coefficient of catchability

3) Changing seasonal regulations:

*DELETE DATA1.137

XI - season code

4) Changing areal regulations:

```
*DELETE DATA1.106
QCATCH - use coefficient for the inshore area
```

5) Changing efficiency of operations in the inshore area:

```
*DELETE DATA1.47
TRANS( 1,1,1 ) - conversion factor for fishmeal.
```

Output

The results obtained from each experiment will depend upon which output reports are requested in data blocks 1 and 3. In general, however, a financial report will provide the net economic returns (before taxes) obtained by harvesters, by processors, and/or by the combined harvester-processor entity. Appendix 3 provides an example of a combined harvester-processor financial report; this report indicates the economic performance of the total system during year 10. The physical reports for stock, processor and harvester sectors will include the variables indicated in Appendix 3. Additionally, a report of the average annual biomasses obtained during each year of simulation will precede the financial report~~(s)~~; these biomass values will provide an index of how population size varies during the course of experiments.

## Alternative ANCHOVY Models

### Stochastic Availability Model

As an alternative to the deterministic model used by SIMPLE ANCHOVY, Tillman (ibid.) offers an hypothesis that monthly availability of anchovies follows a stochastic pattern. Assuming that availability is distributed normally about the average value for each month, this stochastic pattern is portrayed in Figure 8. In this figure the hatched line represents the pattern of mean monthly values and the 2 solid lines, the patterns of 2 standard deviations above and below each monthly mean.

To implement this model, special coding must be inserted into the SIMPLE ANCHOVY program. Since this program is on tape, these insertions can be done on a temporary basis with the UPDATE system just as they were for the input data set. Furthermore, both a program update and a data update can be made on the same computer run. The program update must precede the data update. The setup to carry out both of these updates just before the execution of an experiment is as follows (using files 6-8 on tape 625):

```
Job card (CM55000,T60)
REQUEST,TAP,VRN=625,FILES=6-8,IN,HI,DI.
MAP(OFF)
COPYBF(TAP,OLDP)
COPYBF(TAP,OLDB)
```

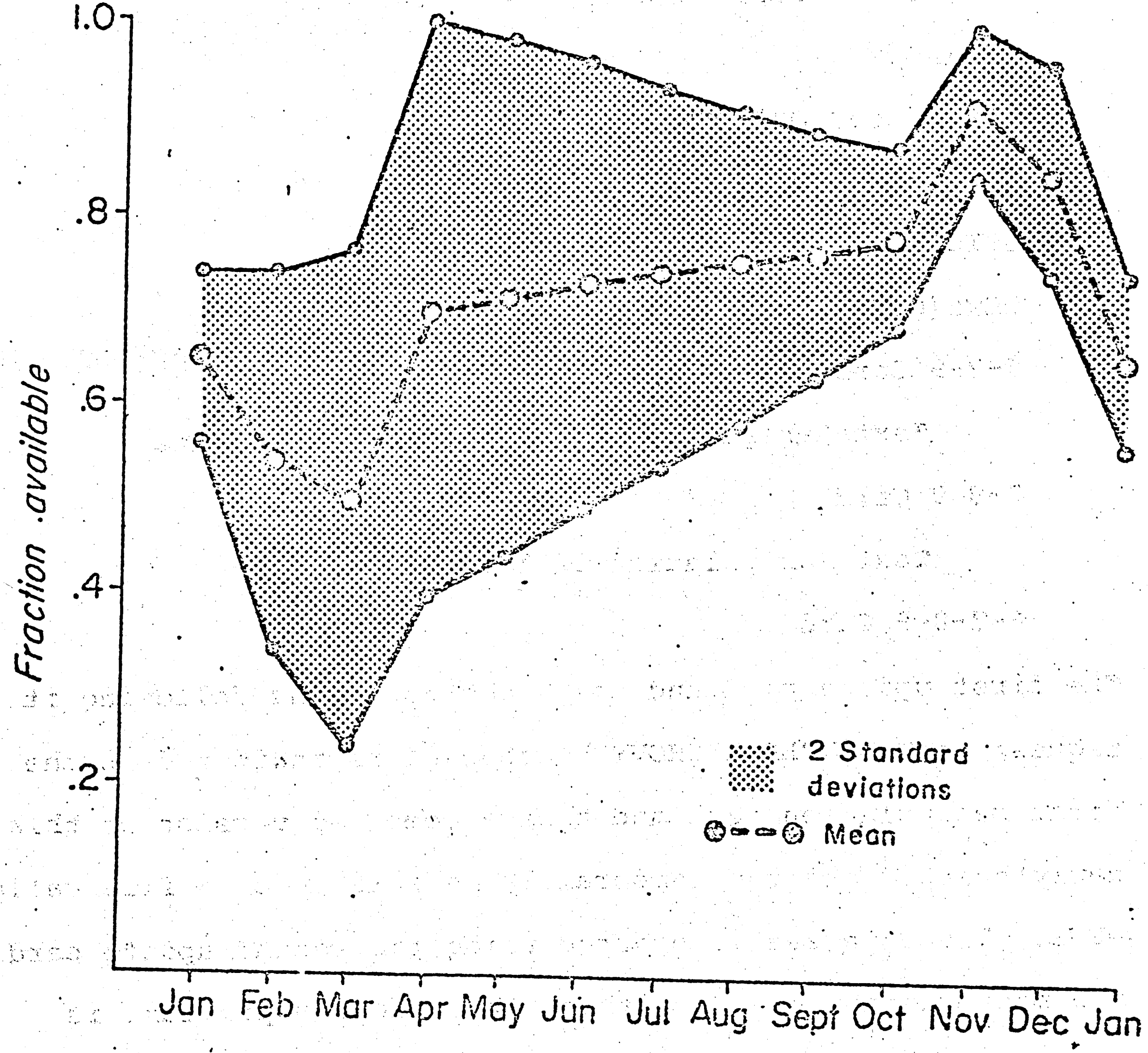


Figure 8 - Stochastic pattern of seasonal availability for northern anchovy.

```
COPYBF(TAP,DATA)
REWIND(OLDP,OLDB,DATA)
UPDATE(P=OLDP,W,Q,L=0)
FORTRAN(I=COMPILE,MODE=G,B=OBJ,L=DUMMY)
REWIND(OBJ)
COPYL(OLDB,OBJ,NEWB)
UPDATE(P=DATA,W,D,8,Q)
SETCORE.
NEWB(COMPILE)
7-8-9 Card
     Text Manipulation Deck for Program Update
7-8-9 Card
     Text Manipulation Deck for Data Update
6-7-8-9 Card
```

The first update card and the 3 control cards following it request that SIMPLE ANCHOVY be changed as indicated in the first manipulation deck and that a compiled version of this newly changed program temporarily be written on a file called NEWB. The data set is updated using the second update card and manipulation deck and then NEWB, the new program, is executed using COMPILE, the new data set. The original versions of the program and data set will still be on the tape after the experiment has been conducted.

The Text Manipulation Deck for the program update must contain the following statements:

```
col. 1
        *IDENT STOCH (or some other name)
        *COMPILE MAIN
        *INSERT NEWG.28
          col. 7
                DUMMY = RANF(.247)
        *COMPILE STOCKS
        *INSERT STOCKS.282
          col. 7
                R1 = RANF(0)  $  R2 = RANF(0)
                ZV = SQRT(-2.0*ALOG(R1))*COS(2.0*3.14159*R2)
                AVFR(S,L,M) = AMAVFR(M)+SDAVFR(M)*ZV
                IF(AVFR(S,L,M).LT.0.0)  AVFR(S,L,M) = 0.0
                IF(AVFR(S,L,M).GT.1.0)  AVFR(S,L,M) = 1.0
```

The cards to be inserted are FORTRAN statements and thus must begin in column 7. The first insertion, involving the MAIN program, is used to change the sequence of random numbers generated in subsequent runs. If the value .247 is not changed, the same sequence of random numbers will be obtained in every experiment. However, if a different 3 digit decimal fraction is used each time, a new sequence will be obtained in each run. The second insertion, involving subroutine

STOCKS, utilizes these sequences of random numbers, the monthly mean availabilities (AMAVFR), and their standard deviations (SDAVFR) to calculate the value of fractional availability applicable during a given month. The method used is described further by Tillman (ibid.). Data statements indicating the mean availability and standard deviation for each month are already built into the simple model and need not be added again. Note that monthly availability values have been constrained to fall within the range (0,1).

Market-Industry Growth Model

According to Tillman (ibid.), the addition of 1-2 new plants to the present system would not be an unlikely event given that favorable market circumstances prevailed. On the other hand, a decrease of capacity might occur if a disaster such as the marketing of a new low cost protein substitute occurred. If subroutine PROCS is modified appropriately, it will simulate these patterns of change in processing capacity. The user sets the number of UNITS to be added (or removed) from the initial number of plants and the year, N, in which the gain (or loss) occurs (takes effect the beginning of that year). Because of the indivisible nature of these large capital inputs, capacity grows incrementally at rather long periods. Hence, growth (or a loss) occurs in the middle of a 10 year simulation period, typically at N = 6.

The harvesting fleet should also grow (or decline) in response to market conditions. Given appropriate modifications, subroutine HARVS will simulate patterns of change in fleet size. It has been assumed that the fleet grows more quickly than does processor capacity, that growth is a constant function of time, i.e. a constant number of vessels is added each year, and that yearly additions to the fleet enter at the start of a season. The user indicates HINCR, the number of units added in each year (zero for the static model) and, if processing capacity steps down, he also enters NSTOP, the year in which growth stops, and HDECR, the number of vessels lost each year following a market-disaster.

As might be surmised, the above required changes in subroutines PROCS and HARVS are induced temporarily by means of the UPDATE system. The exact control card setup used for the Stochastic Availability Model is also used for this model. Of course, a new Text Manipulation Deck is required for this program update. The following statements must be included in this new manipulation deck:

```
*IDENT GRW

*COMPILE PROCS

*DELETE CHNT3.176
```

```
col. 7
          DATA UNITS/-2.0/
          DATA N/6/
  *COMPILE HARVS
          DATA HINCR(1)/4.0/
          DATA NSTOP/7/
          DATA HDECR(1)/-5.0/
```

The specific values on the 5 DATA cards will, of course, change depending on which market-industry growth situation the user wishes to simulate. The above values describe the situation wherein 2 plants are lost in year 6 and the fleet size follows a triangular growth pattern, adding 4 vessels per year until year 7 and losing 5 per year thereafter. The following generalized table indicates the values to be entered on these DATA cards for different growth situations:

| | UNITS | N | HINCR | NSTOP | HDECR |
|---|---|---|---|---|---|
| 1) Add plants in year N, constant fleet growth | +R | I<10 | +R | I>10 | 0 |
| 2) Add plants in year N, stop fleet growth at NSTOP | +R | I<10 | +R | I<10 | 0 |
| 3) Lose plants at N, stop fleet growth at NSTOP | -R | I<10 | +R | I<10 | 0 |
| 4) Lose plants at N, triangular fleet growth | -R | I<10 | +R | I<10 | -R |
| 5) No change in plants, constant fleet growth | 0 | I>10 | +R | I>10 | 0 |

Where R = whole real number (punch decimal)

I = positive integer number

0 = zero

Concerning the data set update which may accompany the program update, the following caveats are made: The length of simulation runs should be shortened to 10 years to agree with the time scale used by PROCS and HARVS. If the user intends to start at some harvester-processor configuration other than the one already on the data file, he must change the following initial values for processors to match his initial number of plants: PTOTPE, PFXDXP, OPLEVI. And for harvesters, EFFDI must be changed to indicate the new number of vessels with which he intends to begin.

## Mixing Stocks Model

Tillman (ibid.) derives an alternative stock structure model in which a mixture of two stocks occurs in the Southern California fishing area. He hypothesizes that both the southern (stock 1) and central stocks (stock 2) contribute some fraction of their respective biomasses to the fishing area. The logical representation of this Mixing Stocks Model is given in Figure 11.

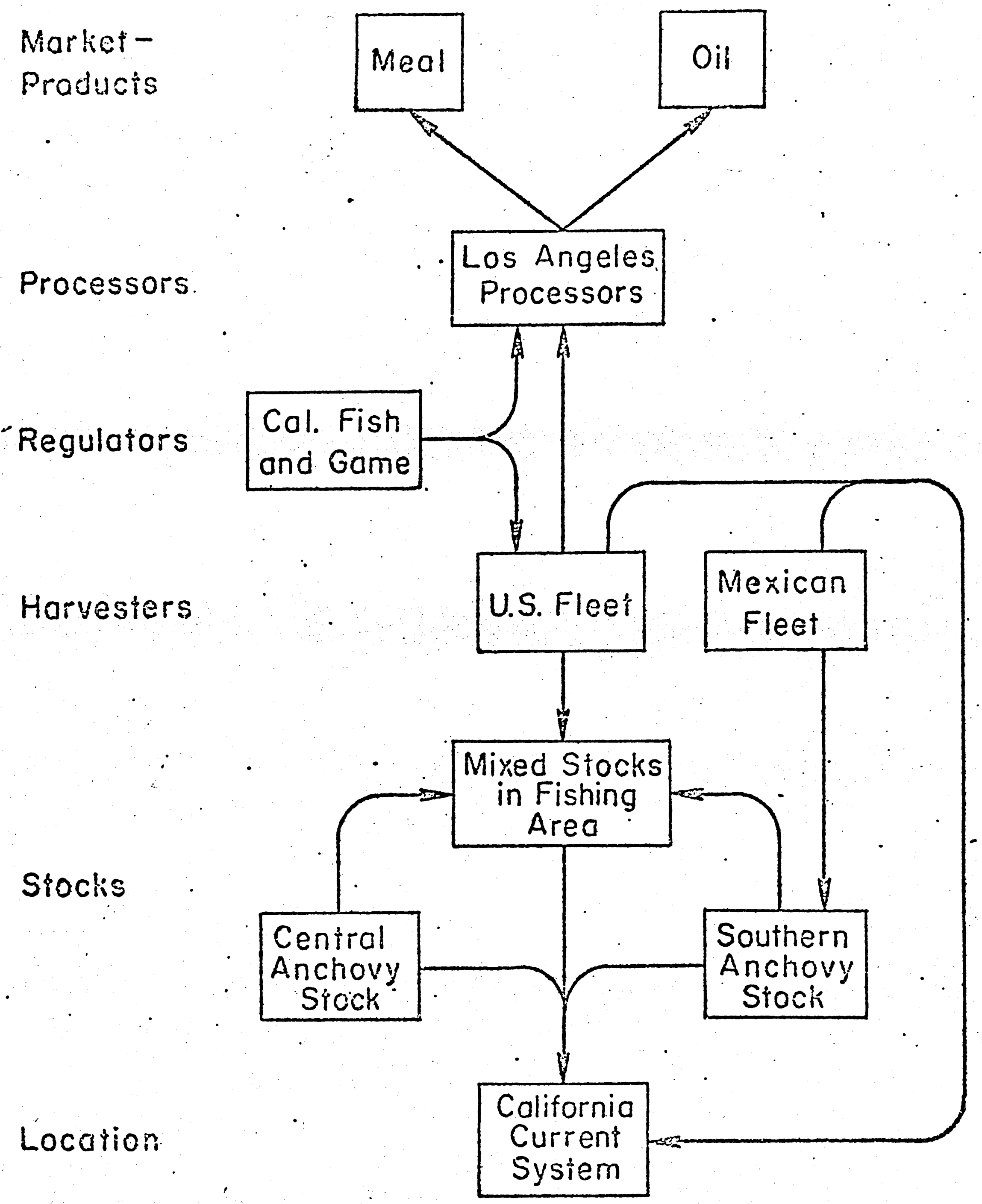


Figure 11 - Graphic representation of the logical relationships between elements of the mixing stocks model.

Biomass in the fishing area is assumed to be 5.0 million tons. Given a contribution by each stock to this same area, biomass is determined by

$$FBMASS = \sum_{i=1}^{2} A_i * BMASS_i \qquad (7)$$

where $BMASS_1$ = 1.0 million tons, $BMASS_2$ = 5.0 million tons, and $A_i$ defines the fraction of each stock which is available to the fishery. Since these fractions are unknown, a variety of possibilities might be derived in order to examine the consequences of mixing. Given $A_1$ fixed, $A_2$ is found by rearranging (7) and substituting the known biomasses. This porcess yields the following pairs of availability factors considered by Tillman (ibid.):

| Southern Stock Contribution | Northern Stock Contribution |
|---|---|
| .25 | .95 |
| .50 | .90 |
| .75 | .85 |

Simulating this system requires that an identical dummy harvester be set up for the central and southern stocks: each having the same number and type of vessels, catch coefficients, and seasonal effort patterns. The fishing power of each, FPWDI, then is adjusted by its above respective availability factor. The catch of each dummy harvester thus represents the contribution of the available part of each stock to the fishery and the sum of these catches, the total catch taken in the fishery area by the U.S. fleet from the given mixture of stocks.

Such extensive changes of SIMPLE ANCHOVY are required to accomodate this model that temporary updates are no longer economical. Comparison of the program listings given in Appendix 4 and 5 will show that changes mainly involve the dimensions of the COMMON Blocks used for storing variables. Appendix 5 is a listing of the program obtained following a permanent update of SIMPLE ANCHOVY. This program, its compiled version, and an appropriate data deck are all placed on a tape to facilitate use of the Mixing Stocks Model (files 9-11 on tape 625). To use the model, the following setup is required (using files 10 and 11 on tape 625):

```
Job card (CM60000,T60)
REQUEST,TAP,VRN=625,FILES=10-11,IN,HI,DI.
MAP(OFF)
COPYBF(TAP,PROG)
COPYBF(TAP,DATA)
REWIND(PROG,DATA)
UPDATE(P=DATA,W,D,8,Q)
SETCORE.
PROG(COMPILE)
7-8-9 Card
     Text Manipulation Deck for Data Update
6-7-8-9 Card
```

Note the change in core requirements, from 55000 to 60000.

For every change in the mixture of stocks, the following statements must be included in the Text Manipulation Deck in order to change the data set appropriately (see Appendix 6 for a listing of current values):

```
*DELETE DATA2.62
PUCAP PROC 1 STOCK 3 = desired value
PUCAP PROC 1 STOCK 1 = desired value
PUCAP PROC 1 STOCK 2 = desired value
*DELETE DATA2.122, DATA2.123
QCATCH STOCK 1 AGE 1 *TO* 7 = ***(desired value(s))
QCATCH STOCK 2 AGE 1 *TO* 7 = ***(desired value(s))
*DELETE DATA2.149
OPLEVI PROC 1 STOCK 3 = desired value
OPLEVI PROC 1 STOCK 1 = desired value
OPLEVI PROC 1 STOCK 2 = desired value
*DELETE DATA2.157
FPWDI HARV 1 STOCK 1 = desired value
FPWDI HARV 2 STOCK 2 = desired value
```

The derivation of appropriate values is as follows:

1) Recall that the $A_i$'s, the availability factors applied to each stock's biomass, are also used to adjust fishing powers of the dummy fleets:

$$\text{FPWDI}(1,1) = A_1*1.726$$

$$\text{FPWDI}(2,2) = A_2*1.726$$

where 1.726 is the fishing power of vessels having 191 MT capacities. Given $(A_1, A_2) = (.25, .95)$,

FPWDI(1,1) = 0.432

FPWDI(2,2) = 1.640

would be the values entered above.

2) OPLEVI(1,3) is the processing capacity which is allocated to anchovies in general. However, since 2 different stocks of anchovies actually are being taken in the fishing area, the program requires the processors allocate this total capacity over both stocks. Assuming an allocation which is proportional to the fraction each stock in the fishing area is of the total biomass in that area, i.e.,

$P_i = A_i\ \mathrm{BMASS}_i/\mathrm{FBMASS}$

then OPLEVI(1,1) = $P_1$*OPLEVI(1,3)

OPLEVI(1,2) = $P_2$*OPLEVI(1,3).

Given $(A_1, A_2) = (.25, .95)$, $(P_1, P_2) = (.05, .95)$.

And if OPLEVI(1,3) = 30576 (see Appendix 6),

OPLEVI(1,1) = 1528.8

OPLEVI(1,2) = 29047.2

would be the values entered

3) Recall that PUCAP is defined as

OPLEVI/PTOTPE.

Thus, given the above values of OPLEVI and PTOTPE = 1.2 million, the following values of PUCAP would be entered:

PUCAP(1,3) = 0.02548

PUCAP(1,1) = 0.001274

PUCAP(1,2) = 0.024206

4) The appropriate QCATCH values for a given set of $A_i$'s are indicated in the following table for different areal restrictions (see Tillman (ibid.) for derivation):

| Mixture of stocks | Age | Coefficient when Inshore closed ($10^{-3}$) | Coefficient when Inshore open ($10^{-3}$) |
|---|---|---|---|
| (.25, .95) | 0 | 0.24 | 0.38 |
| | 1 | 2.81 | 4.14 |
| | 2-6 | 9.04 | 9.04 |
| (.50, .90) | 0 | 0.25 | 0.39 |
| | 1 | 2.92 | 4.22 |
| | 2-6 | 9.04 | 9.04 |
| (.75, .85) | 0 | 0.26 | 0.40 |
| | 1 | 2.97 | 4.31 |
| | 2-6 | 9.04 | 9.04 |

Mexican Entry Model

If the user examines Figure 11 closely, he will find that Mexican entry into the anchovy system can be simulated by the Mixing Stocks Model. By changing the data set appropriately, one can activate a second harvester which operates exclusively upon the southern stock (stock 1). Thus, it has been assumed that Mexican vessels fish only on the remnant of the southern stock which does not migrate north into the Southern California fishing area.

To implement this model, one uses the control card setup previously discussed for the Mixing Stocks Model.

In addition to the statements listed in the Text Manipulation Deck for that model, the user also must include the following statements:

```
      *DELETE DATA2.160
      EFFDI HARV 4 STOCK 1 = no. of vessels
      *DELETE DATA2.162
      FPWDI HARV 4 STOCK 1 = adjusted value
```

The adjusted value of FPWDI is determined as follows from the value of $A_1$ applied to the southern stock:

$$FPWDI(4,1) = (1-A_1)*1.726$$

Given $A_1 = .25$, the appropriate value would be

$$FPWDI(4,1) = 1.294$$

## Literature Cited

Gales, Lawrence E. 1972. NEW GAMES: a multi-purpose interactive resource management program. Quant. Sci. Paper 32. 40p.

Tillman, Michael F. 1972. The economic consequences of alternative systems: a simulation study of the fishery for northern anchovy, *Engraulis mordax* Girard. Unpub. Ph.D. Thesis, University of Washington. 204p.

# Appendix 1 - User's manual for GAMES

Quantitative Science Paper No. 32*

# NEW GAMES: A MULTI-PURPOSE INTERACTIVE RESOURCE MANAGEMENT PROGRAM

Programmed by

Lawrence E. Gales

CENTER FOR QUANTITATIVE SCIENCE
IN FORESTRY, FISHERIES AND WILDLIFE

University of Washington
Seattle, Washington 98195

February 1972

*A series prepared under the general sponsorship of the Quantitative Ecology and Natural Resource Management Program supported by Ford Foundation Grant Number 68-183.

## I. Introduction

GAMES* was a first attempt at writing a really large multi-purpose simulation program. It was conceived as a set of processes which could be linked together through input cards in order to model a specific system. However, GAMES proved far from easy to use in practice, mainly due to the large amounts of input and output and to the need to re-program in order to meet differing storage requirements.

New GAMES is an attempt to ease these problems. It incorporates programming philosophies and software packages of quite general applicability, and can be thought of as a system in which one may embed a simulation program. It is designed to handle the most characteristic (yet overlooked) features associated with simulation: those that stem from the fact that the real world is inherently complex, possesses many parts, and continually changes. These features are of necessity reflected in the computer program which tends to be large, complex, and contains a very large number of subscripted variables. The very size greatly aggravates the problems of input, output, and code modification, and dictates a new programming approach.

These problems are usually overlooked by the programmer who concentrates attention on the simulation algorithms and leaves I/O and code modification as an afterthought. Normally, it is the unfortunate user who realizes the inadequacy of this approach after searching through 28 pages of output to find six relevant numbers. Just as computers were forced to evolve from "bare brains" with peripheral (i.e., "unimportant") I/O devices, to large systems where the central processor is a small part of the whole, so the difficulties in usage require that large simulation programs be embedded in a flexible I/O system designed with the user in mind, and attended by a program editing system.

Specifically, large simulation programs should include the following features:

1. Modular Structure.

It is essential to split a large program into a number of independently compiled blocks. This technique not only reduces costs because parts of the program can be altered without recompiling and relisting the whole, but it also enables the programmer to isolate logical units so they can be understood and changed without knowledge of the entire program. A useful device is to write the main program using only "if statements", "do-loops", and subroutine "call" statements. Then the overall structure of the program is evident at a glance.

* See Quantitative Science Paper #17.

2. Global Variables with Variable Dimensions.

Global variables are needed to pass information from one independent subroutine to another. Common blocks, as occur in Fortran, are inadequate because the programmer must redeclare and redimension dozens or hundreds of variables in dozens of subroutines. Variable dimensions are also needed to allow for storage needs that depend on the particular system being modelled.

Global variables are simulated in new GAMES by means of a set of routines that replace the symbolic dimensions of variables in labelled common blocks with user-defined numeric dimensions and which insert the resulting common blocks in programs wherever they are referenced by name.

3. Array Protection.

The most mysterious "bugs" in all of programming arise from the use of subscripts that lie outside the range permitted by the associated dimension statements. These "bugs" may wipe out data, parts of the program, I/O buffers, or even parts of the operating system and are often exceedingly hard to find because the effects may become evident long after the cause. This problem is greatly magnified in simulation programs because a) the program tends to be quite large, b) the large number of subscripted variables implies a large number of potential causes, and c) the ranges of subscripts must be set by data cards in order to maintain program generality, hence input values can cause the program to exceed dimension limitations even when the program is thoroughly debugged.

There is no satisfactory software solution* to the problem because of the very large amount of input that would have to be examined and the large number of instances in the program where subscripted variables occur.

4. Editing Systems.

Because of the great size of the program and the need for frequent changes, the programmer needs automatic means of locating and redefining variables, inserting blocks of code, changing dimensions, etc.

A few of these functions may be accomplished by the routines described in section two above.

5. A User Oriented Input Package.

The input must be visible, self-documentary, position tolerant, self-scheduling, must override previous input, and must possess default values. It is visible if the user has the option of printing out the input for each run; it is self-documentary if the data are embedded in text that define it to the user and keywords that define it to the program; it is position tolerant if a card can be inserted anywhere within a block; it is self-scheduling if a block is read at a time determined by parameters given on the first card of a block; it overrides previous input

* However, the new SCOPE3.3 Extended Fortran now offers this capability.

if the values of variables associated with data cards of a certain format are determined by the most recently read cards of that format; it possesses default values if an entire input setup can be stored on tape and then selectively overriden by input cards at execution time. The above input package is fully implemented and is described in detail in later sections.

6. A Selective Output Package.

A large simulation program may generate mountains of output, most of which is useless at any given time. The package must allow the user to select the output which meets his current needs and suppress the rest (this is especially important if he is running from a remote terminal). Therefore the output package should provide output in the form of relatively small labelled blocks which can be manipulated by the user through a command language in the input data. The commands, which should be flexible and self-explanatory, should allow the user to specify the form, frequency, and destination of output, and should allow him to create new output blocks or modify existing ones. For example, if CØST(1,1), CØST(1,2) and RENT(3) are existing output variables, the user should be able to print out

CØST(1,1) + CØST(1,2) + RENT(3).

A limited version of this output package is implemented and is described later.

## II. General Description

New GAMES is a Fortran IV program designed to simulate the activities of major sectors involved in the harvesting and marketing of renewable resources. The sectors simulated by the program are: products, markets, processors, harvesters, regulators, stocks, and locations. A specific resource system is modeled by stipulating through input data to the program the number of entities in each sector and the links between them.

The following figure illustrates the conceptualization of a specific system. The names on the left are the sectors, the numbers in the boxes are the 1st, 2nd, 3rd, etc. entities in that sector, and the arrows are logical links specified by the user. In this example there are 1 product, 2 markets, 2 processors, 1 regulator, 2 harvesters, 1 location, and 2 stocks.

A GAMES model may be used either with or without the active participation of players. Players may assume the role of certain of the entities in the system, making periodic decisions for each of the processors, regulators, and harvesters. In this case, monthly decisions for these entities are input to the program through data cards. Alternatively, periodic decisions for these entities may be obtained from user-generated subroutines. All major components of the computer program are coded as separate subroutines so that alterations to the basic operation of the program can be made easily should that be required by the peculiarities of a specific system to be modeled.

The general operation of the model is as follows. The stock sector "grows" the resources and determines the amount of each stock which is available for harvest in each location. Harvesters operate in locations of their choosing and collect a portion of the available stocks. Harvesters sell quantities of the stock which they have collected to the processors which are offering to buy these stocks. The processors convert the stocks which they have purchased into finished goods and offer them for sale in the markets which are available. Demand and the marketing activities of the processors determine sales by each processor of each product in each market. The regulators impose restrictions of various types on the activities of the processors and harvesters. During the operation of the model, the program compiles statistics on the operation of the system and prints out summaries of these statistics as desired.

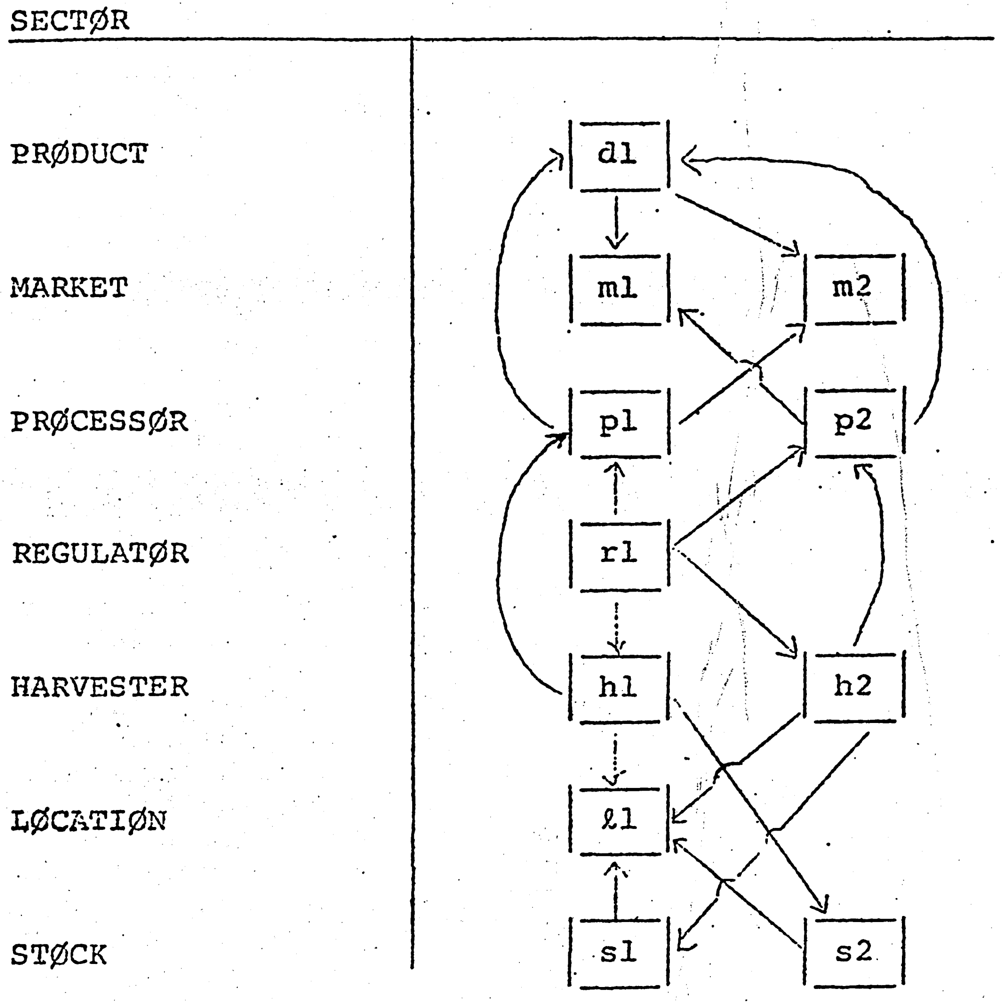

SECTØR
PRØDUCT
d1
MARKET
m1
m2
PRØCESSØR
p1
p2
REGULATØR
r1
HARVESTER
h1
h2
LØCATIØN
ℓ1
STØCK
s1
s2

Figure No. 1

The entities that make up each sector are briefly described below. A product is any material sold by a processor. It may be linked to any market. A market is an outlet for products produced by processors. Associated with each product in each market is a demand curve that determines the amount of potential demand generated for a product offered at a given price. Each demand curve may be shifted left or right by marketing decisions made by the processors.

A processor is a managerial entity that operates in one physical location (that is, a location specified by a pair of x and y coordinates). Processors buy stocks (raw materials) from harvesters, transform them into finished products, and sell them in markets. Each processor has a share of the market for a product that depends on his previous market share and his marketing expenditures and product price relative to the marketing expenditures and product price of other processors linked to the same product and market. The principal variable costs of a processor are the direct labor costs, costs of buying raw materials, operating costs, and direct cost of sales. A processor may be linked to any market and to any product.

A harvester is a managerial entity that operates in one physical and logical location. Each harvester may own a number of harvesting groups, where his $i^{th}$ group harvests stock i and stock i only (for that reason the terms "harvester group" and "stock" are sometimes used interchangeably). Each harvesting group operates as a semi-independent unit constrained only by links to a location and one set of processors. At the start of each month, a group moves from its initial location to the harvesting location. The group harvests there for a specified number of days and then sells its stock to the processor which results in maximum profit to the group. If the processor cannot absorb the group's total supply, the group sells what it can and then travels to the next most profitable processor, and so on. A group may only sell to that set of processors linked to the harvester that owns the group. Any harvest unsold at month's end is recorded and discarded and has no economic value. Each harvester group starts off each month with zero catch (that is, harvesters cannot accumulate a stock supply for more than a month's time). The harvesting time for each group is limited not only by harvester and regulator decisions, but by the group's harvesting capacity. If it starts to exceed capacity during a trip its harvesting time is automatically reduced. The primary operating costs for a harvester are distance costs, time costs, harvest proportional costs, and license fees. A harvester may be linked to any processor or stock and to any one location at a time. It may alter its physical and/or logical location from month to month but it may never be linked to more than one location at a time.

A regulator is an agency that imposes restrictions or fees on the activities of harvesters or processors. A regulator may set

(1) license fees,
(2) annual harvesting quotas,
(3) size limits,
(4) gear efficiency limits,
(5) effort limits (i.e., no. of harvesting units),
(6) operating limits for processors,
(7) harvesting seasons,
(8) monthly harvesting quotas.

All regulations are set for a given stock in a given location except (6) which is for a given stock for a given processor. A regulator automatically reduces any quantities (e.g., gear efficiencies, number of harvesting units, harvesting season, etc.) if the values exceed the regulator determined limits or if the projected annual or monthly catch is too high. A regulator may be linked to any processor or harvester.

A stock is any type of renewable resource (e.g., a species of fish). A stock may be linked to any location, meaning that it may exist in that location. The quantity of each stock available in each location is calculated and, at any given time, it is this quantity which is available to the harvesters.

## III. Detailed Description

This section describes the purpose and organization of the main program and all of its subroutines, and includes a subset of the Fortran vocabulary. The following subscript conventions are used throughout the remaining sections:

d = product<br>
m = market<br>
p = processor<br>
r = regulator<br>
h = harvester<br>
s = stock<br>
ℓ = location<br>
a = age

For example, INVNT(p,d) = the number of units of inventory of product d possessed by processor p.

The main program consists of a number of labelled common blocks followed by a string of subroutine call statements embedded in a double "do-loop" indexed by month and year. Each subroutine in the double loop is executed once a month and reads input, simulates components or interactions between components, or produces output. The double "do-loop" is the principle timing mechanism of the program. All months are of equal length and are about 30.5 days long.

The following is a partial listing of the main program with the purpose of each subroutine indicated opposite the statement that references it:

```
      DØ 110 IYEAR=1,1000
      DØ 100 MØNTH=1,12

      CALL TAPEIN : reads in the default input from TAPE1
      CALL CARDIN : reads in the input from data cards
      CALL PRØCS : makes programmed monthly decisions for the
                   processors
      CALL HARVS:: makes programmed monthly decisions for the
                   harvesters
      CALL REGLS : makes programmed monthly decisions for the
                   regulators
      CALL STØCKS : simulates monthly growth of resources
      CALL HRVST : harvests stocks for one month
      CALL RMARKT : sells raw materials to processors
      CALL PRCES : transforms resources into finished goods
      CALL CMARKT : sells finished goods to final consumers
      CALL STATS : outputs physical reports
      CALL SMSTAT : accumulates and outputs cumulative physical
                    reports

100   CØNTINUE
110   CØNTINUE
```

Subroutine TAPEIN and subroutine CARDIN (an entry point in TAPEIN) are responsible for handling all input. The input is in an English language free-field format* and determines some of the structures and relationships in the model, the characteristics of the major components, the monthly decisions for processors, harvesters, and regulators, and the form and frequency of output. The input is divided into blocks that correspond to logical units in the program. Any input may be changed on a monthly basis but all previous input remains in effect until it is specifically overridden by new input data. Thus, in any given month, the user may change all, some, or none of the input. The input structure corresponds exactly to the "user oriented input package" described in the introduction.

TAPEIN reads the default input from a file named TAPE1 and CARDIN reads the same type of input from cards. The organization and sequencing of the input system are fully explained in section IV. Here, we will only illustrate the block structure of the input and define the variables whose values are assigned by it.

Block 0 : controls the printout of default and card input.

Block 1 : controls the form and frequency of financial and physical output.

Block 2 : controls the number of entities in each sector and the links between entities in different sectors.

Block 3 : defines the sets to which processors and harvesters belong (the financial statements for all elements in a set are summed together and printed out as a single processor or harvester), and determines the processor and harvester sets whose financial statements are summed together and printed out as a single processor-harvester entity.

Block 4 : assigns values to the following marketing parameters:

ALPHA(d,m) - a parameter that controls the sensitivity of market shares to changes in marketing expenditures,

BETA(d,m) - a parameter that controls the sensitivity of market shares to changes in the price of a product listed by a processor,

GAMMA(d,m) - a parameter that controls the sensitivity of the demand curves to changes in marketing expenditures,

QDC(d,m,n),PDC(d,m,n) - the x and y coordinates of points on the demand curves,

TRANSP(p,s,d) - a two dimensional matrix for each processor that transforms raw materials into finished goods,

* See KEYWRD: A FREE FIELD INPUT PACKAGE, Program No. FRG 721

TØTMRK(d,m) - total value of the market,
TØTMNG(d,m) - total marketing expenditures.

Block 5 : assigns values to the following processor parameters:
MSHARE(p,d,m) - market shares,
INVNT(p,d) - no. of units in inventory,
DIN(p,d) - dollar value of inventory,
PTØTPE(p) - total plant and equipment in dollars,
PCØSTU(p) - cost per unit of plant and equipment,
PCASH(p) - initial cash,
PRTAIN(p) - retained earnings,
PBLØAN(p) - bank loans,
PUCAP(p,s) - processing capacity per dollar of plant and equipment,
DLCRAW(p,d) - direct labor cost per unit of finished goods produced by processing raw materials,
PDRATE(p) - depreciation rate,
PFXDXP(p) - fixed (or administrative) costs,
PINTRS(p) - interest rate,
PCAPTL(p) - capital,
PLØCX(p), PLØCY(p) - x and y coordinates of the processor's location,
PLTLIB(p) - long term liabilities,
PØPERX(p,s) - operating cost per unit of raw materials processed.

Block 6 : assigns values to the following harvester parameters:
HPLOCX(h),HPLOCY(h) - the x and y coordinates of a harvester's location at the start and end of each month,
HCASH(h) - initial cash,
HTØTPE(h) - total plant and equipment,
HCØSTU(h) - cost per unit of plant and equipment,
HBLØAN(h) - bank loans,
HCAPTL(h) - capital,
HDRATE(h) - depreciation,
HFXDXP(h) - fixed (or administrative) costs,
HINTRS(h) - interest rate,
HRTAIN(h) - retained earnings,
HLTLIB(h) - long term liabilities,
HVRCPE(h,s) - direct operating cost per day per harvesting unit,
HVRDST(h,s) - cost per unit distance traveled per harvesting unit,
HRCØST(h,s) - direct harvesting cost per unit harvested,
DINH(h,s) - dollar value of inventory.

Block 7 : assigns values to the stock sector of the model. The following variables are used with the Peruvian anchovy model; in general different stock models will use different variables. Some of the input parameters act as multipliers and are used to scale the values of other parameters. They are marked with a '+' and must precede the parameters they multiply.

AGEMAX(s) - the number of ages for stock s,
NØGRND(s) - the number of spawning grounds for stock s,
SPGRND(s,ℓ) - the index of spawning grounds for stock s in location ℓ,
RECRTLT(s) - length of recruits for stock s (in cm.),
SURVN(s) - the normal egg survival rate,
RECINT(s) - the interval from spawning to recruitment, in months,
AVFR(s,ℓ,mo) - the fraction of fish that are available for harvesting in month "mo",
NMØRT(s,a) - natural mortality rate,
LINFN(s) - asymptotic length,
VBK(s) - a von Bertalanffy equation parameter,
TZERØ(s) - a von Bertalanffy equation parameter,
VBQ(s) - a length weight multiplier,
ALNGTH(s,a,ℓ) - length of an individual fish (in cm.),
WEIGHT(s,a,ℓ) - weight of an individual fish (in metric tons),
LMULT(s,k) - length increment multiplier,
WINF(s,ℓ) - maximum stock biomass in metric tons,
FRMATFL(s,k) - length of females as a function of spawning fraction,
FRMATML(s,k) - length of males as a function of spawning fraction,
FFISH$^+$ - a factor which multiplies the numbers of fish (CENSUS,SPFEMN),
FEGGS$^+$ - a factor which multiplies the numbers of eggs (EFEM),
FCØEF$^+$ - a factor which multiplies the coefficient of catchability (QCATCH),
FRECR$^+$ - a factor which multiplies the numbers of recruits (RECRT),
RECRT(s,g,mo) - the no. of recruits in spawning ground g, in month "mo",
NCYCL(s,mo) - the number of times per month to cycle the stock sector of the model,
SPFEMN(s,g,mo) - the maximum number of spawning females in months "mo",
MARK(s,mo) - the marking fraction,
CENSUS(s,a,ℓ) - the number of fish,
FRFEM(s,a) - fraction which is female,
FRMATF(s,a) - female mating fraction,
FRMATM(s,a) - male mating fraction,
EFEM(s,a) - eggs per female,
WF(s,a) - weighting vector,
FRSPAWN(s,mo) - fraction of spawners that spawn in a given month,
QCATCH(s,a) - coefficient of catchability,
SMULT(s,k) - egg survival multiplier.

Block 8 : reads in the following processor decisions:
ØPLEVI(p,s) - input value of operating level,
PADDPE(p) - additional plant and equipment,
PRANDD(p) - research and development expenditures,
MRKTNG(p,d,m) - marketing expenditures,
PURPRI(p,s) - purchase price of raw materials,
PRICES(p,d,m) - selling price of finished products,
PRLØAN(p) - loan repayment.

Block 9 : reads in harvester decisions:
HADDPE(h) - additional plant and equipment,
HRANDD(h) - research and development expenditures,
EFFDI(h,s) - the number of harvesting units,
FPWDI(h,s) - the average harvesting efficiency per unit,
HDELI(h,s) - the number of days per month spent harvesting,
HCAPAC(h,s) - the maximum capacity per harvesting unit per trip,
DYSTRP(h,s) - the time in days for each harvesting trip,
HRLØAN(h) - loan repayment,
HLØCX(h), HLØCY(h) - the x and y coordinates of the harvesting location.

Block 10 : reads in regulator decisions:
CQUØTA(s,ℓ) - annual harvesting quota,
HLIN(s,ℓ) - license fees for harvesters,
SLIMIT(s,ℓ) - size limits in terms of cm for harvested stock (that is, stock below a certain size may not be harvested),
EFFL(s,ℓ) - harvesting effort limit,
FPWL(s,ℓ) - harvesting efficiency limit,
RDELT(s,ℓ) - harvesting season,
XI(s,ℓ) - a set of multipliers to be applied to the catch quotas,
CMQ(s,ℓ) - monthly harvesting quota,
ØPLL(p,s) - maximum operating levels for processors,
NAUT(1) - a parameter that determines whether or not the annual harvesting quota is enforced; if NAUT(1) = 0 it is enforced, if NAUT(1) = 1 it is not,
NAUT(2) - if NAUT(2) = 0, the monthly harvesting quota is enforced; if NAUT(2) = 1 it is not.

Subroutine PRØCS makes (programmed) monthly decisions for the processors. In the current model, PRØCS does nothing since all processor decisions are obtained from cards.

- Subroutine HARVS makes (programmed) monthly decisions for the harvesters. In the current model, HARVS does nothing since all harvester decisions are obtained from cards.

- Subroutine REGLS makes (programmed) monthly decisions for the regulators. It computes

  HDELT(h,s) - no. of days per month spent harvesting,
  EFFD(h,s) - no. of harvesting units,
  FPWD(h,s) - gear efficiency per unit.

It computes HDELT(h,s) as follows: First, it sets DELTAT(s,ℓ) = RDELT(s,ℓ). Then it determines the total current harvesting effort. Based on current eff: rt and harvesting season it predicts the harvest for the current month. If the predicted harvest exceeds the annual or monthly quota, it reduces DELTAT(s,ℓ) by the appropriate amount. Then it sets

$$\text{HDELT}(h,s) = \min(\text{HDELI}(h,s),\text{DELTAT}(s,\ell)\times\text{XI}(s,\ell))$$

for each harvester operating in location ℓ. It computes EFFD(h,s) and FPWD(h,s) as follows. Let the phrase "h in ℓ" mean "those harvesters operating in location ℓ". Let

$$t = \sum^{h}\text{EFFDI}(h,s),\ \text{for "h in } \ell\text{", and } f = \text{EFFL}(s,\ell).$$

If t > f then

$$\text{EFFD}(h,s) = (f/t)\times\text{EFFDI}(h,s),\ \text{for "h in } \ell\text{".}$$

If t ≤ f then

$$\text{EFFD}(h,s) = \text{EFFDI}(h,s),\ \text{for "h in } \ell\text{".}$$

Also FPWD(h,s) = min(FPWDI(h,s),FPWL(s,ℓ)) for "h in ℓ".

- Subroutine STØCKS simulates the growth of different stocks (renewable resources). The main parameters it needs from the rest of the model are EFFD(h,s) (the number of harvesting units), HDELT(h,s) (the harvesting time), HCAPAC(h,s) (capacity per unit per harvest group per trip), FDWD(h,s) (harvesting efficiency), and NCYCL(s,m) (the number of times per month to cycle a given stock). The output is the harvest matrix CATCH(s,a,h) which gives the harvest by stock, age, and harvester in tons. The nature of STØCKS depends heavily on the type of resource being simulated and so will not be described here.

- Subroutine HRVST computes the following arrays:

QUANTH(h,s) - quantity (in metric tons) of stock s obtained by harvester h,
CCATCH(s,ℓ) - harvest accumulated from the month of January,
HVCØST(h) - harvest proportional costs,

where

$$\mathrm{QUANTH}(h,s) = \sum^{a}(\mathrm{CATCH}(s,a,h)\times \mathrm{WF}(s,a)),$$

$$\mathrm{CCATCH}(s,\ell) = \mathrm{CCATCH}(s,\ell) + \sum^{h}\left(\sum^{a}\mathrm{CATCH}(s,a,h)\right),$$

$$\mathrm{HVC\O ST}(h) = \sum^{s}(\mathrm{QUANTH}(h,s)\times \mathrm{HRC\O ST}(h,s)).$$

The array CATCH(s,a,h) is the harvest (in metric tons) of stock s, age a, taken by harvester h, and WF(s,a) is an array of weighting factors. The array CCATCH is set to zero at the beginning of each January.

- Subroutine RMARKT simulates the sale of raw materials to the processors in the following manner. Each harvester group acts as a semi-independent unit, constrained only by those links to a location and a set of processors established for the harvester that owns the group. At the start of a month, a group moves from an initial location to the harvesting location. It harvests there for a specified number of days and then sells its stock to the processor which maximizes the profit for the group, where

$$\text{profit} = p\times q - c$$

and p = purchase price offered by the processor per unit of stock, q = QUANTH(h,s) = quantity of stock currently possessed by the harvester group, and c = the cost of travel from the group's current location to the processor's location. If the processor cannot absorb the group's full supply, the group sells what it can and then travels to the next most profitable processor, and so on. Each sale reduces the harvest matrix QUANTH(h,s) and increases the processor-stock matrix QUANTP(p,s), by the number of units sold. If the profit on the above transaction is negative or zero, the group does not sell.

The subroutine also keeps track of the distance covered by each harvester group. HDIST(h,s) is the distance covered by group s in moving from the initial location to the harvesting location, to the location in turn of each of the processors buying stock from group s, and back to the initial location. Thus a group always starts and ends a trip in the same place. The total distance covered by group s of harvester h in one month is n×HDIST(h,s), where n is the number of trips per month.

. Subroutine PRCES transforms the raw material obtained by the processors into finished goods. It does this by multiplying the raw materials matrix QUANTP(p,s) by a transformation matrix TRANSP(p,s,d) and adding the result to the inventory matrix INVNT(p,d). That is,

$$V + V + Q \times T(p)$$

where V = INVHT(p,d),
Q = QUANTP(p,s),
T(p) = TRANSP(p,s,d), the transformation matrix for processor p.

. Subroutine CMARKT sells the finished goods to the final consumers. This process is simulated in the following manner. Steps A. through E. are executed for each product and each market.

A. Shift the demand cutve.
There is a quantity-price demand curve (with "price" as the ordinate and "quantity" as the abscissa) for each product in each market. The curve is shifted by the amount Δ where

$$\Delta = \lambda\left(\frac{\text{change in total marketing}}{\text{total market}}\right)$$

and where λ is the array GAMMA(d,m) input in the subroutine TAPEIN.

B. Compute weighted average price.

$$\text{Weight. Ave. Pr.} = \sum(\text{prices})(\text{market shares})$$

where the summation is over the processors.

C. Compute the change in market shares for each processor.

$$\text{share}_t = \text{share}_{t-1} + \Delta 1 + \Delta 2$$

where

$$\Delta 1 = \alpha\left(\frac{\text{marketing by this processor}}{\text{ave. marketing}} - 1\right)$$

$$\Delta 2 = \beta\left(1 - \frac{\text{Price set by this processor}}{\text{Weighted ave. price}}\right)$$

and where α and β are the parameters ALPHA(d,m) and BETA(d,m) input in the subroutine TAPEIN.

D. Obtain the total demand from the demand curve based on the weighted average price.

E. Compute potential of demand for each processor based on the total demand and the processor's market share.

F. Since some processors may not have sufficient inventory to meet their share of the demand, we must distribute some of the excess demand to other processors. This is simulated by a three step process that is executed for each product in turn:

1. Reduce the demand share of those processors lacking sufficient inventories. Temporarily store the excess demand in a pool for each market.
2. Distribute the excess demand in each market pool to those processors having excess inventories, in proportion to their market shares.
3. Now go back and reduce the demand share of those processors who just increased their inventories by absorbing some of the demand in the market pools, if their augmented demand exceeds their current inventories.

G. Reduce the total demand by the amount that cannot be filled for lack of inventory.

H. Compute the new market shares, inventories, and total market values based on the re-computed (actual) demand shares.

Subroutine STATS computes and outputs reports covering the physical and financial activities of various sectors in the model. The physical reports cover the activities, for the current month, of individual entities in the stock, market, processor, or harvester sectors. The financial reports contain an income statement, statement of cash flow, and a balance sheet for each processor set, harvester set, or processor-harvester set, where the financial statistics for harvesters or processors in each set are summed together.

Among the statistics computed by STATS are the variable costs for processors and harvesters. The principle variable costs for processors are:

$$\text{direct labor cost} = \sum^{d}(G(d)\times C(p,d)),$$

$$\text{cost of buying raw materials} = \sum^{s}(\Omega(p,s)\times P(p,s)),$$

$$\text{operating cost} = \sum^{s}(\Omega(p,s)\times \emptyset(p,s)),$$

$$\text{direct cost of sales} = \sum^{d}(A(d)\times \sum^{m} S(p,d,m)),$$

where

$G(d)$ = the increase in inventory for product d from buying and transforming raw materials,

$C(p,d)$ = the labor cost per unit output of finished goods produced by processing raw materials,

$Q(p,s)$ = quantity of stock s possessed by processor p,
$P(p,s)$ = purchase price offered by processor p per unit of stock s,
$\emptyset(p,s)$ = operating cost per unit of raw materials processed,
$A(d)$ = average value of an item of product d in the processor's inventory,
$S(p,d,m)$ = no. of items of product d sold in market m by processor p.

The principle variable costs for a harvester are:

$$\text{distance costs} = \sum^{s}(D(h,s)\times P(h,s)\times N(h,s)\times H(h,s)),$$

$$\text{time costs} = \sum^{s}(T(h,s)\times N(h,s)\times C(h,s)),$$

$$\text{harvest costs} = \sum^{s}(Q(h,s)\times R(h,s)),$$

$$\text{license fees} = \sum^{s}(N(h,s)\times L(h,s)),$$

where
$D(h,s)$ = distance covered per trip by group s of harvester h,
$P(h,s)$ = number of trips per month (group s, harvester h),
$N(h,s)$ = number of harvesting units (group s, harvester h),
$H(h,s)$ = cost per unit distance traveled (group s, harvester h),
$T(h,s)$ = number of days spent harvesting (group s, harvester h),
$C(h,s)$ = cost per day (group s, harvester h),
$Q(h,s)$ = quantity harvested (group s, harvester h),
$R(h,s)$ = cost per unit harvested (group s, harvester h),
$L(h,s)$ = license fee (group s, harvester h).

• STATS calls the subroutines FNDSET (which locates the sets to which harvesters or processors belong), FINACC (which accumulates the financial statistics), and PRNSTA (which prints out the financial statements). The parameters which control the frequency of output and other characteristics are explained in section V and the output format is illustrated in appendix V.

• Subroutine STATS2 prints out monthly reports on the activities of the stock, market, processor, and harvester sectors. The output is sent to the printer and (optionally) to a disk file named TAPE2 for further use. The frequency and format of output are explained in section V and illustrated in Appendix IV.

- Subroutine SMSTAT selects user-specified variables from the stock, market, processor, and harvester sectors, combines them into user-specified groups, accumulates them over user-specified times, and prints them out in tabular form. The frequency and format of the output are explained in section V.

- The following routines are also in the program:

TABLE: a function that linearly interpolates between points in the function Y = Y(x).

SETDCT, INPTC, CDPARS: three subroutines that make up the English language free-field input package used for all input. For a complete description see

KEYWRD: A FREE FIELD INPUT PACKAGE,
Program No. FRG 721

## IV. Input

The input package should be self-documentary, position tolerant, self-scheduling, must override previous input, and must possess default values. These features are all present in the following default and card input package.

. The default part reads the following input from a file named TAPE1:

```
{Group 1}
-header card for group 2-
{Group 2}
-header card for group 3-
{Group 3}
    .
    .
    .
-header card for group n-
{Group n}
End-of-File
```

where n ≥ 0 (if n = 0, there is no default input). The header card that precedes each group (except group 1) is of the form

```
DEFAULT INPUT FØR MØNTH a AND YEAR b
```

and specifies the (simulated) month and year in which the group is read. The data cards that appear in the current default group override data cards of the same type that appeared in any previous default or card input group. Each group in turn is divided into 10 blocks as follows:

```
{Block 1}
-blank card-
{Block 2}
-blank card-
    .
    .
    .
{Block 10}
-blank card-
```

Each block contains a comment card to identify the block, followed by any number (including zero) of data cards belonging to the block. For example, block 9 might be punched as follows:

```
C      -----------BLØCK 9 : HARVESTER DECISIØNS.
HADDPE  ADDITIØNAL PLANT AND EQUIPMENT FØR HARVESTERS 1 *TØ* 3 = 0
HCAPAC  CAPAC. PER BØAT PER TRIP, FØR HARV 1, STØCK 1 = 500 TØNS
```

A comment card is indicated by a "C" in column 1 followed by a blank in column 2. Comment cards may be present anywhere in a block. The position of cards within a block does not normally matter. A block with no information is represented by the identifying comment card only:

```
C     -----------BLØCK 10 : REGULATØR DECISIØNS.
```

Except for block 1, all data cards are of the form:

Cols. 1-8 : the name of the input variable to be affected, left adjusted,
Cols. 9-80: free form text and numbers punched according to the rules of the KEYWRD package.

Block 1 specifies the output options and is entirely free form.

The card input part reads the following input from the standard input file:

```
{Group 0}
-blank card-
{Group 1}
-header card for group 2-
{Group 2}
-header card for group 3-
{Group 3}
     :
     :
-header card for group m-
{Group m}
End-of-File
```

where m ≥ 1. Group 0 controls the printout of default and card input and may contain any number of comment cards plus the following two control cards:

```
     PRINT ØUT DEFAULT INPUT
(or SUPPRESS PRINTØUT ØF DEFAULT INPUT)
     PRINT ØUT CARD INPUT
(or SUPPRESS PRINTØUT ØF CARD INPUT)
```

The header card that precedes each group (except group 0 and group 1) is of the form

```
CARD INPUT FØR MØNTH a AND YEAR b
```

and specifies the (simulated) month and year in which the group is read. Except for group 0, all groups are identical to the default groups in form and content.

The input sequence is as follows:

1. card group 0 (must be present).
2. default group 1 (if present)
3. card group 1 (must be present)

From then on the program reads whatever group, card or default, is scheduled next. If a default group and a card group are scheduled to be read at the same time (i.e., the header cards for both groups contain the same month and year) then the default group is read first. The program stops when it encounters an end-of-file in the card input stream or when it exceeds the termination date specified in the output control cards of block 1.

The user should note that the program will stop at the start of the second month if he failed to follow card group 1 with a header card for group 2 even if card group 2 is absent. This is because the program pre-reads the first card of the next group in order to know when to read the rest of the group. If the header card is absent the program will read an end-of-file and stop. Therefore, the user must always follow his last card group with a suitably dated header card.

## V. Output

The output package should provide output in the form of relatively small labelled blocks which can be manipulated by the user through a command language in the input data. The commands, which should be flexible and self-explanatory, should allow the user to specify the form, frequency, and destination of output, and should allow him to create new output blocks or modify existing ones.

In new GAMES, the labelled blocks are financial statements and physical reports. The user can control the form only to the extent of specifying when to restore pages before printing; the frequency by specifying the years and months in which to print each individual block; the destination to the extent of directing some or all of the physical reports to a file named TAPE2 as well as the printer; and he can create new blocks by summing the financial statements for specified harvesters or processors, and by summing together specified variables in the physical reports. With one exception, all output controls are specified in block 1 of the input.

### A. Financial reports.

#### 1. Monthly reports.

Financial reports consist of an income statement, statement of cash flow, and a balance sheet for all processor sets, harvester sets, and processor-owned harvester sets for a specified month (for the definition of processor, harvester, and processor-owned harvester sets, see F. below). The user specifies the printout times through data cards of the form:

```
FINANCIAL REPØRTS ARE PRINTED FØR YEARS <list of years>
FINANCIAL REPØRTS ARE PRINTED FØR MØNTHS <list of months>
```

#### 2. Cumulative reports.

Cumulative reports sum together all the monthly reports for one or more years at a time. To accumulate reports for one year only, the user includes the following cards in block 1:

```
FINAN. STATS. ARE ACCUMULATED FØR ØNLY  ØNE YEAR AT A TIME
CUMULATIVE FINAN. STATS. ARE PRINTED IN YEAR(S) <list
                                        of years>
```

To accumulate reports from the time of last cumulative printout, he includes the following cards in block 1:

```
FINAN. STATS. ARE ACCUMULATED FROM TIME OF LAST PRINTOUT
CUMULATIVE FINAN. STATS. ARE PRINTED IN YEAR(S) <list of
                                                years>
```

### B. Physical reports.

#### 1. Monthly physical reports.

The physical reports describe activities in the stock, market, processor, and harvester sectors. The user may direct the program to print out any or all of the four reports in any month or year by including the following cards in block 1:

```
PHYSICAL REPØRTS ARE PRINTED FØR YEARS <list of years>

STØCK REPØRTS ARE PRINTED FØR MØNTHS <list of months>
MARKET REPØRTS ARE PRINTED FØR MØNTHS <list of months>
PRØCESSØR REPØRTS ARE PRINTED FØR MØNTHS <list of months>
HARVESTER REPØRTS ARE PRINTED FØR MØNTHS <list of months>
```

2. Cumulative physical reports.

Cumulative physical reports sum together user-specified variables in the stock, market, harvester, and processor sectors for one or more years at a time.

To accumulate reports for one year only, the user includes the following cards in block 1:

```
PHYSICAL REPØRTS ARE ACCUM. FØR ØNLY  ØNE YEAR AT A TIME
CUMUL. PHYSICAL REPØRTS ARE PRINTED FØR YEARS <list of years>
```

To accumulate reports from the time of last cumulative printout, he includes the following cards in block 1:

```
PHYSICAL REPØRTS ARE ACCUM. FRØM TIME ØF LAST  PRINTØUT
CUMUL. PHYSICAL REPØRTS ARE PRINTED FØR YEARS <list of years>
```

The user selects subscripted variables and combines them into groups. He assigns a unique number to each variable and a group number to each subset of variables, where the subsets need not be disjoint. The input is of the form:

```
VARIABLE 1 IS <variable name> (<subscript list>)
VARIABLE 2 IS <variable name> (<subscript list>)
VARIABLE 3 IS <variable name> (<subscript list>)
          .
          .
          .
GRØUP 1 CØMBINES VARIABLES <list of variables in 1st subset>
GRØUP 2 CØMBINES VARIABLES <list of variables in 2nd subset>
          .
          .
          .
```

The program sums together the variables within a subset and prints out tables of the form:

```
-*-*-*-*-* *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-

///YEAR: year /// ANNUAL SUMMARY TABLE FOR PHYSICAL REPORTS

GROUP      JAN  FEB  MAR ... DEC   ROW SUM    ROW AVE
GROUP 1     -    -    -       -       -          -
GROUP 2     -    -    -       -       -          -
   .
   .
   .
GROUP 10    -    -    -       -       -          -

COL SUM     -    -    -       -       -          -
COL AVE     -    -    -       -       -          -
-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
```

at user specified times. The maximum number of groups is 10. The following table defines the variable names that may be used. The subscripts, which are given symbolically in the table, must be punched as numbers on the data cards.

| variable name | meaning |
|---|---|
| CENSUS(s,a,ℓ) | population of fish of stock s, age a, loc. ℓ |
| WEIGHT(s,a,ℓ) | weight of fish of stock s, age a, loc. ℓ, in metric tons |
| ALNGTH(s,a,ℓ) | length of fish of stock s, age a, loc. ℓ, in cm |
| CATCH(s,a,ℓ) | catch by weight of fish of stock s, age a, loc. ℓ |
| TØTMRK(d,m) | total value of market for product d in market m |
| TØTMNG(d,m) | total marketing expenditures for product d in market m |
| QUANP(p,s) | tons of stock s bought by processor p |
| FGØØDS(p,d) | tons of product d added to processor p |
| INVNT(p,d) | no. of inventory units of product d possessed by proc. p |
| PSALES(p,d,m) | no. of units of product d sold by proc. p in market m |
| PRICES(p,d,m) | selling price of prod. d, in market m, by proc. p |
| MSHARE(p,d,m) | share of mark. of proc. p selling prod. d in mark. m |
| MRKTNG(p,d,m) | mark. expend. by proc. p for prod. d in mark m |
| SQNTH(h,s) | units of stock s harvested by harvester h |
| HSALES(h,s,p) | units of stock s sold by harv. h to proc. p |
| QUANH(h,s) | units of stock s left unsold by harv. h |
| EFFD(h,s) | no. of harv. units for harv. h, stock s |
| TDIST(h,s) | dist. traveled by harv. h fishing for stock s |
| HDELT(h,s) | no. of days spent harvesting by harv. h for stock s |

Examples:

```
VARIABLE 1 IS CENSUS( 1,1,1 )
VARIABLE 2 IS CENSUS( 1,1,2 )
VARIABLE 3 IS CENSUS( 1,1,3 )
VARIABLE 4 IS CATCH( 2,1,1 )
VARIABLE 5 IS CATCH( 2,2,1 )
VARIABLE 6 IS CATCH( 2,3,1 )
VARIABLE 7 IS EFFD ( 1,1 )
VARIABLE 8 IS EFFD ( 2,1 )
VARIABLE 9 IS EFFD ( 3,1 )
GRØUP 1 CØMBINES VARIABLES 1 *TØ* 3
GRØUP 2 CØMBINES VARIABLES ***( 1,3 )
GRØUP 3 CØMBINES VARIABLES 4 *TØ* 6
GRØUP 4 CØMBINES VARIABLES 7 *TØ* 9
GRØUP 5 CØMBINES VARIABLES ***( 7,8 )
GRØUP 6 CØMBINES VARIABLES ***( 7,9 )
```

C. Graphical output (not yet implemented).

Currently, the user may print out any or all of the physical reports on a file named TAPE2 by including any of the following cards in block 1:

```
STØCK REPØRTS ARE PRINTED ØN TAPE2
MARKET REPØRTS ARE PRINTED ØN TAPE2
PRØCESSØR REPØRTS ARE PRINTED ØN TAPE2
HARVESTER REPØRTS ARE PRINTED ØN TAPE2
```

The purpose of TAPE2 is to save the output for use by other programs which will scan it in order to generate tables or graphs. The mechanism for writing the reports on TAPE2 is in the program now but the programs to process it are not yet available.

D. Page restore.

The user may restore a page before printing any output blockby including the following cards in block 1:

```
RESTØRE THE PAGE FØR EACH REPØRT FØR YEARS <list of years>
RESTØRE THE PAGE FØR EACH REPØRT FØR MØNTHS <list of months>
```

E. Zeroing out output controls.

It was previously stated that all input could be changed on a monthly basis by cards that appear in a later group. However, for block 1 it is best to zero out the output controls and follow with all new controls. This can be done with the card:

```
ZERØ ØUT ALL PREVIØUS ØUTPUT CØNTRØLS
```

F. Processors, harvesters, and processor-owned harvester sets.

Processors and harvesters may be combined into sets and printed out as a unit. Harvester sets and processor sets may also be combined under the heading processor-owned harvester sets. The financial statements for all entities in a set arc summed together. All processors and harvesters must reside in at least one set but the sets need not be disjoint. The user defines sets by including cards of the form:

```
NPSET   PRØC. SET 1, ENTRIES 1 *TØ* 2, CØNTAINS PRØC. 1 *TØ* 2
NHSET   HARV. SET 2, ENTRY 1, CØNTAINS HARVESTER 3
NHPSET  HARVESTER SET 2 IS ØWNED BY PRØC. SET 1
```

in block 3 (not block 1).

## VI. Variable Dimensions

Global variables with variable dimensions are desirable because of the large number of subscripted variables in the program and the need for frequent dimension changes. Since the Fortran language lacks this feature, we must implement it by means of a preprocessor.

Consider the format of a typical Fortran subroutine:

```
        SUBRØUTINE name (<argument list>)
        {any number of labelled common blocks}
        {code}
        RETURN
        END
```

Each common block is of the form:

```
        CØMMØN/name/<list of variables>
        <type declarations>(if any)
```

In large programs, the argument list is usually empty and all non-local variables are in common.

It would be nice if we could include common blocks in subroutines merely by listing their names without their associated variables, dimensions, and type declarations. For example, suppose SUB1 is a subroutine using variables in three common blocks named CØM1, CØM2, and CØM3. Then SUB1 could be punched as follows:

```
        SUBRØUTINE SUB1
B       CØM1, CØM2, CØM3
        {code}
        RETURN
        END
```

where the "B" in column 1 denotes a non-Fortran type statement. Such a language extension would greatly reduce the problem of dimension changes since dimensions need appear but once in a program.

The following preprocessor* provides such capabilities and more. It is written for the CDC 6400 in Fortran and SNØBØL and transforms a symbolically dimensioned set of common blocks and a pseudo-Fortran program into a legal Fortran program with suitable dimensions. The preprocessor requires four things:

* FRG 736 : ALTER - A Code Modification Program.

1. a list of all common blocks used in the program (henceforth called the *Variable Dimension Block*) with some or all dimensions in symbolic form. Any type declarations for variables in a block must immediately follow the block, e.g.

   ```
   CØMMØN/CØM1/ALPHA(S,A,3),IR(H,M),NUM(H,R,6)
   REAL IR,NUM
   ```

2. a list of equivalences (henceforth called the *Equivalence Block*) between symbolic dimensions and the numeric dimensions currently desired, e.g.

   ```
   S=10 $ A=5
   H=4 $ M=12
   ```

3. the pseudo-Fortran program with all common blocks replaced by statements of the form:

   ```
   B     name1,name2,...,namek
   ```

   where "namei" is the name of a common block, e.g.

   ```
   B     CØM1,CØM2,CØM3.
   ```

   Such a program is incorrect as it stands and would never compile.

4. a set of four routines, ALTER, MKMAC, MACRØ, and MØDFØR which are stored on tape and activated by a catalogued procedure call. ALTER is a CDC control card program that directs the sequence of program executions; MKMAC is a Fortran program that reformats the Equivalence Block and merges it with the Variable Dimension Block in a form suitable for input to MACRØ; MACRØ is a SNØBØL program that replaces those symbols in the Variable Dimension Block which also appear to the left of an equal sign in the Equivalence Block with their right hand numeric equivalents; and MØDFØR scans the pseudo-Fortran program for "B" statements, replaces them with the appropriate common blocks, and writes out a legal Fortran program with the dimensions currently desired.

Actually, the system just described is much more general in that it can modify and insert named blocks of code (not just labelled common blocks) and can carry out macro substitutions with parameter transfers. However, a description of its other capabilities would only serve to complicate the discussion.

In practice, the pseudo-Fortran program is usually replaced by an UPDATE correction deck. The UPDATE deck contains directives to delete cards in the Fortran deck followed by the appropriate "B" cards. The correction deck is modified by MØDFØR and merged with the program via the UPDATE system. This approach has two advantages:

1. it is faster since MØDFØR need scan only the UPDATE deck which is far smaller than the program,

2. one can modify an *existing* program by deleting the existing common blocks and inserting the appropriate "B" cards in the UPDATE correction deck.

The setup needed to redimension a program is:

```
{CDC control cards}
7-8-9
{Equivalence Block}
7-8-9
{Variable Dimension Block}
7-8-9
{UPDATE correction deck}
6-7-8-9
```

where the control cards contain requests for a) the tape which contains the programs ALTER, MKMAC, MACRØ and MØDFØR, b) the tape that contains the program to be redimensioned, and c) the tape on which the redimensioned program is to be written. The setup needed to redimension GAMES is given in Appendix II.

## APPENDIX I.

The modifications made to old GAMES include minor changes in computation and program structure and major changes in I/O.

A. Error corrections.

1. the statement STATS.200 was changed from

```
        IF(.NØT.PMLINK(I,K).ØR..NØT.MPLINK(J,K))GØ TØ 175
```

   to

```
        IF(.NØT.PMLINK(I,K).ØR..NØT.MPLINK(J,K))GØ TØ 180
```

2. the statement STØCKS.206

```
30    CAPAC(H,S)=EFFD(H,S)*HCAPAC(H,S)*HDELT(H,S)
                *DMØ/NCYCL(S,M)*1.0E6
```

   was removed and replaced by the statements

```
      TRPNØS=HDELT(H,S)/DYSTRP(H,S)
      CAPAC(H,S)=EFFD(H,S)*HCAPAC(H,S)*TRPNØS/NCYCL(S,M)
30    CØNTINUE
```

3. the statement FEB71.3 was removed

4. the statement G=SPGRND(S,L) was inserted after STØCKS.289

B. All constants associated with subscripts were replaced by variables in order to facilitate automatic redimensioning.

C. All weight measures were converted from grams and metric tons to metric tons only.

D. Structural changes.

1. the main program was replaced by the one displayed near the beginning of section II.

2. the input routines RDSTRU, RDPARA, RDINTL, RDPLAY, STKINP, CØNSTN, ANPRØC, ANHARV, and ANREGL were replaced by subroutines TAPEIN and CARDIN.

3. the subroutines STATS2 and SMSTAT were added.

4. the decision routines PRØCS, HARVS, and REGLS no longer read input but are reserved for programmed decisions made by the processors, harvesters, and regulators.

5. the following routines were deleted: DIJK, DIJ, XØUT, PRØC, PRCS1,...,PRCS10, HARV, HRVT1,...,HRVT10, TAXER, and NNPTS.

6. all input may now be changed on a monthly basis.

## APPENDIX II

The setup needed to redimension GAMES is:

```
<job card with CM=100000$_8$>
REQUEST,TAP,VRN=aaa,IN,HI,FILES=x-(x+3),DI.
REQUEST,TAP2,VRN=bbb,IN,HI,FILES=y,DI.
REQUEST,TAP3,VRN=ccc,OUT,HI,FILES=z,DI.
BEGIN(ALTER,TAP,TAP,INPUT,INPUT,INPUT,OUTX)
REDUCE.
RFL(60000)
REWIND(OUTX)
UPDATE(P=TAP2,I=OUTX,W,L=0,N=TAP3)
REWIND(TAP,TAP2,TAP3)
7-8-9
      H=4 $ S=3 $ P=4 $ D=2 $ M=2
      R=2 $ L=5 $ A=7 $ N=8
      MO=12 $ MPH=4 $ MDS=3
      G=10 $ K=10
7-8-9
C     COMMON BLOCKS FOR ANCHOVEY GAMES
      COMMON/ACCNT1/MSHARE(P,D,M),PRICES(P,D,M),MRKTNG(P,D,M)
      REAL MSHARE,MRKTNG
      COMMON/ACCNT2/INVNT(P,D),TOTMRK(D,M),TOTMNG(D,M),DIN(P,D)
      REAL INVNT
      COMMON/MMODIF/ALPHA(D,M),BETA(D,M),GAMMA(D,M)
      COMMON/DCURVE/QDC(D,M,N),PDC(D,M,N),NPTSDC(D,M)
      COMMON/NSCTRS/NPROD,NMARK,NPROC,NREGL,NHARV,NLOCN,NSTOK
      COMMON/ACCNTP/PRANDD(P),PINVST(P),PMRKNG(P),OPLEVL(P,S),PURPRI(P,S
     1),OPLEVI(P,S)
      COMMON/PECON1/PSALES(P,D,M),PFXDXP(P),POUTLN(P),PTOTPE(P)
      COMMON/PECON2/PCASH(P),PTOTDP(P),PCAPTL(P),PRTAIN(P),PLTLIB(P),PDR
     1CST(P),PCOSTU(P)
      COMMON/PECON3/DLCRAW(P,D),PDRATE(P),POPERX(P,S),PINTRS(P)
      COMMON/PECON4/PBLOAN(P),PADDPE(P),PRLOAN(P),PUCAP(P,S)
      COMMON/HECON1/HSALES(H,S,P),HFXDXP(H),HOUTLN(H),HTOTPE(H)
      COMMON/HECON2/HCASH(H),HTOTDP(H),HCAPTL(H),HRTAIN(H),HLTLIB(H),HVC
     1OST(H),HRCOST(H,S),HFXCPE(H),HVRCPE(H,S),HVRDST(H,S),HCOSTU(H),HOP
     2CST(H)
      COMMON/HECON3/HAVCST(H),HDRATE(H),HOPERX(H),HINTRS(H)
      COMMON/HECON4/HBLOAN(H),HADDPE(H),HRLOAN(H),DINH(H,S)
      COMMON/ACCNTH/HRANDD(H),HINVST(H),HLOCAT(H)
      COMMON/SECTRS/SECTOR(10),NHSET(H,H),NPSET(P,P),NHPSET(MPH,MPH),MON
     1ACC,CBONLY,SPONLY,SPCOMB,PL(4,MPH,15),CF(4,MPH,15),BL(4,MPH,15)
      INTEGER SECTOR
      LOGICAL CBONLY,SPONLY,SPCOMB
      COMMON/LINKS/SLLINK(S,L),HLLINK(H,L),RHLINK(R,H),HPLINK(H,P),PMLIN
     1K(P,M),MPLINK(D,M),PPLINK(P,D),HSLINK(H,S),RPLINK(R,P)
      LOGICAL SLLINK,HLLINK,RHLINK,HPLINK,PMLINK,MPLINK,PPLINK
      LOGICAL HSLINK,RPLINK

      COMMON/INREC/CARDS(80)
      INTEGER CARDS
      COMMON/PARSVAL/LOCFC(40),LNGTH(40),VALUE(40),ITEM
      COMMON/DICTON/N,DCTNRY(40),FP(40),RPT(181),NSWRD
      INTEGER DCTNRY
      COMMON/PLAY/NYEAR1,NYEAR2,YEAR,MONTH,NTAPE,NPRND,NPRNC,NREADD,NCFS
     1O,NCPRO,MONAC1,IOC,IOCON(100),NTP2(4),NRESTR(2),NAMSCR(50,5),GROUP
     2(10,12),DMONTH,DYEAR,CMONTH,CYEAR
      INTEGER YEAR,DMONTH,DYEAR,CMONTH,CYEAR
      COMMON/COSTAC/TOTSLS,DIRCST,MEXPNS,ADMINC,TAXLIC,DEPRTE,RANDD,INTR
     1ST,TCURXP,NETINC,BLOANS,TOTCPV,DLCOST,PBUY,OPCUST,ADDPE,RLOANS,TOT
     2CO,CASHFL,CASH,INVTRY,TOTCAS,TOTPE,TOTDP,TOTFAS,TOTAST,OUTLN,LTLIB
     3,TOTLIB,CAPTL,RTAIN,TOTNW,TOTLNW
      REAL MEXPNS,INTRST,NETINC,INVTRY,LTLIB

      COMMON/QUNDST/QUANTP(P,S),QUANTH(H,S),DYSTRP(H,S),HDIST(H,S),FGOOD
     1S(P,D),SQNTH(H,S)
      COMMON/LOCATN/PLOCX(P),PLOCY(P),HLOCX(H),HLOCY(H),CHLOCX(H,S),CHLO
     1CY(H,S),HPLOCX(S),HPLOCY(S)
      COMMON/TRAN/TRANSP(P,S,D)
      COMMON/FISHES/CENSUS(S,A,L),FRFEM(S,A),FRMATF(S,A),FRMATM(S,A),FRS
     1PAWN(S,MO),EFEM(S,A),SPGRND(S,L),SMULT(S,K),RECRT(S,G,MO),AVFR(S,L
     2,MO),QCATCH(S,A),CATCH(S,A,H),NMORT(S,A),LMULT(S,K),ALNGTH(S,A,L),
     3WEIGHT(S,A,L),WINF(S,L),SLIMIT(S,L),FRMATFL(S,K),FRMATML(S,K),NCYC
     4L(S,MO),MARK(S,MO),SPFEMN(S,G,MO)
      REAL NMORT,LMULT,MARK
      INTEGER SPGRND
      COMMON/FLIFE/AGEMAX(S),NOGRND(S),RECRTLT(S),SURVN(S),RECINT(S),LIN
     1FN(S),VBK(S),TZERO(S),VBQ(S),RATIO2(K)
      REAL LINFN
      INTEGER AGEMAX,RECINT
      -FSHY- SHOULD NOT BE PUT IN ANY ROUTINE EXCEPT -STOCKS-
      COMMON/FSHY/F(H,S),BMASS(S,L),X(K),SPFEM(G),SPMAL(G),TWINF(S,G),EG
     1G(G),CAPAC(H,S),CATCHN(A),NDEAD(A),FM(10),XTRA(A),F2(H),Z(A),Z2(A)
     2,CN(A),CW(H),XN(A),HDELTA(H,S)
      REAL NDEAD
      COMMON/RRCSS/TFEC(S,L),DLT(S,L),DELTAT(S,L),DMD(P),QNH(H),HP(H,P),
     1QUANT(P,M),QNQDC(N),PNPDC(N),FPOOL(P),POOL(M),DI(MDS),FR(MDS),AVC(
     2MDS),GAIN(P),SPL(15),SCF(15),SBL(15),TDIST(S),CCH(A),OPLL(P,S)
      LOGICAL HP
      COMMON/REGLAT/CQUOTA(S,L),EFFD(H,S),FPWD(H,S),TFE(S,L),CCATCH(S,L)
     1,WF(S,A),HCAPAC(H,S),HDELT(H,S),HLICEN(H,S),EFFL(S,L),FPWL(S,L),NA
     2UT(4),HDELI(H,S),RDELT(S,L),XI(S,L),CMQ(S,L),EFFDI(H,S),FPWDI(H,S)
     3,HLIN(H,S)
7-8-9
*IDENT VARD1
*DELETE MAIN.4,MAIN.59
B     ACCNT1,ACCNT2,MMODIF,DCURVE,NSCTRS,ACCNTP,PECON1,PECON2
B     PECON3,PECON4,HECON1,HECON2,HECON3,HECON4,ACCNTH,SECTRS
B     LINKS,INREC,PARSVAL,DICTON,QUNDST,LOCATN,TRAN,PLAY
B     FISHES,FLIFE,FSHY,RRCSS,REGLAT,COSTAC
```

```
*DELETE NEWG.56,NEWG.115
B      ACCNT1,ACCNT2,MMODIF,DCURVE,NSCTRS,ACCNTP,PECON1,PECON2
B      PECON3,PECON4,HECON1,HECON2,HECON3,HECON4,ACCNTH,SECTRS,RRCSS
B      LINKS,PLAY,DICTON,QUNDST,TRAN,FISHES,FLIFE,REGLAT
B      LOCATN
*DELETE REGLS.19,REGLS.43
B      FISHES,FLIFE,REGLAT,PLAY,RRCSS,LINKS,NSCTRS,ACCNTP,PECON1,PECON4
*DELETE HRVST.18,HRVST.46
B      FISHES,FLIFE,REGLAT,LINKS,NSCTRS,QUNDST,HECON2,PLAY
*DELETE RMARKT.29,NEWG.854
B      LINKS,HECON1,HECON2,QUNDST,LOCATN,ACCNTP,NSCTRS,REGLAT
B      FISHES,RRCSS
*DELETE PRCES.16,PRCES.26
B      ACCNT2,QUNDST,TRAN,NSCTRS,LINKS
*DELETE PRCES.16,PRCES.26
B      ACCNT2,QUNDST,TRAN,NSCTRS,LINKS
*DELETE CMARKT.22,CMARKT.35
B      ACCNT1,ACCNT2,MMODIF,DCURVE,PECON1,PECON2,NSCTRS,RRCSS
B      LINKS
*DELETE STATS.45,STATS.96
B      ACCNTH,ACCNT1,ACCNT2,ACCNTP,QUNDST,SECTRS,LINKS,PECON1
B      PECON2,PECON3,PECON4,HECON1,HECON2,HECON3,HECON4,TRAN
B      FISHES,FLIFE,RRCSS,REGLAT,NSCTRS,PLAY,COSTAC
*DELETE FNDSET.17,FNDSET.19
B      NSCTRS,SECTRS
*DELETE FINACC.9,FINACC.15
B      SECTRS,COSTAC
*DELETE PRNSTA.8,PRNSTA.11
B      SECTRS,PLAY
*DELETE STATS2.8,STATS2.39
B      ACCNT1,ACCNT2,DCURVE,PECON1,HECON1,LINKS,NSCTRS,QUNDST
B      TRAN,FISHES,FLIFE,REGLAT,PLAY,RRCSS
*DELETE NEWG.1220,NEWG.1253
B      ACCNT1,ACCNT2,DCURVE,PECON1,HECON1,LINKS,NSCTRS,QUNDST
B      TRAN,FISHES,FLIFE,REGLAT,PLAY
*DELETE NEWG.1399,NEWG.1403
B      PLAY
*DELETE STOCKS.118,STOCKS.166
B      PLAY,LINKS,NSCTRS,FISHES,REGLAT,FSHY,FLIFE,QUNDST
6-7-8-9
```

The routines ALTER, MKMAC, MACRØ, and MØDFØR are on files x, x+1, x+2 and x+3 of tape "aaa", GAMES is on file "y" of tape "bbb", and the redimensioned GAMES is written on file "Z" of tape "ccc". The "BEGIN" control card scans the file "TAP" for the program ALTER, which initiates and controls the activities of the other three routines. The latter read in the Equivalence Block, the Variable Dimension Block, and the UPDATE correction deck from the standard input file, and write out the modified UPDATE deck on a file named ØUTX. ØUTX is subsequently used to update the version of GAMES on tape "bbb".

The Equivalence Block is between the first pair of '7-8-9' cards and defines a particular configuration (a maximum of 2 harvesters, 3 stocks, 4 processors, etc.) which a user can change simply by punching different numbers to the right of each equals sign. The Variable Dimension Block is between the second pair of '7-8-9' cards and the UPDATE correction deck is between the third '7-8-9' card and the '6-7-8-9' card.

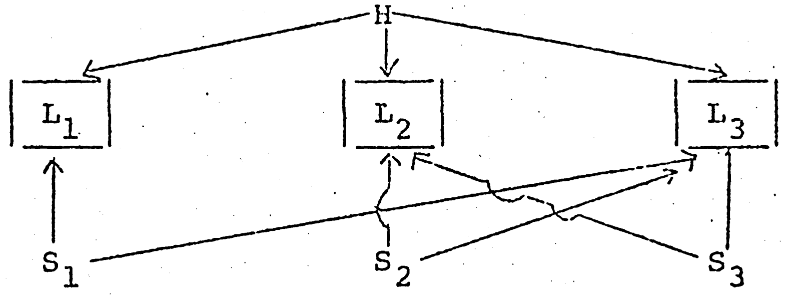
The Equivalence Block is the only thing a user changes in order to redimension GAMES. The necessary symbolic dimensions are defined in the following table:

| symbolic dimension | meaning |
|---|---|
| H | no. of harvesters |
| S | no. of stocks |
| P | no. of processors |
| D | no. of products |
| M | no. of markets |
| R | no. of regulators |
| L | no. of locations |
| A | no. of ages |
| N | no. of points on the demand curve + 1 |
| MØ | no. of months per year (usually=12) |
| MPH | the maximum of P and H |
| MDS | the maximum of D and S |
| G | no. of spawning grounds |
| K | a dimension used in the STOCKS program which currently equals 10 |

## APPENDIX III

GAMES was intended to be as general as possible, but no program can anticipate all situations. A user who intends to model a complex system must be prepared to interpret some system components in non-standard ways and to make minor programming changes by means of the UPDATE system.

Example: a harvester and a set of stocks and locations are linked as follows:



The above configuration is illegal in GAMES because a harvester may be linked to at most one location in a given month (however, he may move to another location in the succeeding month). The user can simulate the configuration by means of three harvesters, each of which is linked to a single location, provided that all three are assigned to the same harvester set (see section V, part F). Then most of the output statistics are summed together and treated as a unit. In general, the logical constraints of the program may require a user to introduce additional entities (harvesters, processors, stocks, etc.) in order to represent a single entity. Fortunately, one can usually combine the output statistics by means of the set and group constructs in the financial and physical reports (section V, parts B.2 and F). The variable dimension feature of GAMES can be extremely useful here.

Example: a harvester is linked to a single location and to several stocks, and allocates fishing effort according to stock biomass. This goes beyond the available logic in GAMES in that fishing effort EFFD(h,s) is obtained from data cards. However, the user may use the UPDATE system to insert code in the routine HARVS which computes EFFDI(h,s) (the input value of EFFD(H,S)) as a function of biomass, where biomass=CENSUS(s,a,ℓ)*WEIGHT(s,a,ℓ).

Example: a user wishes to create output categories that cannot be implemented within the rules of the current output package. This is possible by inserting UPDATE corrections to the routine SMSTAT which assigns values to one or more of the groups defined in section V, part B.2. The general form of the correction deck is:

```
*INSERT NEWG.1324
        GRØUP(i,MØNTH) = {some expression}
```

In general, the user may insert any code in SMSTAT provided it does not conflict with the rest of the routine (for example, any statement number should exceed 300).

For example, a user may observe monthly biomass for the subset consisting of fish of stock 1, age 1, in location 1, and fish of stock 2, age 1, in location 3, by means of the following UPDATE deck:

```
*INSERT NEWG.1334
        GRØUP(1,MØNTH)=CENSUS(1,1,1)*WEIGHT(1,1,1)+CENSUS(2,1,3)
                                    *WEIGHT(2,1,3)
```

In this case, the biomass is assigned to group 1 but it could be assigned to any group n where 1 ≤ n ≤ 10. The expression to the right of the equals sign may involve any variable from the following common blocks: ACCNT1, ACCNT2, DCURVE, PECØN1, HECØN1, LINKS, NSCTRS, QUNDST, TRAN, FISHES, FLIFE, FISHES, REGLAT and PLAY. The variables in the above common blocks are displayed in APPENDIX II.

## APPENDIX IV: Sample Input

The keywords in block 0, block 1, and a few in block 6 are underlined; all others are punched left justified in columns 1-8. Some of the keywords or combinations initiate mutually exclusive actions and cannot be included in the same sample input; these are listed below.

Block 0:

```
      SUPPRESS ØUTPUT ØF DEFAULT INPUT
      SUPPRESS ØUTPUT ØF CARD INPUT
```

Block 1:

```
      FINAN. STATS. ARE ACCUM. FRØM TIME ØF LAST PRINTØUT
      PHYSICAL REPØRTS ARE ACCUM. FRØM TIME ØF LAST PRINTØUT
      ZERØ ØUT PREVIØUS ØUTPUT CØNTRØLS
```

Block 2: the user may "delink" components that were linked by previous data cards by means of the keyword DELINK, e.g.

```
PPLINK  PRØCESSØRS 1 *TØ* 3 ARE DELINKED FRØM PRØDUCT 1
SLLINK  STØCK 1 IS DELINKED FRØM LØCATIØNS 1 *TØ* 3
```

Block 3: only one of the following three keywords may appear in block 3 at any one time: CBØNLY, SPCØMB, SPØNLY

```
CBØNLY  WRITE ØUT CØMBINED REPØRTS ØNLY FØR HARV. AND PRØC.
SPCØMB  WRITE ØUT BØTH SEPARATE AND CØMBINED HARV-PRØC REPØRTS
```

```
C      NEW VERSION OF ANCHOVY STOCK MODEL
C
C      -----------BLOCK 0 : DEFAULT AND CARD OUTPUT CONTROLS
       PRINT OUT DEFAULT INPUT
       PRINT OUT CARD INPUT

C      -----------BLOCK 1 : INPUT-OUTPUT CONTROLS
       THE SIMULATION RUNS FROM 1970 TO 1973
       FINANCIAL REPORTS ARE PRODUCED FOR YEARS ***( 1970,1972 )
       FINANCIAL REPORTS ARE PRODUCED FOR MONTH 7
       CUMULATIVE FINAN. STAT. ARE PRINTED IN YEAR 1970
       FINAN. STATS. ARE ACCUM. FOR ONLY ONE YEAR AT A TIME
       PHYSICAL REPORTS ARE PRINTED FOR YEARS ***( 1970,1971 )
       STOCK REPORTS ARE PRINTED FOR MONTHS ***( 1,2,7 )
       MARKET REPORTS ARE PRINTED FOR MONTH 1
       PROCESSOR REPORTS ARE PRINTED FOR MONTH 1
       HARVESTER REPORTS ARE PRINTED FOR MONTH 1
       CUMULATIVE PHYSICAL REPORTS ARE PRINTED FOR YEAR 1970
       PHYSICAL REPORTS ARE ACCUM. FOR ONLY ONE YEAR AT A TIME
       RESTORE THE PAGE FOR EACH REPORT FOR YEARS 1970 *TO* 1971
       RESTORE THE PAGE FOR EACH REPORT FOR MONTHS 1 *TO* 12
C
C      THE FOLLOWING VARIABLES ARE TO BE COMBINED
C      INTO GROUPS AND PRINTED OUT IN THE PHYSICAL SUMMARIES
       VARIABLE 1 IS CENSUS( 1,1,1 )
       VARIABLE 2 IS CENSUS( 1,1,2 )
       VARIABLE 3 IS CENSUS( 1,1,3 )
       VARIABLE 4 IS CATCH ( 1,1,1 )
       VARIABLE 5 IS CATCH ( 1,1,2 )
       VARIABLE 6 IS CATCH ( 1,1,3 )
       GROUP 1 COMBINES VARIABLES 1 *TO* 3
       GROUP 2 COMBINES VARIABLES 1 *TO* 6
       GROUP 3 COMBINES VARIABLES 4 *TO* 6
```

```
C       -----------BLOCK 2 : STRUCTURE INPUT: NO. OF ENTITIES AND LINKS
NPROC    3 PROCESSORS
NPROD    1 PRODUCT
NSTOK    1 STOCK
NLOCN    3 LOCATIONS
NREGL    1 REGULATOR
NMARK    1 MARKET
NHARV    3 HARVESTERS
PPLINK   PROCESSORS 1 *TO* 3 SELL PRODUCT 1
MPLINK   PRODUCT 1 IS SOLD IN MARKET 1
PMLINK   PROCESSORS 1 *TO* 3 SELL TO MARKET 1
HPLINK   HARVESTERS 1 *TO* 3 SELL TO PROCESSORS 1 *TO* 3
RHLINK   REGULATOR 1 CONTROLS HARVESTERS 1 *TO* 3
HLLINK   HARVESTERS 1 *TO* 3 FISH IN LOCATIONS 1 *TO* 3
SLLINK   STOCK 1 IS FOUND IN LOCATIONS 1 *TO* 3
HSLINK   HARVESTERS 1 *TO* 3 FISH FOR STOCK 1
RPLINK   REGULATOR 1 CONTROLS PROCESSORS 1 *TO* 3

C       -----------BLOCK 3 : HARV., PROC., AND PROC.-OWNED HARV. SETS
NPSET    PROCESSOR SET 1, ENTRIES 1 *TO* 3, CONTAINS PROCESSORS 1 *TO* 3
NHSET    HARVESTER SET 1, ENTRIES 1 *TO* 3, CONTAINS HARVESTERS 1 *TO* 3
NHPSET   HARVESTER SET 1 IS OWNED BY PROCESSOR SET 1
SPONLY   ONLY WRITE OUT SEPERATE REPORTS FOR HARVESTERS AND  PROCESSORS

C       -----------BLOCK 4 : MARKETING PARAMETERS
ALPHA ( 1,1 )= .1
BETA  ( 1,1 )= .1
GAMMA ( 1,1 )= 1.0
TRANS ( 1 *TO* 3,1,1 )= .2
PDC      PRICE AXIS FOR PROD 1 IN MARK 1 = ***( 275,210,150,120,90 )
QDC      QUANT. AXIS FOR PROD 1 IN MARK 1 = ***( 18000,33000,50000,65000,83000 )
TOTMRK   TOTAL VALUE OF MARKET FOR PRODUCT 1 IN MARKET 1 = 165000
TOTMKG   TOTAL MARKETING EXPEN. FOR PROD 1 IN MARK 1 = 2080000

C       -----------BLOCK 5 : PROCESSOR STRUCTURE
MSHARE   MARK SHARE FOR PROC 1 *TO* 3,PROD 1,IN MARK 1 = ***( .245,.355,.4 )
```

```
INVNT    INVENTORY FOR PROCESSOR 1, PRODUCT 1 = 100000 UNITS
INVNT    INVENTORY FOR PROCESSOR 2, PRODUCT 1 = 150000 UNITS
INVNT    INVENTORY FOR PROCESSOR 3, PRODUCT 1 = 200000 UNITS
DIN      DOLLAR VALUE OF INVENTORY FOR PROC 1 PROD 1 = 500000
DIN      DOLLAR VALUE OF INVENTORY FOR PROC 2 PROD 1 = 750000
DIN      DOLLAR VALUE OF INVENTORY FOR PROC 3 PROD 1 = 1000000
PTOTPE   PLANT AND EQUIP. FOR PROCESSOR 1 = 17500000
PTOTPE   PLANT AND EQUIP. FOR PROCESSOR 2 = 45000000
PTOTPE   PLANT AND EQUIP. FOR PROCESSOR 3 = 60000000
PCOSTU   COST PER UNIT OF PLANT AND EQUIP. FOR PROC 1 *TO* 3 = 100
PCASH    INITIAL CASH FOR PROC. 1 *TO* 3 = ***( 100000000,140000000,150000000 )
PRTAIN   RETAINED EARNINGS FOR PROCESSOR 1 *TO* 3 = 0
PBLOAN   BANK LOANS FOR PROCESSORS 1 *TO* 3 = 0
PUCAP    CAPACITY PER $ OF PLANT FOR PROC 1 *TO* 3 STOCK 1 = .0119
DLCRAW   DIRECT LABOR COST PER UNIT OUTPUT OF RAW MAT., P 1 PROD 1 = 55
DLCRAW   DIRECT LABOR COST PER UNIT OUTPUT OF RAW MAT., P 2 PROD 1 = 50
DLCRAW   DIRECT LABOR COST PER UNIT OUTPUT OF RAW MAT., P 3 PROD 1 = 50
PDRATE   DEPRECIATION RATE FOR P. AND E. FOR PROC 1 *TO* 3 = .00415
PFXEXP   FIXED EXPENSE FOR PROCESSOR 1 *TO* 3 = ***( 500000,910000,1340000 ).
PINTRS   INTEREST RATE FOR PROCESSOR 1 *TO* 3 = .1
PCAPTL   CAPITAL FOR PROCESSORS 1 *TO* 3 = ***( 118000000,185750000,221000000 )
PLOCK    LOCATION FOR PROCESSORS 1 *TO* 3 = ( 1,1 )
PLTLIB   LONG TERM LIABILITIES FOR PROCESSORS 1 *TO* 3 = 0
POPERX   RAW MATERIAL OP. COSTS FOR PROC 1 *TO* 3, STOCK 1, = 1

C        -----------BLOCK 6 : HARVESTER STRUCTURE
HPLOCX   INITIAL LOCATION OF HARVESTERS 1 *TO* 3 = ( 1,1 )
HCASH    INITIAL CASH FOR HARV. 1 *TO* 3 = ***( 40000000,55000000,65000000 )
HTOTPE   TOTAL PLANT AND EQUIPMENT FOR HARVESTER 1 = 10000000
HTOTPE   TOTAL PLANT AND EQUIPMENT FOR HARVESTER 2 = 11000000
HTOTPE   TOTAL PLANT AND EQUIPMENT FOR HARVESTER 3 = 14430000
HCOSTU   COST PER UNIT OF PLANT AND EQUIP FOR HARV. 1 *TO* 3 = 100
HBLOAN   BANK LOANS FOR HARVESTERS 1 *TO* 3 = 0
HCAPTL   CAPITAL FOR HARVESTERS 1 *TO* 3 = ***( 40960000,66000000,79470000 )
HDRATE   DEPRECIATION RATE FOR HARVESTERS 1 *TO* 3 = 0.0033
HFXEXP   FIXED EXPENSE FOR HARV. 1 *TO* 3 = ***( 2000000,1500000,2275000 )
HINTRS   INTEREST RATE FOR HARVESTERS 1 *TO* 3 = .1
HRTAIN   RETAINED EARNINGS FOR HARVESTERS 1 *TO* 3 = 0
HLTLIB   LONG TERM LIABILITIES FOR HARVESTERS 1 *TO* 3 = 0
HVROPE   DIRECT OP. COST PER UNIT H PER DAY,H 1 *TO* 3,S 1 = 1000
HRCOST   DIRECT HARVESTING COST/UNIT HARVESTED FOR H 1 *TO* 3,S 1 = 1.0
HVRCST   DISTANCE RELATED COST PER HARV UNIT PER UNIT DIST, H 1 *TO* 3,S 1 = 1
DINH     DOLLAR VALUE OF INVENTORY FOR H 1 *TO* 3, S 1 = 0
```

```
C         -----------BLOCK 7 : STOCK INPUT
AGEMAX  MAXIMUM AGE FOR STOCK 1 = 6
NOGRID  NUMBER OF SPAWNING GROUNDS FOR STOCK 1 = 1
SPGRID  INDEX OF SPAWNING GROUNDS FOR STOCK 1 LOC 1 *TO* 3 = 1
RECR*   LENGTH OF RECRUITS FOR STOCK 1 = 6.31 CM
SURVN   NORMAL EGG SURVIVAL RATE FOR STOCK 1 = .00015
RECINT  INTERVAL FROM SPAWNING TO RECRUIT, STOCK 1 = 9 MONTHS
LINFN   ASYMPTOTIC LENGTH, STOCK 1 = 16.2
VBK     A VON BERTALANFFY PARAMETER FOR STOCK 1 = 1.1844
TZERO   FOR STOCK 1 = 0.0
VBQ     LENGTH-WEIGHT MULTIFLIER FOR STOCK 1 = .00000000673
WINF    MAX. STOCK SIZE FOR STOCK 1 IN LOC 1 = 10000000 TONS
WINF    MAX. STOCK SIZE FOR STOCK 1 IN LOC 2 = 12500000 TONS
WINF    MAX. STOCK SIZE FOR STOCK 1 IN LOC 3 = 2500000 TONS
FFISH   MAGNITUDE OF FISH NUMBERS = 1000000000
CENSUS  NO. OF FISH,S 1,AGE 1 *TO* 6,L 1 = ***( 293.9,80.1,19.8,4.9,1.1,.24 )
CENSUS  NO. OF FISH,S 1,AGE 1 *TO* 6,L 2 = ***( 367.4,100.1,24.7,6.1,1.4,.29 )
CENSUS  NO. OF FISH,S 1,AGE 1 *TO* 6,L 3 = ***( 73.5,20,4.9,1.2,.27,.06 )
FRFEM   FRACTION WHICH IS FEMALE,STOCK 1,AGE 1 *TO* 6 = 0.5
FRMATF  FEMALE MATING FRACTION,S 1,A 1 *TO* 6 = ***( .05,.45,.68,.75,.78,.85 )
FRMATM  MALE MATING FRACTION,STOCK 1,A 1 *TO* 6 = ***( .05,.45,.68,.75,.78,.85 )
FEGGS   MAGNITUDE OF EGGS PER FEMALE= 1000.0
EFEM    EGGS PER FEMALE,STOCK 1,AGE 1 *TO* 6 = ***( 1.8,6,12,15.9,18.2,19.3 )
FCOEFF  MAGNITUDE OF COEFFICIENT OF CATCHABILITY= 0.001
QCATCH  COEFFICIENT OF CATCHABILITY,STOCK 1,AGE 1 *TO* 6 = 8
NMORT   NATURAL MORT.,STOCK 1 AGE 1 *TO* 6 = ***( 1,1.2,1.2,1.4,1.4,2 )
ALNGTH  LENGTH S 1,A 1 *TO* 6,L 1 = ***( 7.24,11.24,13.46,14.68,15.36,15.72 )
ALNGTH  LENGTH S 1,A 1 *TO* 6,L 2 = ***( 7.24,11.24,13.46,14.68,15.36,15.72 )
ALNGTH  LENGTH S 1,A 1 *TO* 6,L 3 = ***( 7.24,11.24,13.46,14.68,15.36,15.72 )
WEIGHT  S 1,A 1 *TO* 4,L 1 = ***( .00000277,.00001036,.00001780,.00002309 )
WEIGHT  S 1,A 1 *TO* 4,L 2 = ***( .00000277,.00001036,.00001780,.00002309 )
WEIGHT  S 1,A 1 *TO* 4,L 3 = ***( .00000277,.00001036,.00001780,.00002309 )
WEIGHT  S 1,A 5 *TO* 6,L 1 = ***( .00002645,.00002847 )
WEIGHT  S 1,A 5 *TO* 6,L 2 = ***( .00002645,.00002847 )
WEIGHT  S 1,A 5 *TO* 6,L 3 = ***( .00002645,.00002847 )
WF      SUMMATION MATRIX FOR STOCK 1,A 1 *TO* 6 = 1
WF      SUMMATION MATRIX FOR STOCK 2,A 1 *TO* 6 = 1
WF      SUMMATION MATRIX FOR STOCK 3,A 1 *TO* 6 = 1
ICYCL   NO. OF CYCLES,S 1 M 1 *TO* 12 = ***( 4,1,4,4,4,1,1,1,4,4,4,4 )
FRSPAWN S 1,M 1 *TO* 12 = ***( 0,.01,.01,.09,.31,.01,.09,.4,.31,.16,.15,.04 )
FRECR   MAGNITUDE OF RECRUIT NOS. = 1000000000
RECRT   R/MO S 1,S 1,M 1 *TO* 12 = ***( 300,400,150,30,0 )
```

```
AVFR     FR.AV S 1,L 1,M 1 *TO* 9 = ***( 1.52,1.18,.92,.96,.83,.85,.75,.66,.61 )
AVFR     FR.AV S 1,L 1,M 10 *TO* 12 = ***( .93,1.45,1.86 )
AVFR     FR.AV S 1,L 2 M 1 *TO* 9 = ***( 1.79,1.21,.88,.79,.89,.76,.73,.71,.86 )
AVFR     FR.AV S 1,L 2 M 10 *TO* 12 = ***( .93,1.37,1.72 )
AVFR     FR.AV S 1,L 3 M 1 *TO* 9 = ***( 1.32,1.34,1.11,.95,.94,.83,.57,.71,.90 )
AVFR     FR.AV S 1,L 3 M 10 *TO* 12 = ***( .93,1.26,1.17 )
SPFEMN   S 1 G 1 MO 1 *TO* 12 = ***( 0,20,20,175,20,20,175,780,640,312,93,78 )
MARK     MARKING FR. S 1,M 1 *TO* 12 = ***( .1,.2,.2,.5,.1,0,.8,0 )
SMULT    EGG SURV MUL. S 1 K 1 *TO* 10 = ***( 10,3,1,.3,.1,.03,.01,.003,.001,0 )
LMULT    LENGTH INCRE. MUL S 1,K 1 *TO* 10 = 1
FRMA*    FEMALE SP L SCH. S 1,K 1 *TO* 10 = ***( 8,8.5,9,10,11,12,13,14,15,16 )
FRMA*    MALE SPWN L SCH. S 1,K 1 *TO* 10 = ***( 8,8.5,9,10,11,12,13,14,15,16 )

C        -----------BLOCK 8 : PROCESSOR DECISIONS
OPLEVI   OPERATING LEVEL FOR PROC 1 *TO* 3,STOCK 1 = 210000
PADDPE   ADDITIONAL PLANT AND EQUIP. FOR PROC. 1 *TO* 3 = 0
PRANDD   RESEARCH AND DEVELOPMENT FOR PROC. 1 *TO* 3 = 0
MRKTNG   MARKETING FOR PROC 1 *TO* 3,PROD 1,MARK 1 = ***( 460000,760000,670000 )
PURPRI   PURCHASE PRICE BY PROC 1 *TO* 3,OF STOCK 1 = $ 10
PRICES   SELLING PRICE FOR PROC 1 *TO* 3,P 1,M 1 = 130
PRLOAN   REPAYMENT OF LOANS BY PROCESSORS 1 *TO* 3 = 0

C        -----------BLOCK 9 : HARVESTER DECISIONS
HADDPE   ADDITIONAL PLANT AND EQUIP. FOR HARVESTERS 1 *TO* 3 = 0
HRANDD   RESEARCH AND DEVELOPMENT FOR HARVESTERS 1 *TO* 3 = 0
HCAPAC   CAPACITY PER BOAT PER TRIP PER H 1 *TO* 3,S 1 = 500 TONS
HRLOAN   REPAYMENT OF LOANS BY HARVESTER 1 *TO* 3 = 0
HDELI    NO. OF DAYS PER MONTH HARV PLANNED BY H 1 *TO* 3,S 1 = 20
DYSTRP   TRIP TIME IN DAYS FOR H 1 *TO* 3,S 1 = 1
EFFDI   NO. OF HARVESTING UNITS USED BY HARV 1 *TO* 3,STOCK 1 = 50
FPWDI   HARVESTING EFFICIENCY/UNIT FOR HARV 1 *TO* 3,STOCK 1 = 1.0
HLOCX    HARVESTING LOCATION FOR HARVESTER 1 = ( 11,11 )
HLOCX    HARVESTING LOCATION FOR HARVESTER 2 = ( 1,1 )
HLOCX    HARVESTING LOCATION FOR HARVESTER 3 = ( 1,1 )

C        -----------BLOCK 10 : REGULATOR DECISIONS
CQUOTA   CATCH QUOTA FOR STOCK 1 IN LOC 1 *TO* 3 = 1250000 TONS
XI       QUOTA MULTIPLIERS, FOR S 1, LOC. 1 *TO* 3 = 1.0
HLIN    LICENSE FEES FOR UNITS HARVESTING STOCK 1 IN LOC 1 *TO* 3 = 1000
C        ----SHUT OFF CATCH QUOTA
NAUT     1 *TO* 2 = 1
OPLL     MAXIMUM OPERATING LEVEL FOR PROC. 1 *TO* 3,STOCK 1 = 300000
SLIMIT   SIZE LIMIT FOR STOCK 1, LOC 1 *TO* 3 = AGE 1
RDELT    FISHING SEASON FOR STOCK 1 LOC. 1 *TO* 3 = 20 DAYS
CMQ      MONTHLY QUOTA FOR STOCK 1 LOC. 1 *TO* 3 = 900000 TONS
EFFL     MAX. NO. OF HARVESTING UNITS FOR S 1 IN LOC 1 *TO* 3 = 50
```

## APPENDIX V: Sample Output

The following sample output was generated by the sample input given in Appendix IV.

```
----------

///YEAR 1970, MONTH  1/// REPORTS FOR STOCK SECTOR:

POPULATION OF          AGE 1      AGE 2      AGE 3      AGE 4      AGE 5      AGE 6      AGE 7      AGE
STOCK 1 IN LOC 1     3.415E+11  8.364E+10  2.236E+10  5.469E+09  1.271E+09  2.664E+08
STOCK 1 IN LOC 2     4.257E+11  1.105E+11  2.858E+10  6.918E+09  1.532E+09  3.267E+08
STOCK 1 IN LOC 3     8.595E+10  2.229E+10  5.535E+09  1.358E+09  3.139E+08  6.552E+07

WEIGHT OF FISH OF      AGE 1      AGE 2      AGE 3      AGE 4      AGE 5      AGE 6      AGE 7      AGE
STOCK 1 IN LOC 1     3.498E-06  8.334E-06  1.639E-05  2.216E-05  2.581E-05  2.801E-05
STOCK 1 IN LOC 2     3.498E-06  8.334E-06  1.638E-05  2.216E-05  2.583E-05  2.793E-05
STOCK 1 IN LOC 3     3.498E-06  8.331E-06  1.637E-05  2.215E-05  2.582E-05  2.802E-05

LENGTH OF FISH OF      AGE 1      AGE 2      AGE 3      AGE 4      AGE 5      AGE 6      AGE 7      AGE
STOCK 1 IN LOC 1       7.82       10.7       13.1       14.5       15.2       15.7
STOCK 1 IN LOC 2       7.82       10.7       13.1       14.5       15.2       15.7
STOCK 1 IN LOC 3       7.82       10.7       13.1       14.5       15.2       15.7

CATCH OF               AGE 1      AGE 2      AGE 3      AGE 4      AGE 5      AGE 6      AGE 7      AGE
STOCK 1 IN LOC 1     3.508E+04  2.731E+04  1.340E+04  4.536E+03  1.236E+03    293
STOCK 1 IN LOC 2     4.305E+04  3.681E+04  1.304E+04  6.095E+03  1.686E+03    385
STOCK 1 IN LOC 3     7.377E+03  5.574E+03  2.716E+03    910        248        58.9
----------

----------
///YEAR 1970, MONTH  1/// REPORTS FOR MARKET SECTOR:

TOTAL MARKET           PROD 1     PROD 2     PROD 3     PROD 4     PROD 5     PROD 6     PROD 7     PROD
MARKET  1            8.548E+06

TOTAL MARKETING        PROD 1     PROD 2     PROD 3     PROD 4     PROD 5     PROD 6     PROD 7     PROD
MARKET  1            2.096E+06

DEMAND CURVE FOR PRODUCT  1 IN MARKET  1
QUANTITY AXIS        1.697E+04  3.500E+04  5.303E+04  7.000E+04  8.803E+04
PRICE AXIS             275        210        150        120        90.0
----------
```

```
----------
///YEAR 1970, MONTH  1/// REPORTS FOR HARVESTER SECTOR:

UNITS HARVESTED         STOCK 1    STOCK 2    STOCK 3    STOCK 4    STOCK 5    STOCK 6    STOCK 7    STOCK
BY HARVESTER  1        6.235E+04
BY HARVESTER  2        1.117E+05
BY HARVESTER  3        1.683E+04

UNITS SOLD BY           STOCK 1    STOCK 2    STOCK 3    STOCK 4    STOCK 5    STOCK 6    STOCK 7    STOCK
HARV 1 TO PROC 1       6.235E+04
HARV 2 TO PROC 2       1.117E+05
HARV 3 TO PROC 3       1.683E+04

UNITS UNSOLD            STOCK 1    STOCK 2    STOCK 3    STOCK 4    STOCK 5    STOCK 6    STOCK 7    STOCK
BY HARVESTER  1        0.
BY HARVESTER  2        0.
BY HARVESTER  3        0.

NO. OF HARV. UNITS      FLEET 1    FLEET 2    FLEET 3    FLEET 4    FLEET 5    FLEET 6    FLEET 7    FLEET
FOR HARVESTERS  1        50.0
FOR HARVESTERS  2        50.0
FOR HARVESTERS  3        50.0

DISTANCE TRAVELED       FLEET 1    FLEET 2    FLEET 3    FLEET 4    FLEET 5    FLEET 6    FLEET 7    FLEET
BY HARVESTER  1          256
BY HARVESTER  2        0.
BY HARVESTER  3        0.

HARVESTING TIME
(IN DAYS)               FLEET 1    FLEET 2    FLEET 3    FLEET 4    FLEET 5    FLEET 6    FLEET 7    FLEET
FOR HARVESTERS  1        20.0
FOR HARVESTERS  2        20.0
FOR HARVESTERS  3        20.0
----------


      CARD INPUT FOR JUNE( MONTH 6 ) 1970
```

```
 ----------
 ///YEAR 1970, MONTH  1/// REPORTS FOR PROCESSOR SECTOR!

 RAW MAT. PURCHASE       STOCK 1     STOCK 2     STOCK 3     STOCK 4     STOCK 5     STOCK 6     STOCK 7     STOC
 BY PROCESSOR  1        8.235E+04
 BY PROCESSOR  2        1.117E+05
 BY PROCESSOR  3        1.338E+04

 FINISHED GOODS ADDED     PROD 1      PROD 2      PROD 3      PROD 4      PROD 5      PROD 6      PROD 7      PRO
 TO PROCESSOR  1        1.657E+04
 TO PROCESSOR  2        2.233E+04
 TO PROCESSOR  3        3.377E+03

 NO. OF INVENTORY UNITS
 (AFTER SALES)            PROD 1      PROD 2      PROD 3      PROD 4      PROD 5      PROD 6      PROD 7      PRO
 FOR PROCESSOR  1       1.027E+05
 FOR PROCESSOR  2       1.434E+05
 FOR PROCESSOR  3       1.754E+05

 UNITS SOLD BY            PROD 1      PROD 2      PROD 3      PROD 4      PROD 5      PROD 6      PROD 7      PRO
 PROC 1 IN MARK 1       1.338E+04
 PROC 2 IN MARK 1       2.394E+04
 PROC 3 IN MARK 1       2.794E+04

 UNIT SALES PRICE BY      PROD 1      PROD 2      PROD 3      PROD 4      PROD 5      PROD 6      PROD 7      PRO
 PROC 1 IN MARK 1         130
 PROC 2 IN MARK 1         130
 PROC 3 IN MARK 1         130

 MARKET SHARE FOR         PROD 1      PROD 2      PROD 3      PROD 4      PROD 5      PROD 6      PROD 7      PRO
 PROC 1 IN MARK 1        .211
 PROC 2 IN MARK 1        .364
 PROC 3 IN MARK 1        .425

 MARKETING EXPENDITURES
 FOR                      PROD 1      PROD 2      PROD 3      PROD 4      PROD 5      PROD 6      PROD 7      PRO
 PROC 1 IN MARK 1       4.600E+05
 PROC 2 IN MARK 1       7.000E+05
 PROC 3 IN MARK 1       6.700E+05
 ----------
 ----------
, ///YEAR 1970, MONTH  2/// REPORTS FOR STOCK SECTOR:

 POPULATION OF             AGE 1       AGE 2       AGE 3       AGE 4       AGE 5       AGE 6       AGE 7       AGE
 STOCK 1 IN LOC 1        3.894E+11   1.223E+11   3.155E+10   7.722E+09   1.834E+09   3.855E+08
 STOCK 1 IN LOC 2        4.844E+11   1.529E+11   3.927E+10   9.595E+09   2.293E+09   4.795E+08
 STOCK 1 IN LOC 3        9.719E+10   3.078E+10   7.679E+09   1.913E+09   4.517E+08   9.543E+07

 WEIGHT OF FISH OF         AGE 1       AGE 2       AGE 3       AGE 4       AGE 5       AGE 6       AGE 7       AGE
 STOCK 1 IN LOC 1        3.552E-06   7.118E-06   1.356E-05   2.005E-05   2.449E-05   2.713E-05
 STOCK 1 IN LOC 2        3.631E-06   7.118E-06   1.355E-05   2.005E-05   2.451E-05   2.717E-05
 STOCK 1 IN LOC 3        3.535E-06   7.116E-06   1.354E-05   2.003E-05   2.449E-05   2.713E-05

 LENGTH OF FISH OF         AGE 1       AGE 2       AGE 3       AGE 4       AGE 5       AGE 6       AGE 7       AGE
 STOCK 1 IN LOC 1         8.10        9.92        12.3        14.0        15.0        15.5
 STOCK 1 IN LOC 2         8.10        9.92        12.3        14.0        15.0        15.5
 STOCK 1 IN LOC 3         8.10        9.92        12.3        14.0        15.0        15.5

 CATCH OF                  AGE 1       AGE 2       AGE 3       AGE 4       AGE 5       AGE 6       AGE 7       AGE
 STOCK 1 IN LOC 1        2.949E+04   2.089E+04   1.093E+04   4.104E+03   1.212E+03    293
 STOCK 1 IN LOC 2        3.771E+04   2.669E+04   1.395E+04   5.230E+03   1.555E+03    374
 STOCK 1 IN LOC 3        6.403E+03   5.955E+03   3.100E+03   1.155E+03    339         82.5
 ----------
C       -----------BLOCK 1 : INPUT-OUTPUT CONTROLS
C       THIS GROUP IS FRO DEMONSTRATION PURPOSES ONLY.  NO PARAMETERS WILL
C       BE CHANGED.

C       -----------BLOCK 2 : STRUCTURE INPUT: NO. OF ENTITIES AND LINKS

C       -----------BLOCK 3 : HARV., PROC., AND PROC.-OWNED HARV. SETS

C       -----------BLOCK 4 : MARKETING PARAMETERS

C       -----------BLOCK 5 : PROCESSOR STRUCTURE

C       -----------BLOCK 6 : HARVESTER STRUCTURE

C       -----------BLOCK 7 : STOCK INPUT

C       -----------BLOCK 8 : PROCESSOR DECISIONS

C       -----------BLOCK 9 : HARVESTER DECISIONS

C       -----------BLOCK 10 : REGULATOR DECISIONS
```

```
///MONTHLY REPORT,  YEAR 1970, MONTH  7///           PROCESSOR SET   1
                                                            .
          .. INCOME STATEMENT ..                            .         .. STATEMENT OF CASH FLOW ..
 TOTAL  SALES                                $ 8.5485E+06   .   TOTAL SALES                            $ 8.5485E+.
 LESS - DIRECT COST OF SALES  $ 3.0202E+06                  .   NEW LOANS                              $ 0.
        MARKETING EXPENDITURES$ 2.0900E+06                  .      TOTAL CASH PROVIDED
        ADMINIS, AND OVERHEAD $ 2.4500E+06                  .   LESS - DIRECT LABOR COST               $ 1.5465E+.
        LICENSES AND TAXES    $ 0.                          .          COST OF RAW MATERIALS           $ 1.4853E+
        DEPRECIATION          $ 5.0837E+05                  .          OPERATING COSTS                 $ 1.4563E+
        RESEARCH AND DEVEL.   $ 0.                          .          MARKETING EXPENDITURES          $ 2.0900E+
        INTEREST PAID         $ 0.                          .          ADMINIS. AND OVERHEAD           $ 2.4500E+
        TOTAL CURRENT EXPENSES                $ 8.0686E+06  .          LICENSES AND TAXES              $ 0.
                                                            .          RESEARCH AND DEVEL.             $ 0.
                                                            .          INTEREST PAID                   $ 0.
                                                            .          ADDL. PLANT AND EQUIP.          $ 0.
                                                            .          LOAN REPAYMENT                  $ 0.
                                                            .             TOTAL CASH OUTLAY
 NET INCOME                                   $ 4.7990E+05  .   NET CASH FLOW
                                                            .
..............................................................................................................

                                       .. BALANCE SHEET ..
              ASSETS                                                         LIABILITIES

 CURRENT ASSETS --                                        CURRENT LIABILITIES --
    CASH                  $ 4.0279E+08                       LOANS OUTSTANDING     $ 0.
    INVENTORY             $ 1.4267E+07                    LONG TERM LIABILITIES --
       TOTAL CURRENT ASSETS           $ 4.1705E+08           LOANS OUTSTANDING     $ 0.
                                                             TOTAL LIABILITIES                       $ 0.

 FIXED ASSETS --                                                     NET WORTH
    PLANT AND EQUIPMENT $ 1.2250E+08                      CAPITAL                  $ 5.2475E+08
    LESS DEPRECIATION   $ 3.5586E+06                      RETAINED EARNINGS        $ 1.1245E+07
       TOTAL FIXED ASSETS             $ 1.1394E+08           TOTAL NET WORTH                         $ 5.3099E+08
 TOTAL ASSETS                         $ 5.3099E+08        TOTAL LIABILITIES AND NET WORTH            $ 5.3099E+08
..............................................................................................................
```

```
///MONTHLY REPORT,  YEAR 1970, MONTH  7///          HARVESTER SET   1
                                                           .
          .. INCOME STATEMENT ..                           .               .. STATEMENT OF CASH FLOW ..
TOTAL SALES                                 $ 1.4863E+06   .   TOTAL SALES                             $ 1.4863E+
LESS - DIRECT COST OF SALES  $ 3.1769E+06                  .   NEW LOANS                               $ 0.
       MARKETING EXPENDITURES$ 0.                          .      TOTAL CASH PROVIDED
       ADMINIS. AND OVERHEAD $ 5.7750E+06                  .   LESS - DIRECT LABOR COST                $ 0.
       LICENSES AND TAXES    $ 1.5000E+05                  .          COST OF RAW MATERIALS            $ 0.
       DEPRECIATION          $ 2.9407E+05                  .          OPERATING COSTS                  $ 3.1769E+
       RESEARCH AND DEVEL.   $ 0.                          .          MARKETING EXPENDITURES           $ 0.
       INTEREST PAID         $ 0.                          .          ADMINIS. AND OVERHEAD            $ 5.7750E+
       TOTAL CURRENT EXPENSES                $ 9.3960E+06  .          LICENSES AND TAXES               $ 1.5000E+
                                                           .          RESEARCH AND DEVEL.              $ 0.
                                                           .          INTEREST PAID                    $ 0.
                                                           .          ADDL. PLANT AND EQUIP.           $ 0.
                                                           .          LOAN REPAYMENT                   $ 0.
                                                           .             TOTAL CASH OUTLAY
NET INCOME                                  $-7.9096E+06   .   NET CASH FLOW
                                                           .
.........................................................................................................

                               .. BALANCE SHEET ..
             ASSETS                                                        LIABILITIES

CURRENT ASSETS --                                        CURRENT LIABILITIES --
   CASH                   $ 1.0795E+08                      LOANS OUTSTANDING    $ 0.
   INVENTORY              $ 2.2352E-08                   LONG TERM LIABILITIES --
      TOTAL CURRENT ASSETS              $ 1.0795E+08        LOANS OUTSTANDING    $ 0.
                                                            TOTAL LIABILITIES                  $ 0.

FIXED ASSETS --                                                         NET WORTH
   PLANT AND EQUIPMENT $ 3.5430E+07                      CAPITAL                 $ 1.9543E+08
   LESS DEPRECIATION   $ 2.0585E+06                      RETAINED EARNINGS       $-5.4104E+07
      TOTAL FIXED ASSETS                $ 3.3372E+07        TOTAL NET WORTH                    $ 1.4133E+0
TOTAL ASSETS                            $ 1.4133E+08     TOTAL LIABILITIES AND NET WORTH       $ 1.4133E+0
.........................................................................................................
```

```
----------

///YEAR 1978, MONTH  7/// REPORTS FOR STOCK SECTOR:

POPULATION OF             AGE 1       AGE 2       AGE 3       AGE 4       AGE 5       AGE 6        AGE 7       AGE
STOCK 1 IN LOC 1        2.735E+10   1.142E+11   1.332E+11   5.026E+10   1.731E+10   4.121E+09
STOCK 1 IN LOC 2        3.431E+10   1.431E+11   1.669E+11   7.337E+10   2.166E+10   5.152E+09
STOCK 1 IN LOC 3        6.338E+09   2.651E+10   3.328E+10   1.534E+10   4.319E+09   1.023E+09

WEIGHT OF FISH OF         AGE 1       AGE 2       AGE 3       AGE 4       AGE 5       AGE 6        AGE 7       AGE
STOCK 1 IN LOC 1        7.633E-06   9.680E-06   1.075E-05   1.274E-05   1.627E-05   2.079E-05
STOCK 1 IN LOC 2        7.637E-06   9.878E-06   1.075E-05   1.274E-05   1.628E-05   2.079E-05
STOCK 1 IN LOC 3        7.627E-06   9.653E-06   1.075E-05   1.274E-05   1.625E-05   2.079E-05

LENGTH OF FISH OF         AGE 1       AGE 2       AGE 3       AGE 4       AGE 5       AGE 6        AGE 7       AGE
STOCK 1 IN LOC 1         10.1        11.1        11.4        12.0        13.1        14.2
STOCK 1 IN LOC 2         10.2        11.1        11.4        12.0        13.1        14.2
STOCK 1 IN LOC 3         10.1        11.1        11.4        12.0        13.1        14.2

CATCH OF                  AGE 1       AGE 2       AGE 3       AGE 4       AGE 5       AGE 6        AGE 7       AGE
STOCK 1 IN LOC 1        2.934E+03   1.657E+04   2.171E+04   1.204E+04   4.571E+03   1.470E+03
STOCK 1 IN LOC 2        3.690E+03   2.193E+04   2.829E+04   1.567E+04   5.946E+03   1.912E+03
STOCK 1 IN LOC 3          565       3.196E+03   4.114E+03   2.261E+03     354          273
----------
```

```
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 THE FOLLOWING FINANCIAL STATEMENT HAS BEEN ACCUMULATED OVER THE PAST   1 YEARS.
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 ///SUMMARY REPORT.  YEAR 1976, MONTH 12///          PROCESSOR SET   1
                                                                  .
              .. INCOME STATEMENT ..                              .           .. STATEMENT OF CASH FLOW ..
 TOTAL SALES                                      $ 3.8434E+07    .  TOTAL SALES                                $ 3.843+E+
 LESS - DIRECT COST OF SALES   $ 3.2939E+07                       .  NEW LOANS                                  $ 0.
        MARKETING EXPENDITURES $ 2.5060E+07                       .     TOTAL CASH PROVIDED
        ADMINIS. AND OVERHEAD  $ 2.9400E+07                       .  LESS - DIRECT LABOR COST                   $ 2.4378E+
        LICENSES AND TAXES     $ 0.                               .         COST OF RAW MATERIALS               $ 2.3371E+
        DEPRECIATION           $ 6.1005E+06                       .         OPERATING COSTS                     $ 2.3373E+
        RESEARCH AND DEVEL.    $ 0.                               .         MARKETING EXPENDITURES              $ 2.5060E+
        INTEREST PAID          $ 0.                               .         ADMINIS. AND OVERHEAD               $ 2.9400E+
        TOTAL CURRENT EXPENSES                    $ 9.3520E+07    .         LICENSES AND TAXES                  $ 0.
                                                                  .         RESEARCH AND DEVEL.                 $ 0.
                                                                  .         INTEREST PAID                       $ 0.
                                                                  .         ADDL. PLANT AND EQUIP.              $ 0.
                                                                  .         LOAN REPAYMENT                      $ 0.
                                                                  .            TOTAL CASH OUTLAY
 NET INCOME                                       $-5.0859E+06    .  NET CASH FLOW
                                                                  .
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

                                        .. BALANCE SHEET ..
                  ASSETS                                                        LIABILITIES

 CURRENT ASSETS --                                             CURRENT LIABILITIES --
    CASH                    $ 3.8394E+08                          LOANS OUTSTANDING     $ 0.
    INVENTORY               $ 1.9326E+07                       LONG TERM LIABILITIES --
       TOTAL CURRENT ASSETS                $ 4.0326E+08           LOANS OUTSTANDING     $ 0.
                                                                  TOTAL LIABILITIES                      $ 0.

 FIXED ASSETS --                                                              NET WORTH
    PLANT AND EQUIPMENT $ 1.2250E+08                           CAPITAL                  $ 5.2475E+08
    LESS DEPRECIATION   $ 6.1006E+06                           RETAINED EARNINGS        $-5.0859E+06
       TOTAL FIXED ASSETS                  $ 1.1640E+08           TOTAL NET WORTH                        $ 5.1966E+
 TOTAL ASSETS                              $ 5.1966E+08        TOTAL LIABILITIES AND NET WORTH           $ 5.1966E+
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 THE FOLLOWING FINANCIAL STATEMENT HAS BEEN ACCUMULATED OVER THE PAST   1 YEARS.
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 ///SUMMARY REPORT,  YEAR 1973, MONTH 12///            HARVESTER SET   1
                                                                  .
             .. INCOME STATEMENT ..                               .
 TOTAL SALES                                                      .        .. STATEMENT OF CASH FLOW ..
                                                $ 2.3370E+07      .    TOTAL SALES                             $ 2.3370E+
 LESS - DIRECT COST OF SALES       $ 3.8675E+07                   .    NEW LOANS                               $ 0.
        MARKETING EXPENDITURES$ 0.                                 .        TOTAL CASH PROVIDED
        ADMINIS. AND OVERHEAD  $ 6.9300E+07                       .    LESS - DIRECT LABOR COST                $ 0.
        LICENSES AND TAXES     $ 1.9000E+06                       .           COST OF RAW MATERIALS            $ 0.
        DEPRECIATION           $ 3.5288E+06                       .           OPERATING COSTS                  $ 3.8675E+
        RESEARCH AND DEVEL.    $ 0.                               .           MARKETING EXPENDITURES           $ 0.
        INTEREST PAID          $ 0.                               .           ADMINIS. AND OVERHEAD            $ 6.9300E+
        TOTAL CURRENT EXPENSES                  $ 1.1331E+08      .           LICENSES AND TAXES               $ 1.9000E+
                                                                  .           RESEARCH AND DEVEL.              $ 0.
                                                                  .           INTEREST PAID                    $ 0.
                                                                  .           ADDL. PLANT AND EQUIP.           $ 0.
                                                                  .           LOAN REPAYMENT                   $ 0.
                                                                  .              TOTAL CASH OUTLAY
 NET INCOME                                     $-8.9935E+07      .    NET CASH FLOW
                                                                  .
 ...........................................................................................................

                                       .. BALANCE SHEET ..
                  ASSETS
                                                                             LIABILITIES

 CURRENT ASSETS --
    CASH                        $ 7.3594E+07                        CURRENT LIABILITIES --
    INVENTORY                   $ 2.2352E-08                           LOANS OUTSTANDING    $ 0.
       TOTAL CURRENT ASSETS                  $ 7.3594E+07           LONG TERM LIABILITIES --
                                                                       LOANS OUTSTANDING    $ 0.
                                                                       TOTAL LIABILITIES                 $ 0.
 FIXED ASSETS --
    PLANT AND EQUIPMENT $ 3.5430E+07                                            NET WORTH
    LESS DEPRECIATION    $ 3.5288E+06                       CAPITAL                         $ 1.9543E+08
       TOTAL FIXED ASSETS                    $ 3.1901E+07           RETAINED EARNINGS               $-8.9935E+07
 TOTAL ASSETS                                $ 1.0550E+08              TOTAL NET WORTH                   $ 1.0550E+08
                                                                    TOTAL LIABILITIES AND NET WORTH     $ 1.0550E+08
 ...........................................................................................................
```

```
///YEAR: 1973///  ANNUAL SUMMARY TABLE FOR PHYSICAL REPORTS.

GROUP            JAN       FEB       MAR       APR       MAY       JUN       JUL       AUG       SEP       OCT

GROUP  1   8.53E+11  9.70E+11  8.35E+11  4.04E+11  3.28E+11  2.97E+11  6.90E+10  8.19E+10  2.65E+11  2.61E+11  2.54E
GROUP  2   8.53E+11  9.70E+11  8.35E+11  4.04E+11  3.28E+11  2.97E+11  6.90E+10  8.19E+10  2.65E+11  2.61E+11  2.54E
GROUP  3   9.21E+04  7.56E+04  6.87E+04  3.93E+04  4.81E+04  3.79E+04  7.44E+03  7.45E+03  1.51E+04  2.34E+04  4.10E

COL SUM    1.71E+12  1.94E+12  1.67E+12  8.08E+11  6.56E+11  5.93E+11  1.38E+11  1.64E+11  5.30E+11  5.21E+11  5.08E
COL AVE    5.69E+11  6.47E+11  5.57E+11  2.69E+11  2.19E+11  1.98E+11  4.60E+10  5.46E+10  1.77E+11  1.74E+11  1.69E
```

## Appendix 2 - Listing of initial input values

```
C     SOUTHERN CALIFORNIA ANCHOVY MODEL                                       DATA1    2
C                                                                             DATA1    3
C      SINGLE STOCK MODEL                                                     DATA1    4
C                                                                             DATA1    5
C       -----------BLOCK 0 - DEFAULT AND CARD OUTPUT CONTROLS                 DATA1    6
        SUPPRESS DEFAULT INPUT                                                DATA1    7
        SUPPRESS CARD INPUT                                                   DATA1    8
                                                                              DATA1    9
C       -----------BLOCK 1 - INPUT-OUTPUT CONTROLS                            DATA1   10
        THE SIMULATION RUNS FROM YEAR 1 TO 20                                 DATA1   11
       FINANCIAL REPORTS ARE ACCUMULATED ONLY ONE YEAR AT A TIME              DATA1   12
       CUMULATIVE FINANCIAL REPORTS ARE PRINTED YEAR 20                       DATA1   13
       PHYSICAL REPORTS ARE PRINTED YEARS 20                                  DATA1   14
       STOCK REPORTS PRINTED MONTHS 12                                        DATA1   15
       HARVESTER REPORTS PRINTED MONTHS 12                                    DATA1   16
       PROCESSOR REPORTS PRINTED MONTHS 12                                    DATA1   17
                                                                              DATA1   18
C       -----------BLOCK 2 - STRUCTURE INPUT                                  DATA1   19
NPROD  2 PRODUCTS                                                             DATA1   20
NMARK  2 MARKETS                                                              DATA1   21
NPROC  1 PROCESSORS                                                           DATA1   22
NREGL  1 REGULATOR                                                            DATA1   23
NHARV  1 HARVESTER                                                            DATA1   24
NLOCN   1 LOCATION                                                            DATA1   25
NSTOK  1 STOCK                                                                DATA1   26
PPLINK  PROCESSOR 1 SELLS PRODUCTS 1 *TO* 2                                   DATA1   27
MPLINK  PRODUCT 1 *TO* 2 SOLD ON MARKET 1 *TO* 2                              DATA1   28
PMLINK  PROCESSOR 1 SELLS TO MARKET 1 *TO* 2                                  DATA1   29
HPLINK  HARVESTER 1 SELLS TO PROCESSOR 1                                      DATA1   30
RHLINK  REGULATOR 1 CONTROLS HARVESTER 1                                      DATA1   31
HLLINK  HARVESTER 1 OPERATES IN LOCATION 1                                    DATA1   32
SLLINK  STOCK 1 FOUND IN LOCATION 1                                           DATA1   33
HSLINK  HARVESTER 1 FISHES ON STOCK 1                                         DATA1   34
RPLINK  REGULATOR 1 CONTROLS PROCESSOR 1                                      DATA1   35
                                                                              DATA1   36
```

```
C          -----------BLOCK 3 - HARV,PROC,PROC-HARV SETS                                      DATA1    37
NPSET    PROCESSOR SET 1, ENTRY 1, CONTAINS PROCESSOR 1                                       DATA1    38
NHSET    HARVESTER SET 1, ENTRY 1, CONTAINS HARVESTER 1                                       DATA1    39
NHPSET   HARVESTER SET 1 OWNED BY PROCESSOR SET 1                                             DATA1    40
SPCOMB   PRINT OUT SEPARATE AND COMBINED REPORTS                                              DATA1    41
                                                                                              DATA1    42
C          -----------BLOCK 4 - MARKETING PARAMETERS                                          DATA1    43
ALPHA( 1 *TO* 2,1 *TO* 2 )= 0.1                                                               DATA1    44
BETA ( 1 *TO* 2,1 *TO* 2 )= 0.1                                                               DATA1    45
GAMMA( 1 *TO* 2,1 *TO* 2 )= 0.1                                                               DATA1    46
TRANS( 1,1,1 ) = 0.20  NA INTO MT OF MEAL                                                     DATA1    47
TRANS( 1,1,2 )= 0.01   NA INTO M TONS OF OIL                                                  DATA1    48
PDC       PRICE AXIS FOR PROD 1 IN MARK 1 = 250                                               DATA1    49
QDC      CUANTITY AXIS FOR PRODUCT 1 IN MARKET 1 = 10000000                                   DATA1    50
PDC       PRICE AXIS FOR PROD 2 IN MARK 2 = 110                                               DATA1    51
QDC      CUANTITY AXIS FOR PRODUCT 2 IN MARKET 2 = 10000000                                   DATA1    52
                                                                                              DATA1    53
C          -----------BLOCK 5 - PROCESSOR STRUCTURE                                           DATA1    54
MSHARE MARK SHARE FOR PROC 1 , PROD 1 *TO* 2, IN MARK 1 *TO* 2 = 1.00                         DATA1    55
PCOSTU  COST PER UNIT PLANT-EQUIPMENT PROC 1 = 300000.                                        DATA1    56
PTOTPE  PLANT AND EQUIPMENT FOR PROC 1 = 1200000                                              DATA1    57
PUCAP  CAPACITY PER $ OF PLANT FOR PROC 1 STOCK 1 = 0.02548                                   DATA1    58
PFXCXP  FIXED COST PROC 1 = 50000                                                             DATA1    59
POPIRX  RAW MATERIAL OPERATING COST PROC 1 STOCK 1 = 5.50 / MT                                DATA1    60
PLOCX   LOCATION OF PROCESSOR 1 = ( 0,0 )                                                     DATA1    61
                                                                                              DATA1    62
C          -----------BLOCK 6 - HARVESTER STRUCTURE                                           DATA1    63
HCOSTU COST PER UNIT OF PLANT AND EQUIP FOR HARVESTER 1 = 279300                              DATA1    64
HTOTPE  TOTALPLANT-EQUIPMENT FOR HARVESTER 1 = 21226800                                       DATA1    65
HFXCXP  FIXEDCOST HARVESTER 1 = 12150 PER VESSEL PER MONTH                                    DATA1    66
HVPOPE DIRECT OP COSTS PER UNIT PER DAY , HARVE 1, STOCK 1 = 77.75                            DATA1    67
HRCOST DIRECT HARVESTING COST PER UNIT HARVESTED FOR HARVESTER 1 , STOCK 1 = 2.2DATA1    68
HPLOCX  INITIAL LOCATION HARVESTER 1 = ( 0,0 )                                                DATA1    69
                                                                                              DATA1    70
```

```
C        ----------BLOCK 7 - STOCK INPUT FOR SINGLE STOCK MODEL                       DATA1   71
C                                                                                     DATA1   72
C      STOCK COMPOSITION                                                              DATA1   73
C                                                                                     DATA1   74
AGEMAX MAXIMUM AGE FOR STOCK 1 = 7                                                    DATA1   75
FRFEM  FRACTION WHICH IS FEMALE,STOCK 1 , AGE 1 *TO* 7 = .56                          DATA1   76
RINF   MAX STOCK SIZE FOR STOCK 1 IN LOCATION 1 =  4550000                            DATA1   77
FFISH  MAGNITUDE OF FISH NUMBERS = 1000000000.0                                       DATA1   78
CENSUS NO OF FISH , S 1, AGE 1 *TO* 5,L 1 = ***( 0.0,228.85,76.18,25.36,8.44 )        DATA1   79
CENSUS NO OF FISH , S 1, AGE 6 *TO* 7,L 1 = ***( 2.81,0.94 )                          DATA1   80
C                                                                                     DATA1   81
C      SPAWNING                                                                       DATA1   82
C                                                                                     DATA1   83
FRMATF FEMALE MATING FRACTION ,S 1,A 1 *TO* 7 ***( .10,.40,.80,.95,1.0,1.0,1.0 )DATA1   84
FRMATM   MALE MATING FRACTION ,S 1,A 1 *TO* 7 ***( .10,.40,.80,.95,1.0,1.0,1.0 )DATA1   85
FRSPAWN S 1, M 1 *TO* 12 ***( .20,.10,.06,.02,.04,.06,.06,.22,.40,.34,.34,.16 )  DATA1   86
FEGGS  MAGNITUDE OF EGGS PER FEMALE = 1000.0                                          DATA1   87
EFEM   EGGS / FEMALE, S 1,A 1 *TO* 7 = ***( 5.2,8.6,11.7,14.4,16.6,18.3,19.6 )        DATA1   88
NOGRND  NUMBER OF SPAWNING GROUNDS STOCK 1 = 1                                        DATA1   89
SPGRND  INDEX OF SPAWNING GROUND STOCK 1 LOC 1 = 1                                    DATA1   90
C                                                                                     DATA1   91
C      RECRUITMENT                                                                    DATA1   92
C                                                                                     DATA1   93
RECRL  LENGTH OF RECRUITS FOR STOCK 1 = 8.9 CM                                        DATA1   94
SURVN  NORMAL EGG SURVIVAL RATE FOR STOCK 1 = 0.00030                                 DATA1   95
SMULT  EGG SURV MULT S 1 K 1 *TO* 10 ***( 22.2,1.9,1,.68,.51,.41,.35,.30,.26,0 )DATA1   96
RECINT INTERVAL FROM SPAWNING TO RECRUIT, STOCK 1 = 6 MONTHS                          DATA1   97
FRECR  MAGNITUDE OF RECRUIT NUMBERS 1000000000                                        DATA1   98
RECRT  R/MO S 1,G 1,M 1 *TO* 12 ***( 12.49,45.81,83.29,70.79,70.79,33.31,0 )          DATA1   99
C                                                                                     DATA1  100
C      MORTALITY                                                                      DATA1  101
C                                                                                     DATA1  102
AVFR   FR,AV/MO S 1,L 1, M 1 *TO* 9 ***( .72,.74,.76,.78,.80,1.0,.93,.67,.53 )  DATA1  103
AVFR   FR,AV/MO S 1,L 1, M 10 *TO* 12 ***( .48,.68,.70 )                              DATA1  104
FCOEFF MAGNITUDE OF COEFFICIENT OF CATCHABILITY = 0.001                               DATA1  105
QCATCH  COEFFICIENT OF CATCHABILITY S 1,A 1 *TO* 7  ***( 0.14,1.37,9.04 )             DATA1  106
NMORT  NATURAL MORTALITY S 1 AGE 1 *TO* 7 = 1.10                                      DATA1  107
```

```
C                                                                                   DATA1  108
C      GROWTH                                                                       DATA1  109
C                                                                                   DATA1  110
LINFN   ASYMPTOTIC LENGTH, STOCK 1 = 15.91 CM                                       DATA1  111
VBK     A VON BERTALANFFY PARAMETER FOR STOCK 1 = 0.32                              DATA1  112
TZERO   FOR STOCK 1 = -2.08                                                         DATA1  113
LMULT    LENGTH INCREMENT MULT S 1,K 1 *TO* 10 = 1.0                                DATA1  114
VBQ      LENGTH-WEIGHT MULTIPLIER STOCK 1 = 0,00000001016                           DATA1  115
ALNGTH LENGTH S 1,A 1 *TO* 7,L 1 ***( 7.9,10.1,11.7,12.9,13.7,14.3,14.7 )           DATA1  116
WEIGHT S 1,A 1 *TO* 4,L 1 = ***( .0000051,.0000106,.0000163,.0000216 )              DATA1  117
WEIGHT S 1,A 5 *TO* 7,L 1 = ***( .0000261,.0000297,.0000326 ).                      DATA1  118
MARK     MARKING FRACT S 1,M 1 *TO* 12 ***( 0,0,0,0,0,0,0,.17,.18,.25,.25,.15 )     DATA1  119
NCYCL    NO. CYCLES S 1, M 1 *TO* 12 = 4                                            DATA1  120
WF      SUMMATION MATRIX FOR STOCK 1,A 1 *TO* 7 = 1                                 DATA1  121
                                                                                    DATA1  122
C        -----------BLOCK 8 - PROCESSOR DECISIONS                                   DATA1  123
OPLEVI   OPERATING LEVEL PROC 1 STOCK 1 = 30576                                     DATA1  124
PURPRI   PURCHASE PRICE PROC 1,STOCK 1 = 25                                         DATA1  125
PRICES SELLING PRICE PROC 1 PROD 1 MARK 1 = 250                                     DATA1  126
PRICES SELLING PRICE PROC 1 PROD 2 MARK 2 = 110                                     DATA1  127
                                                                                    DATA1  128
C        -----------BLOCK 9 - HARVESTER DECISIONS                                   DATA1  129
EFFOI    NO. OF HARVESTING UNITS HARV 1 STOCK 1 = 76                                DATA1  130
HCAPAC   CAPACITY PER VESSEL PER TRIP HARV 1 STOCK 1 = 191                          DATA1  131
FPWOI    AVE. HARVESTING EFFICIENCY HARV 1 STOCK 1 = 1.726                          DATA1  132
DYSTRP   TRIP TIME IN DAYS HARV 1 STOCK 1 = 1.0                                     DATA1  133
HLOCX    HARVESTING LOCATION HARV 1 = ( 0,25 )                                      DATA1  134
                                                                                    DATA1  135
C        -----------BLOCK 10 - REGULATOR                                            DATA1  136
XI     FLEET-GROWTH,SEASON CONTROL FOR STOCK 1 LOC 1 = 3                            DATA1  137
NQUT     1 *TO* 2 = 1 ANNUAL AND MONTHLY QUOTAS NOT ENFORCED                        DATA1  138
OPLL    MAX OPERATING LEVEL P 1 S 1 = 10000000                                      DATA1  139
EFFL     MAX NO. HARVESTING UNITS STOCK 1 LOC 1 = 500                               DATA1  140
RSELT    HARVESTING SEASON STOCK 1 LOC 1 = 30                                       DATA1  141
FPML     HARVESTING EFFICIENCY LIMIT STOCK 1 LOC 1 = 20.00                          DATA1  142
                                                                                    DATA1  143
```

# Appendix 3 - Typical financial and physical output reports

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
THE FOLLOWING FINANCIAL STATEMENT HAS BEEN ACCUMULATED OVER THE PAST   1 YEARS.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

///SUMMARY REPORT.  YEAR   10. MONTH 12///   PROCESSOR SET    1 --- HARVESTER SET    1

           .. INCOME STATEMENT ..                                 .          .. STATEMENT OF CASH FLOW ..
TOTAL SALES                                    $ 5.8020E+07      .  TOTAL SALES                          $ 5.8020E+07
LESS - DIRECT COST OF SALES  $ 2.6972E+07                         .  NEW LOANS                            $ 0.
       MARKETING EXPENDITURES$ 0.                                 .    TOTAL CASH PROVIDED                              $ 5.8020E+07
       ADMINIS. AND OVERHEAD $ 1.9588E+07                         .  LESS - DIRECT LABOR COST             $ 0.
       LICENSES AND TAXES    $ 0.                                 .         COST OF RAW MATERIALS         $ 1.9060E+07
       DEPRECIATION          $ 0.                                 .         OPERATING COSTS               $ 7.9120E+06
       RESEARCH AND DEVEL.   $ 0.                                 .         MARKETING EXPENDITURES        $ 0.
       INTEREST PAID         $ 0.                                 .         ADMINIS. AND OVERHEAD         $ 1.9588E+07
       TOTAL CURRENT EXPENSES                  $ 4.6560E+07      .         LICENSES AND TAXES            $ 0.
                                                                  .         RESEARCH AND DEVEL.           $ 0.
                                                                  .         INTEREST PAID                 $ 0.
                                                                  .         ADDL. PLANT AND EQUIP.        $ 0.
                                                                  .         LOAN REPAYMENT                $
                                                                  .           TOTAL CASH OUTLAY                             $ 4.6560E+07
NET INCOME                                     $ 1.1460E+07      .  NET CASH FLOW                                      $ 1.1460E+07
                                                                  .
..........................................................................................................

----------
///YEAR   10. MONTH 12/// REPORTS FOR STOCK SECTOR:

CUMULATIVE CATCH
STOCK 1 IN LOC 1     7.624E+05
----------

----------
///YEAR   10. MONTH 12/// REPORTS FOR PROCESSOR SECTOR:

UNITS SOLD BY          PROD 1     PROD 2     PROD 3     PROD 4     PROD 5     PROD 6     PROD 7     PROD 8     PROD 9     PROD 10
PROC 1 IN MARK 1     1.525E+05  0.
PROC 1 IN MARK 2     0.         7.624E+03

UNIT SALES PRICE BY    PROD 1     PROD 2     PROD 3     PROD 4     PROD 5     PROD 6     PROD 7     PROD 8     PROD 9     PROD 10
PROC 1 IN MARK 1      250        0.
PROC 1 IN MARK 2     0.          110
----------

----------
///YEAR   10. MONTH 12/// REPORTS FOR HARVESTER SECTOR:

UNITS HARVESTED        STOCK 1    STOCK 2    STOCK 3    STOCK 4    STOCK 5    STOCK 6    STOCK 7    STOCK 8    STOCK 9    STOCK 10
BY HARVESTER   1     7.624E+05

UNITS UNSOLD           STOCK 1    STOCK 2    STOCK 3    STOCK 4    STOCK 5    STOCK 6    STOCK 7    STOCK 8    STOCK 9    STOCK 10
BY HARVESTER   1     0.

NO. OF HARV. UNITS     FLEET 1    FLEET 2    FLEET 3    FLEET 4    FLEET 5    FLEET 6    FLEET 7    FLEET 8    FLEET 9    FLEET 10
FOR HARVESTERS  1      122

HARVESTING TIME
(IN DAYS)              FLEET 1    FLEET 2    FLEET 3    FLEET 4    FLEET 5    FLEET 6    FLEET 7    FLEET 8    FLEET 9    FLEET 10
FOR HARVESTERS  1      215
----------
```