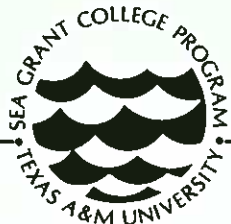


# File Copy

## User Guide for General Bioeconomic Fishery Simulation Model (GBFSM)



Wade L. Griffin  
William E. Grant  
TAMU-SG-83-204 • April 1983



USER GUIDE  
FOR GENERAL BIOECONOMIC  
FISHERY SIMULATION MODEL (GBFSM)

by

Wade L. Griffin  
Department of Agricultural Economics

and

William E. Grant  
Department of Wildlife and Fisheries Sciences  
Texas A&M University  
College Station, Texas 77843

April 1983

TAMU-SG-83-204

Partially Supported through Institutional Grant NA81AA-D00092

to Texas A&M University

by the Office of Sea Grant

National Oceanic and Atmospheric Administration

U.S. Department of Commerce

Text available from:

Marine Information Service  
Sea Grant College Program  
Texas A&M University  
College Station, Texas 77843  
\$5

Data set tape available from:

Dr. Wade L. Griffin  
Department of Agricultural Economics  
Texas A&M University  
College Station, Texas 77843  
\$50

TAMU-SG-83-204  
300 April 1983  
NA81AA-D00092  
R/F-3

## TABLE OF CONTENTS

	Page
INTRODUCTION .....	1
CONCEPTUAL MODEL .....	2
DESCRIPTION OF PROGRAM .....	4
Transportability .....	4
Variable Dimensioned Arrays .....	4
Setup, Use and Modification of the Model .....	5
VARIABLE INPUT DESCRIPTION .....	8
1. Model Parameter for Driver Program .....	9
2. Options Available .....	9
3. Time Parameters .....	11
4. Recruitment and Movement Coefficients .....	13
5. Growth Coefficients .....	14
6. Natural Mortality Coefficients .....	14
7. Harvest Coefficients .....	16
8. Relative Fishing Power Coefficients .....	17
9. Economic Coefficients .....	18
10. Adjustment of Effort .....	19
11. Closure Parameters .....	20
12. Labels .....	21
13. Stochastic Parameters .....	21
14. Nominal Days Fished .....	22
15. Flexibility Equations (Optional) .....	22
SUBJECTIVE ESTIMATION USING ITERATIVE SIMULATION PROCEDURE .....	24
MODEL VALIDATION .....	29
ESTIMATING CURVES WITH GBFSM .....	30
EXAMPLES OF POLICY ANALYSIS .....	34

TABLE OF CONTENTS (Cont'd.)

	Page
REFERENCES .....	46
Appendix A - TABLE .....	47
Appendix B - SBUILD .....	59
Appendix C - SORIGIN .....	63
Appendix D - JCL .....	97
Appendix E - SIMULATE .....	101
Appendix F - OUTPUT OF GBFSM .....	107

## List of Figures

Figure		Page
1	Conceptualization of the major biological and economic aspects of an annual crop marine fishery .....	3
2	Movement of fish as indicated by arrows .....	15
3	Yield curves at various levels of ICOF .....	25
4	Landings offshore at 766 nominal days fished .....	27
5	Actual annual landing offshore (O) compared to GBFSM estimated landing offshore ( $\Delta$ ) .....	28
6	Comparison of GBFSM simulated curves versus Garcia's estimates and annual actual data .....	31
7	Comparison of GBFSM simulated data to actual data .....	32
8	Comparison of annual landings offshore (a) and revenue and cost (b) for Model 1 for CMESH = 20.03 (—) and CMESH = 23.11 (----) in inshore harvesting .....	36
9	Yield curves of boats for baseline (solid line) and for closing the Texas offshore June and July (dashed lines) .....	37
10.	Yield curves of vessels for baseline (solid line) and for closing the Texas offshore June and July (dashed lines) .....	38
11.	Total value product and total factor cost curves of boats for baseline (solid lines) and for closing the Texas offshore June and July (dashed lines) .....	39
12	Total value product and total factor cost curves of vessels for baseline (solid lines) and for closing the Texas offshore June and July (dashed lines) .....	40
13	Yield curves of vessels for baseline (solid line) and for eliminating Texas Count Law (dashed lines) .....	43
14	Total value product and total factor cost curves of vessels for baseline (solid lines) and for a Texas No Count Law (dashed lines) .....	44

## INTRODUCTION

An objective of marine fisheries management is to ensure wise use of the renewable natural resources of the oceans. Systems analysis and simulation are finding increased application, and mathematical models representing the behavior of marine fisheries systems is an integral part of many, if not most, management programs. This manual describes in detail a General Bioeconomic Fisheries Simulations Model (GBFSM) designed for use in management programs of marine fish species that do not exhibit a significant relationship between the size of the parental population and the number of young recruited into the fishery. Uses of GBFSM to evaluate alternate management policies for a specific marine fishery are presented in the reference list.

The purpose of GBFSM is to predict how alternate management policies will affect a fishery. Effects are assessed in terms of total harvest; species, size-class and seasonal distribution of the harvest; total revenue, fishing costs and rent in the fishery; and distribution of revenue, costs and rent among different classes of fishing vessels. The user may select any number of species, size classes, fishing areas, depths and vessel classes for inclusion in the model. The time-step, extent and resolution of model output are also variable. The model can be deterministic, or it can have stochastic components. GBFSM's design is versatile enough to be applicable to a wide range of fisheries. Therefore, the list of options and parameters is greater than might be needed for a particular problem.

Information needed and steps involved in the use of GBFSM for a given fisheries management problem are presented in seven sections: (1) a concep-

tual model, showing the major parts and general flow; (2) a description of the general operation of the program; (3) a description of the input and available options; (4) a discussion on how to subjectively estimate unknown variables; (5) a discussion of model validation; (6) a discussion of how to estimate revenue cost-and-yield curves with GBFSM; and (7) a description of how the model can be used for policy analysis, with examples. In addition, Appendix A includes a table for variables, data description and data format information and Appendices B through F contain supporting information.

We suggest that the user first read the description sections to obtain a general understanding of the model's operations, then that he or she use the table in Appendix A to set up data cards. That table provides enough information that, with a little experience, the user can become familiar with all available options.

#### CONCEPTUAL MODEL

A conceptualization of the major biological and economic aspects of an annual-crop marine fishery system is presented in Figure 1. The biological submodel represents the recruitment of new organisms into the fishery by species, sex, and fishing area. Recruitment is always in depth one, and movement is always to a greater depth and between fishing areas. Organisms grow and move from one size class to another, and mortality results from both natural causes and fishing. Bait fishing and recreational fishing can remove organisms from depth one only in each area. By-catch of non-target species can occur as a result of fishing.

The economic submodel represents the fishing effort (nominal days



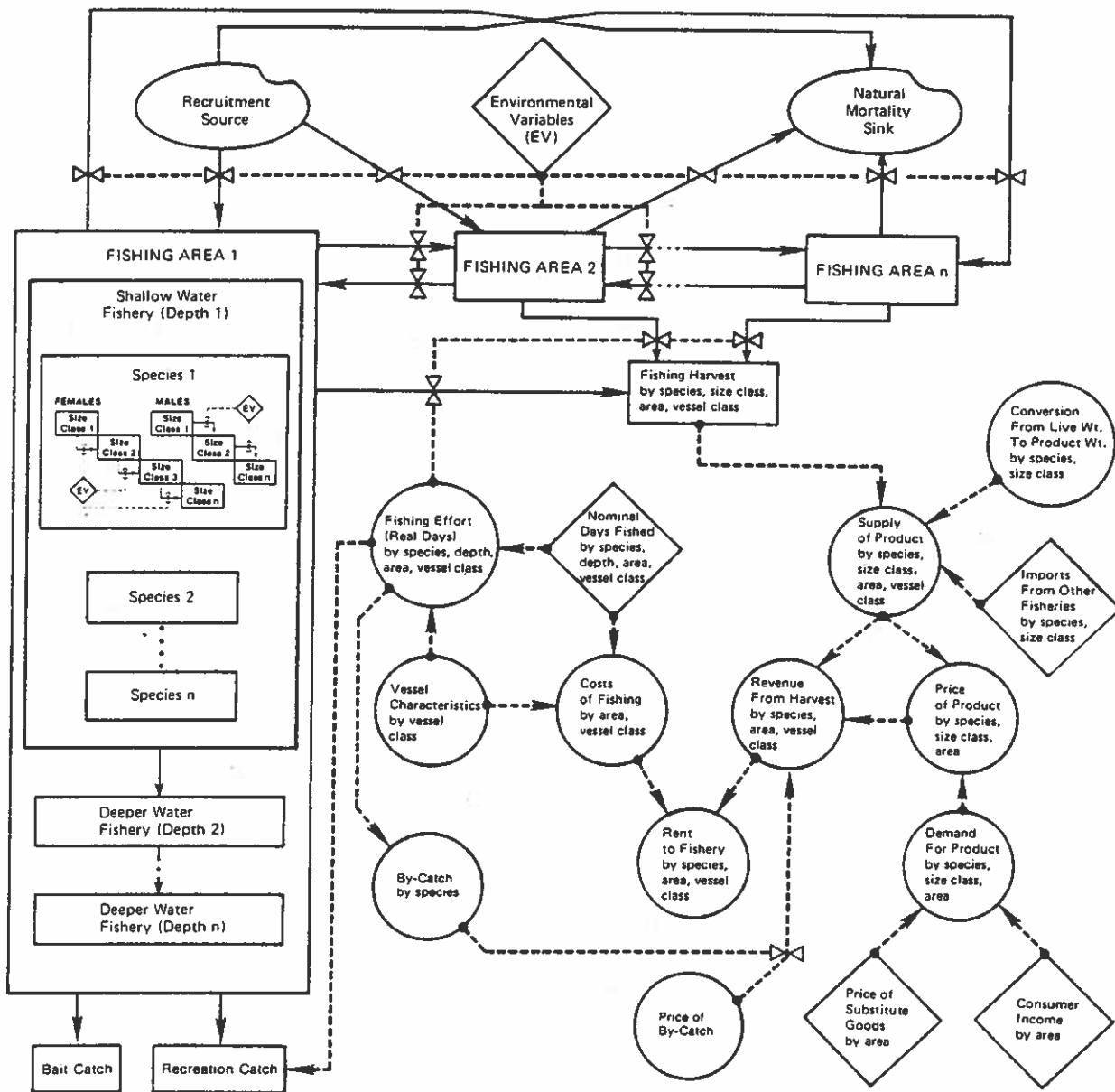


Figure 1. Conceptualization of the major biological and economic aspects of an annual crop marine fishery.

fished converted to real days fished) exerted on each species and on by-catch. Monetary costs of fishing, value of harvest, and rent to the fishery are calculated. The economic submodel allows the option of including flexibility equations to determine price of the harvested catch.

## DESCRIPTION OF PROGRAM

### Transportability

GBFSM consists of a series of FORTRAN programs and the necessary JCL (Job Control Language) to set up and execute them. The simulation system itself should be an extremely "transportable" system because of its adherence to standard FORTRAN syntax, available at most computer installations. This is a desirable characteristic, because it allows anyone familiar with a given computer system to set up GBFSM without knowing how GBFSM works. The user needs to know only how to use GBFSM.

### Variable Dimensioned Arrays

It was undesirable to set a maximum size on the arrays because most of the time this maximum value would never be approached when the system is used. This problem required that the size of the arrays in the FORTRAN subroutines be variable. FORTRAN language specifications state that an array may be variable-dimensioned in a sub-routine only if it has been fixed-size dimensioned at some other place. FORTRAN allocates storage space for variables when it is compiled and variables may not exceed this storage space value when the program is run. This type of storage allocation is called static allocation. A language that allocates space for program variables during execution (i.e., one that allows dynamic allocation)

would not have this problem (e.g., PLI, ALGOL, or PASCAL). However, those languages are machine-specific and, therefore, not transportable.

To overcome the problem, a two-step procedure was developed allowing the system to be written in FORTRAN. This two-step procedure allows arrays of different sizes at different points during the life of the system. Thus, the user can change the size of the array to fit his particular problem, and the system is easily transported from one system to another.<sup>1</sup>

### Setup, Use and Modification of the Model

The programs and JCL for installing GBFSM on a computer, running simulations with the GBFSM, and modifying GBFSM are described below. All examples of JCL are for use on the AMDAHL computer at Texas A&M University.

The following programs and JCL comprise GBFSM:

SBUILD, shown in Appendix B, is a FORTRAN program that reads the model parameters (array sizes) and creates a driver program for the subroutines by using a simple "fill-in-the-blank" technique involving simple WRITE/FORMAT statement combinations.

JCLBUILD. The JCL for running this program is the first set of JCL in Appendix D. This is the first program that must be run in installing the GBFSM system on the computer. It creates a load module (binary program) called NEW(LMNBUILD).

SORIGIN is a complete version of the simulation program, complete with a sample driver program, written in FORTRAN. It is shown in Appendix C.

---

<sup>1</sup>There are some fixed dimensioned arrays in the two subroutines DEMAND and SASOUT. If these array sizes are exceeded, the user must change the size of these arrays in these two subroutines.

JCLSETUP is the second set of JCL shown in Appendix D. This JCL uses SORIGIN to create the load module NEW(LMNSYSTEM) and must be run second after SBUILD.

JCLMODEL is the third JCL set shown in Appendix D. This JCL deck with data specifies the model parameters to be used by the program that builds the drive routine. It involves the load module NEW(LMNBUILD) created from SBUILD and creates a FORTRAN driver that is then compiled and overlaid into the load module NEW(LMNSYSTEM), replacing the old driver. This JCL must be run third to establish array sizes of the model parameters. Model parameters are defined in Section 1, page 48. This JCL is also used any time the array sizes must be changed.

JCLSIM is the fourth set of JCL found in Appendix D. This JCL deck involves the current version of the system NEW(LMNSYSTEM) and produces a simulation run.

SIMULAT1 and SIMULAT2 are two different data input streams used to obtain simulation runs and are shown in Appendix E. The data input is described in Sections 2 through 16 on pages 9 through 50. Appendix F shows the output for SIMULAT1.

JCLCHBIN is the fifth set of JCL found in Appendix D. It is used to replace any subroutine in the load module NEW(LMSYSTEM) that the user wants to modify. The user simply makes the modifications to the appropriate subroutine (or subroutines) from SORIGIN and then files that subroutine(s) in a work file called CHBIN. JCLCHBIN then compiles the subroutine(s) in CHBIN and replaces them in the load module NEW(LMSYSTEM).

In summary, SBUILD is run first and only once using JCLBUILD, creating the load module NEW(LMNBUILD). Next, SORIGIN is run only once using JCLSETUP

to create the load module NEW(LMSYSTEM). Third, JCLMODEL is run to set the array sizes for the model parameters and is run to change the array sizes when needed. SIMULAT1 (or SIMULAT2 or any data stream created by the user) is then used along with JCLSIM to make simulation runs.

JCLCHBIN is used whenever a subroutine must be modified in the load module NEW(LMNSYSTEM).

## VARIABLE INPUT DESCRIPTION

Fifteen different data sets can be input into the system. The first data set (Model Parameters for DRIVER) allocates the array sizes of the program and is used with the JCL deck called NEWMODEL in Appendix E. The second data set (Option Available) determines what option is to be executed in the program. The next 12 sets of required data contain various types of information, such as growth mortality, movement, harvesting, economics, etc. The last set of data is optional and is information to utilize a demand equation. Input data 2 through 15 are used with the JCLSIM deck in Appendix D.

The subheading in this section corresponds to the numbered heading in Appendix A. In addition, Appendix E contains two examples of data sets or FORTRAN flows utilized by GBFSM, which will be referred to clarify the discussion. In the FORTRAN flow in sections 2 through 15, the variable name "Q" is used each time a new section starts and/or each time a new set of data is started within a section. "Q" is used to describe the section and/or set of data the user is inputting. In using the model, the user constantly refers to the data set, and this labeling is very helpful.

Some variables in the FORTRAN flow are dimensioned by one or more arrays. These data sets may require one to several data cards each, depending on the size of arrays for each variable. In this program, all data are entered by incrementing the array on the right and working to the left, according to the dimension statement of the variable. For example, ICOF (NSP, NA) is dimensioned by area (NA) where fish are caught and by the number of species (NSP) caught. If  $NA = 5$  and  $NSP = 2$ , then 10 values must be entered, and the data are entered by incrementing NA first as:

## ICOF Values

---

Data Card 1 (1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 1), (2, 2), (2, 3)

Data Card 2 (2, 4), (2, 5)

Notice that there are 10 values, of which eight go on data card 1 and two go on data card 2. The user is cautioned to pay close attention to format statements and the number of values that will go in each data card. This procedure is followed through the user guide.

### 1. Model Parameter for Driver Program

Eleven model parameters set the array sizes of variables in the program. These model parameters, with a description of each, are found in Section 1 in Appendix A. Only one data card is required to input these model parameters. These data are used with JCLMODEL only.

The minimum value that any value can have is one (1). If there is no by-catch, NBC must equal one. The Options Available (next sub-section) allow the user not to consider by-catch, even if NBC equals one.

### 2. Options Available

Options available are shown in the table of Appendix A in Section 2 and in lines 1 through 3 in Appendix E in SIMULAT1 and SIMULAT2. The first data card is "Q," which describes the data contained in Section 2 of the FORTRAN flow. The second and third data cards make up the control vector, A, which implements desired options in the program. Section 2 in Appendix A lists the options specified in the control vector. A "T" in the proper field (or combination of "T's" in different option fields) implements an option, while an "F" or a blank space will cause that option to be skipped.

Option A(1) lists the number and sizes of organisms in each cohort or age class for each species, depth and area. Option A(2) specifies that the printouts from A(1) occur at the end of each simulation run if more than one run is included in a job. Option A(3) causes landing for A(4) through A(8) and A(21) through A(23) to be output after each iteration. Option A(4) produces a breakdown, by statistical collection period, of all landings in the fishery. Options A(5) through A(7) produce finer breakdowns of primary catch, and each is described in Appendix A. Option A(8) calls the ECON subroutine and produces, by statistical collection period, a list of relevant economic statistics, such as owner's packing charge, rent, variable cost, and crew's share.

Option A(9) causes the fishing mortality factor to be displayed for each species, area, depth and time-step. If fishing mortality exceeds 100%, the program automatically generates a diagnostic option and renormalizes fishing mortality over all vessel classes to 100% total.

Option A(10) prints the input data in sections 2 through 14. Options A(11) through A(18) are used when a job includes more than one run. The number of iterations to be run is set by the ITER variable in Section 3 in the Table in Appendix A. For each iteration, the program will reinitialize the state variables after each run. Options A(11) through A(13) will stochasticize a chosen variable according to a chosen distribution. Options A(14) through A(18) indicate which variables can be stochasticized.

Option A(19) is used in conjunction with variables YR and TYR, which adjust the days fished (DFN in Section 14) input to the fishery. Options A(26) and A(27) can be used with A(19) to adjust the number of vessels (NV in Section 9) in the fishery. Options A(26) and A(27) adjust NV by YR and



TYR, respectively, only if A(19) is true. YR and TYR are explained in Section 10 in the Table in Appendix A. Option A(20) produces a matrix containing landings, revenue, rent and total costs by species and vessel class for each run in a job and outputs it to an output file. The output file must be assigned a DD name and specified on the card after the STEPLIB card in the JCL section (See Row 6 in Appendix D, Set 4).

Option A(21) (begin third data card) produces a summarized economics table showing landings, total cost, revenue and rent. It includes opportunity costs and fixed costs in all economic calculations and outputs the total by vessel class and species. It prints out this information for both vessel owner and crew. It also prints totals that are summarized over vessel class and species, and a grand total that is summarized over vessel owner and crew. Option A(22) prints the same information as Option A(21) but by vessel class, and Option A(23) prints the same information by species. Option A(24) calls the DEMAND subroutine containing the flexibility equations when calculating prices. Option A(31) prints out the prices that are calculated.

Option A(25) allows the user to print monthly values in A(3) through A(8). Option A(28) and A(29) causes calculations of by-catch and related information to be printed out. A(28) calculates by-catch as a function of effort, and A(29) calculates by-catch as a function of catch of the target species.

### 3. Time Parameters

This group contains a number of variables related to closures, iteration and time. The first two cards are description cards (Q) and the third

card is the data card, including time parameters. The first variable is IC and indicates the number of closures in a given time period. The maximum number of closures per period is 18. When IC has a value of one or greater, the third data card in Section 11 (Closure Parameters) must be included. For example, in Appendix E, SIMULAT1, row 6, IC = 4; therefore, four data cards (rows 126-129) must be included in Section 11. If IC = 0, no data cards are included, as shown in Appendix E, SIMULAT2.

ITER is the number of periods or iterations required per run. For example, if a period equals one year and the user wants to run the same year 20 times for stochastic purposes, a 20 would be entered in the appropriate columns. FTIME is the number of time steps per period. For example, if the time period is one year and the user want fish growth, mortality, etc. to be on a weekly time step so that each cohort is updated weekly, the user would enter 48 as shown in Appendix E, SIMULAT1, Row 6. CCT can be used to adjust how often the cohort is updated. In Appendix E, SIMULAT1, Row 6, because the time step is one a week and cohorts are to be defined on a weekly basis, CCT is 1.0. If cohorts are to be defined on a two-week basis, then CCT would be 2.0.

CN, CER and CPH are timing coefficients for the time-dependent variables associated with NM, NER and NPH, respectively. Each may vary on a different time schedule (month, day, etc.). Each is derived from the following equation, where C is the respective timing coefficient (CN, CER, or CPH):

$$C = \text{FTIME} / (\text{NM, NER or NPH, respectively}) + E,$$

where  $E < 1/(\text{NM, NER or NPH, respectively})$ . For example, if the time period is one year, and the simulation is being run on a weekly time-step (FTIME = 48), with NM equal to 12 (monthly, statistical collection periods),

then

$$CN = 48/12 + E = 4.0 + E$$

Because  $1/12$  is slightly more than .0833, a suitable value for CN is 4.01, as shown in Appendix E, SIMULAT1. For another example, if FTIME = 12 and NM = 12, then CN = 1.01, as in Appendix E, SIMULAT 2.\*

C8 is the minimum number of individuals that can be in a cohort before it is eliminated.

#### 4. Recruitment and Movement Coefficients

Young fish are introduced into the fishery at some maximum rate, ICOF (NSP, NA), per time step, which is adjusted according to a time-dependent factor (NSP, NER) that varies from 0 to 1. When fish are introduced, the number in the currently filling cohort is assumed to be zero. When the number in a cohort is less than a certain minimum (C8, see section 3) the cohort is automatically zeroed. The new organisms are all given an initial size ISZ(NSP, NSX) when entering the currently filled cohort. If each species (NSP) is distinguished by sex (NSX = 2), then males and females can be entered at different sizes. If the user assigns values to E such that  $\sum E(NSP, \cdot) = 1$ , then ICOF(NSP, NA) is an estimate, by species and area, of the number of organisms entering the fishery at size ISZ(NSP, NSX) when NSX = 1 and  $1/2$  the number of organism when NSX = 2.

Fish can move between depths and areas. The variable ER(NSP, ND, NA, NA),

---

\* If NM = 12 and FTIME = 52 for the actual number of weeks in a year, then there will be five time steps every three statistical collection periods and four time steps in all the other statistical collection periods. It is suggested that the user use only 48 weeks per year so there will be an equal number of time steps for each statistical collection period.

dimensioned by species, depth, area moved from, and area moved to, respectively (see Figure 2), contains the percentage of individuals moving from one cell to another. After it has been determined whether or not the organisms in a chart are of a size greater than SZCOF(NSP, NSX), the number moving between the first and second depths is calculated, if there is more than one depth. Transfers between any subsequent depths are calculated in the same manner, except that there are no size restrictions on movement in depths greater than depth one. If there is only one area, then the calculation stops there; otherwise, the net number entering and leaving from the adjacent area(s) is calculated.

#### 5. Growth Coefficients

The von Bertalanffy growth equation is used to calculate the growth of organisms, using the parameters of the growth coefficient, GCOF(NSP, NSX), and the upper asymptotic length, MSZC(NSP, NSX). The growth is also modified by multiplication with a time-dependent growth factor, GRT(NSP, NPH). Lengths of fish are converted to weights using the coefficients C3 (NSP) and C4(NSP) in the following equation:

$$\text{WEIGHT} = \text{C3}(\text{length})^{\text{C4}}$$

#### 6. Natural Mortality Coefficients

Natural mortality is calculated by multiplying the constant NMCOF (NSP, ND, NSC) by the number of organisms present in that cohort to obtain the percentage killed.

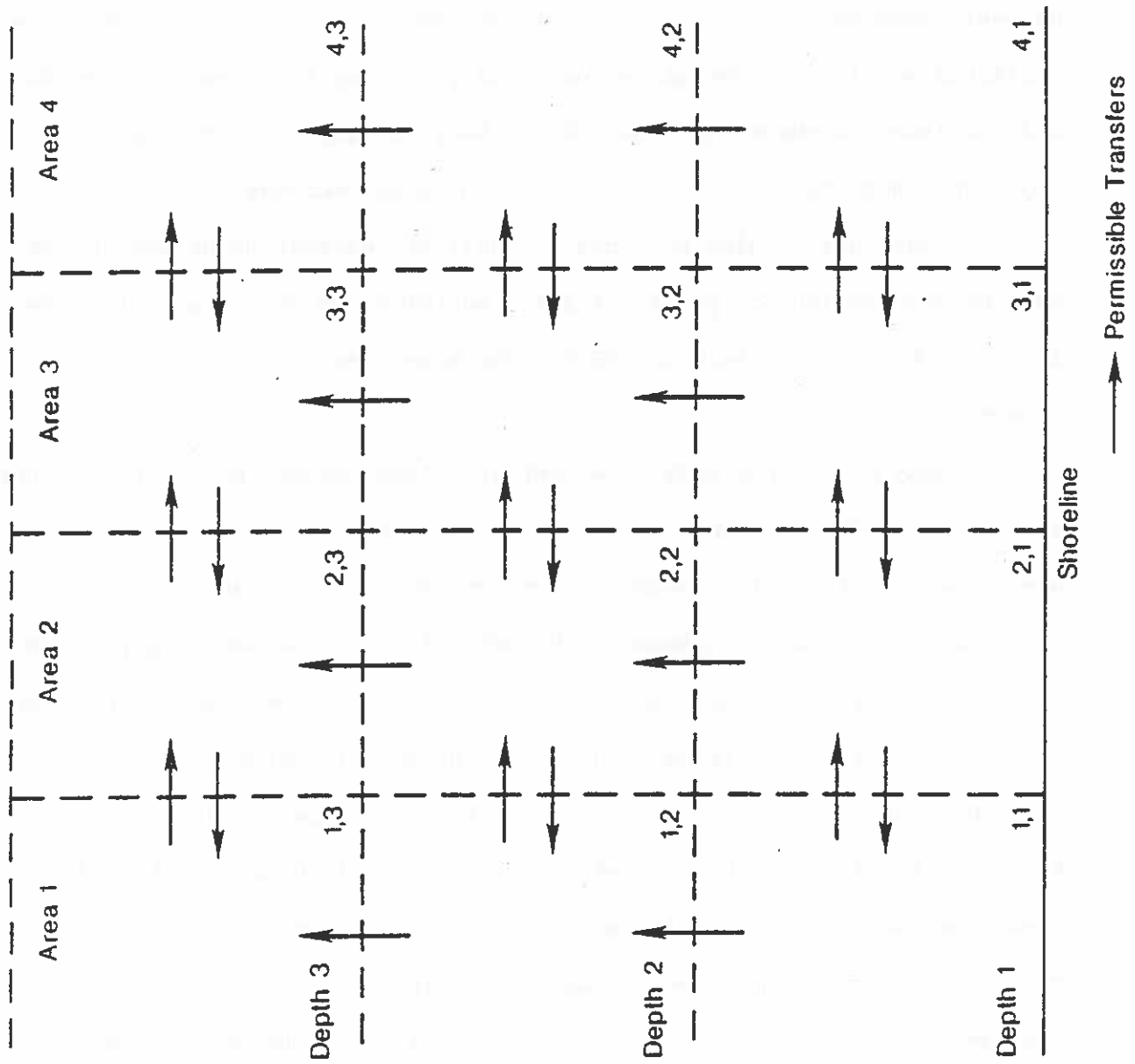


Figure 2. Movement of fish as indicated by arrows.

## 7. Harvest Coefficients

Harvest coefficients pertain to bait, recreational and commercial fishing for the target species and by-catch. In many fisheries there are secondary and tertiary fishing mechanisms. Bait fishing provides a way to harvest organisms on a straight percentage basis where no effort data are available available. Mortality due to bait fishing is represented by BC, and the lower catchable size is BMESH. Bait fishing is restricted to depth one, and the size class of the catch is not recorded.

Recreational fishing provides an auxiliary harvest mechanism, based on a percentage (RC) of primary fishing mortality in depth one only. The lower catchable size limit is RMESH. The size class of the catch is not recorded.

In many cases, the form or weight of a fish changes before it is sold at the dock. ETM is a conversion ratio that changes the units used to express the weight of fish when they are growing or are caught (e.g., grams) to those units used to express weights when fish are landed (e.g., pounds).

For commercial harvesting, CMESH(NSP, ND) represents the lower catchable size limit. To eliminate knife-edge effects in harvest, organisms between C5 and C6 in size, generally smaller than legal catch size, are killed by the fraction C7 of fishing mortality. All organisms killed between the lower catchable size and the lower legal size are counted as culls, except for a small percentage, which are counted in the smallest harvestable size class of legal commercial catch and are stored by depth, vessel class and size class. CF(NVC, ND) is the percentage of culls that can be counted in the legal commercial catch. SL(NSP, NSC) is the boundary between size classes of fish and is related to the ex-vessel

price of fish (PMCSP) found in Section 9.

VKIL(NSP, ND, NM) is the smallest size that can be legally landed. FMAX(NSP, ND, NA) is the density factor for commercial fishing mortality. If the element VKIL(NSP, ND, NM), corresponding to species, depth, and statistical collection period, contains a size less than or equal to the lower limit of the smallest size class (SL), all the organisms killed are placed in the lower size class. If VKIL contains larger values, then culls can occur. An example of VKIL is given in Appendix E, SIMULAT1. Values for VKIL are given in rows 55 through 60. Row 55 is species 1, depth 1 (bay waters) for January through December. The value is 45.0 mm for May through July (the value of CMESH in Row 46). However, a law sets the minimum legal size of species 1 caught in bay waters in January through April and August through December at 67.6 mm. That percentage (1-CF) of fish caught between 45.0 mm and 67.6 mm will be culls.

By-catch (fish other than the target species) is calculated as a factor of days fished. This factor is PERF(NBC, NVC, NM).

#### 8. Relative Fishing Power Coefficients

Because vessels of different sizes and construction have different fishing powers per nominal days fished, nominal days must be converted to real days fished. Real days fished are simply the product of relative fishing power and nominal days fished. Relative fishing power (RFP) is calculated as

$$RFP(NVC) = \frac{HP(NVC)^{C1(NSP)} \quad FLR(NVC)^{C2(NSP)}}{HP(1)^{C1(NSP)} \quad FLR(1)^{C2(NSP)}}$$

where the first vessel class is always the standard vessel.

## 9. Economic Coefficients

This section contains all the economic information used in the model except the flexibility equations. The first variable, CONV(NSP), converts the fish that is caught to the product that sold at the dock. For example, shrimp may be headed at sea and sold heads-off at the dock. CONV(NSP) would then be used to convert the weight of shrimp from a heads-off. NV(NVC) is the number of vessels in each class. C(NVC) is the number of crew for each vessel class. If the vessel is owner-operated, then C(NVC) does not include the captain. If the owner hires a captain and crew, then C(NVC) includes both captain and crew.

FC(NVC) is the fixed cost per period per vessel. If the period is one year, then FC(NVC) is the annual fixed cost per vessel. COST(NVC, ND) is the variable cost per vessel per nominal day fished by vessel class and depth. Depth is included because the cost per nominal day fished varies depending on how far the vessel must travel. On some vessels, the crew shares part of the variable cost; CVC(NVC) is the crew's variable cost per nominal day fished by vessel class. OCO(NVC) is the owner's opportunity cost per period per vessel by vessel class; whereas, OCC(NVC) is the crew's opportunity cost per period per vessel per crew member. PC(NVC) has been included for those vessels that are charged at the dock for each unit of fish landed. SHARE(NVC) is the percentage share, by vessel class, of landings that the crew received. PMCSP(NSP, NSC, NM) is the ex-vessel price of fish by species, size class and statistical reporting division. In Appendix E, SIMULAT1, Rows 99 through 114 show the ex-vessel prices of shrimp. The first row (99) is the price per pound of heads-off shrimp for PMCSP(1, 1, NM). In this case, NM is in months and row 99 reads January



through December from left to right. Because there are two species and eight size classes of shrimp, rows 99 through 106 describe species 1 and rows 107 through 114 describe species 2. Rows 99 and 107 are the prices for the larger size class of shrimp, and rows 106 and 114 are the prices for the smaller size class of shrimp.

Costs for owner are kept separate from crew cost. Total cost for owners is calculated as

$$\begin{aligned}
 TC_o = & \text{COST}(NVC, ND) * \text{DFN}(NSP, \cdot, \cdot, \cdot, NVC, KM) \\
 & + (\text{FC}(NVC) + \text{OCO}(NVC)) * \text{NV}(NVC) * \frac{\text{DFN}(NSP, \cdot, \cdot, \cdot, NVC, \cdot)}{\Sigma \text{DFN}} \\
 & + \text{SHARE}(NVC) * \text{Landings} * \text{PMCSP}(NSP, NSC, NM) \\
 & + \text{PC}(NVC) * \text{SHARE}(NVC) * \text{Landing}
 \end{aligned}$$

Notice that FC(NVC) are porportional by nominal days fished.

Total cost for the crew is calculated as:

$$\begin{aligned}
 TC_c = & \text{CVC}(NVC) * \text{DFN}(NSP, \cdot, \cdot, \cdot, NVC, \cdot) \\
 & + \text{OCC}(NVC) * \frac{\text{DFN}(NSP, \cdot, \cdot, \cdot, NVC, \cdot)}{\Sigma \text{DFN}}
 \end{aligned}$$

The last variable in this section is the price per unit of by-catch, PCATCH(NVC), expressed in the same units as the units of by-catch caught. It only varies between species of by-catch.

#### 10. Adjustment of Effort

YR(100) is a set of 100 numbers used to modify fishing pressure during the corresponding iteration or run number, (i.e. if the first number was 0.5, fishing effort by the commercial fleet would be only one half of the inputed effort). If A(19) = T and A(26) = T, there must be a value for YR(100) for each iteration (ITER) in that run of the model. Also, when

YR(100 is used, then  $ITER \leq 100$ . TYR(ITER, NA, NSP, NVC) has a similar function, but the variation exists in iteration, area, species and vessel class. This allows the user to be selective in adjusting effort. The user should be aware that TYR(ITER, NA, NSP, NVC) is actually dimensioned in the model as TYR(NC, NA, NSP, NVC), so that for any run of the model,  $ITER \leq NC$ .

PRNT(ITER) is included so that the user can choose the printed output for any run of the model. For example, if  $ITER = 10$  and the user only wants every second run printed, then the user would input "1" in columns 2, 4, 6, 8 and 10. PRNT(ITER) overrides Option A(3) when "1" is entered; however, a blank or a zero allows Option A(3) to work for that iteration. PRNT(ITER) is actually dimensioned in the model as PRNT(NC), so for any run of the model,  $ITER \leq NC$ .

## 11. Closure Parameters

This section is included so that the user can adjust nominal days fished on a seasonal basis. These adjustments are made by time divisions for landing statistics, time steps within the statistical time division, area, depth and species. In Appendix E, SIMULAT1, IC = 4 (see row 6). Therefore, four data cards must be included in rows 126 through 129 of this section. Because this example is an annual model in which "month" is the statistical time division, KBD = 5 and KED = 5 in row 126, meaning that nominal day fished will be adjusted in May. Because the time steps are in weeks (FTIME = 48), KBP = 1, means that this adjustment will begin the first week in May, and KEP = 2 means it will end after the second week in May. KAR = 1, KDT = 1 and KSP = 1 mean that this nominal days fished

adjustment will take place in area 1, depth 1 for species 1, respectively. ADJ = 0.0 means that nominal days fished will be zero during this period. Thus, the fishery is closed from May 1 through May 15 in area 1 and depth 1 for brown shrimp (species 1). A total of 18 different closures is allowed ( $IC \leq 18$ ).

## 12. Labels

Labels are used when printed output is required by statistical time division. These labels are entered in an array called COL(NM). In Appendix E, SIMULAT1, the time division is months and labels are abbreviations of the months, as shown in row 131.

## 13. Stochastic Parameters

Three different distributions can be used, and IX(J,I) is the random seed corresponding to them. They are the uniform, triangular, and normal distributions (see Options A(11), A(12), and A(13), respectively). Each one uses a random number generator to produce uniformly distributed random numbers between 0 and 1 and convert them to the proper distribution. In each case the random variate is a percentage of the mean or median included in input data for the respective stochasticized parameter. Random factors are incorporated into the parameters through a ZB variable found in the various subroutines.

The algorithm for the triangular distribution was obtained from simple trigonometry. The median value determines the position of the apex of the triangle, which must have an area of one between the upper and lower limits. H represents the height of the triangle and is calculated as

$$H = 2/(SP1 + SP2)$$

The uniform distribution is generated by specifying upper and lower limits of the distribution by specifying the fractional variation from the mean of the variable(s) to be stochastized.

The normal distribution is created by use of GAUSS, with the first parameter being 0.0 (the mean), and the second parameter being the standard deviation, which should be greater than 0.0 and less than or equal to 1.0.

Variables that can be stochastized are the introduction coefficient (ICOF) movement rates (ER), natural mortality coefficient (NMCOF), density factor for fishing mortality (FMAX), and growth rate (GRT) (See Options A(14) through A(18), respectively). The model generates mean standard deviation and upper and lower confidence limits for several different aggregations, depending on which output the user requests. If the uniform distribution is used with ICOF, the standard deviations are meaningless.

#### 14. Nominal Days Fished

Nominal days fished DFN(NSP, NA, ND, NVC, NM) are entered by species, area, depth, number of vessel class, and statistical time division. Appendix E, SIMULAT1 shows DFN(NSP, NA, ND, NVC, NM) in rows 140 through 151.

#### 15. Flexibility Equations (Optional)

The flexibility option cannot be included in the model analysis unless A(24) is true. Data input is only included when A(24) is true.

Price flexibilities by size class of fish are included in the model as FLEX(NSC) (See row 154, in Appendix E, SIMULAT1). Actual landings or mean landings XBAR(NSC, NM) are entered into the model by size class of fish and by statistical time period (See rows 156 through 167, in Appendix

E, SIMULAT1). The price flexibility equation is

$$F = \frac{\frac{\bar{P} - P}{\bar{P}}}{\frac{\bar{X} - X}{\bar{X}}}$$

or

$$P = \bar{P} \left[ 1.0 - F \frac{\bar{P}(\bar{X} - X)}{\bar{X}} \right],$$

where  $P$  is the estimated price of fish by GBFSM;  $\bar{P}$  is the price of fish in the same dollars as costs in section 9;  $X$  is the landings of fish by GBFSM; and  $\bar{X}$  is mean landings, which must be expressed in the same units as those generated by the model.

The mean price,  $\bar{P}$ , is  $PMCS(1, NSC, NM)$  from section 9. Notice that mean price of the first species of fish in the model, the DEMAND subroutine, combines all species in calculation of price. Also, prices are estimated by statistical time division. If the statistical time division is in months, then GBFSM aggregates all species landed during that month. Estimated price by size class of fish for that month is the same for all species.

## SUBJECTIVE ESTIMATION USING ITERATIVE SIMULATION PROCEDURE

Parameters in the model should be estimated using information in the literature whenever possible. When parameters are not available in the literature, these parameter values can be determined, however, from subjective estimation using an iterative simulation procedure. This technique can best be explained by an example.

In a study of the Ivory Coast shrimp fishery, all parameter in the model could be determined from the literature except ICOF and FMAX found in Appendix E, SIMULAT2, rows 9 and 41, respectively. In this example NSP = 1, NA = 1, and ND = 2, therefore, ICOF has one value and FMAX has two values for the two different depths. The goal was to adjust ICOF, FMAX(1), and FMAX(2) such that the model would be a good predictor of historical catches in the fishery. More specifically, a value was arbitrarily assigned to ICOF, and then values of FMAX(1) and FMAX(2) were adjusted such that inshore landings were approximately 628,000 kg at 45,700 nominal days fished and offshore landings were approximately 425,000 kg at 1,531 nominal days fished, representing average catch and effort levels in the fishery for the period 1975 through 1978.

Several sets of values were derived in this manner, and a family of yield curves, relating catch to effort, were generated. As ICOF decreased from  $2.0 \times 10^8$  to  $1.0 \times 10^8$  (and FMAX(1) and FMAX(2) increased from  $4.43 \times 10^{-4}$  to  $1.024 \times 10^{-3}$  and from  $3.5 \times 10^{-4}$  to  $2.38 \times 10^{-3}$ , respectively), the shape of the yield curves changed from a nearly linear curve to a strongly concave curve that leveled off at roughly 450,000 kg landed at effort levels greater than roughly 3,000 nominal days fished (Figure 3).

### YIELD CURVES

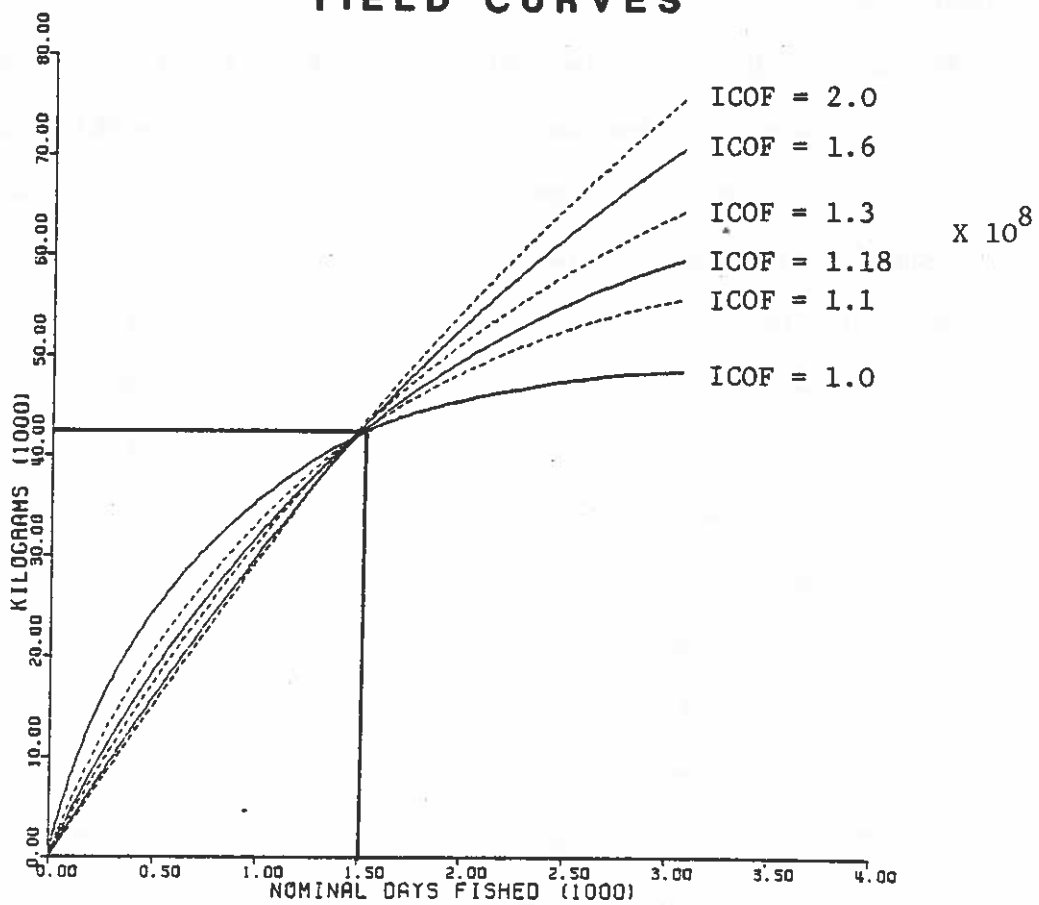


Figure 3. Yield curves at various levels of ICOF

Source: Griffin and Grant, 1982.

The sensitivity of model predictions on offshore landings to changes in ICOF (and FMAX(1) and FMAX(2)) decreases as ICOF increases (and FMAX(1) and FMAX(2) decrease) (Figure 4). Figure 4 relates model predictions of offshore landings to ICOF, FMAX(1) and FMAX(2) when offshore effort is one-half of the average effort levels for 1975-1978. The model's sensitivity to these parameters under relatively low effort levels was of particular interest, because one of the management policies to be evaluated in a study of the shrimp fishery on the Ivory Coast involved reduction in fishing effort.

Because of the uncertainty with which ICOF, FMAX(1), and FMAX(2) could be estimated, and because of the sensitivity of model predictions to changes in the parameters, subsequent analyses in the Ivory Coast study were conducted with two versions of the model representing two different sets of values for ICOF, FMAX(1) and FMAX(2). Values of ICOF, FMAX(1) and FMAX(2) in Model 1 were  $1.3 \times 10^8$ ,  $7.3 \times 10^{-4}$  and  $8.27 \times 10^{-4}$  and in Model 2 were  $1.1 \times 10^8$ ,  $9.0 \times 10^{-4}$  and  $1.4 \times 10^{-4}$ , respectively.

After all unknown parameters have been estimated by this procedure, the user should determine whether discrepancies exist between model estimates and historical values. In the Ivory Coast study, initial simulations inadequately predicted the size-class distribution of actual landings offshore. Landings of larger shrimp were underestimated, and landings of smaller shrimp were overestimated (Figures 5a and 5c). When the values of the growth coefficient (see Appendix E, SIMULAT2, rows 20 and 22) were increased by four and five percent (GCOF and MSZC, respectively) for both males and females in Model 1 and by five and six percent for both males and females in Model 2, model predictions of the size-class distribution of landings off-



### LANDINGS AT 766 DAYS FISHED

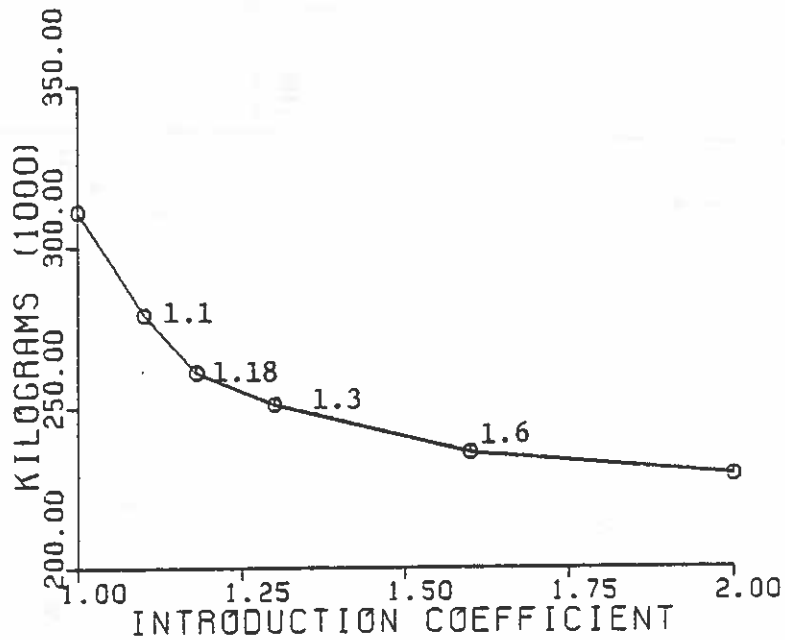


Figure 4. Landings offshore at 766 nominal days fished.

Source: Griffin and Grant, 1982.

## SIZE DISTRIBUTION

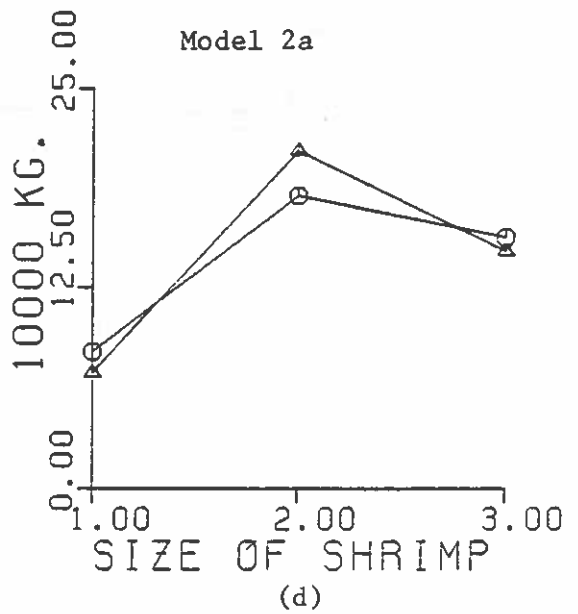
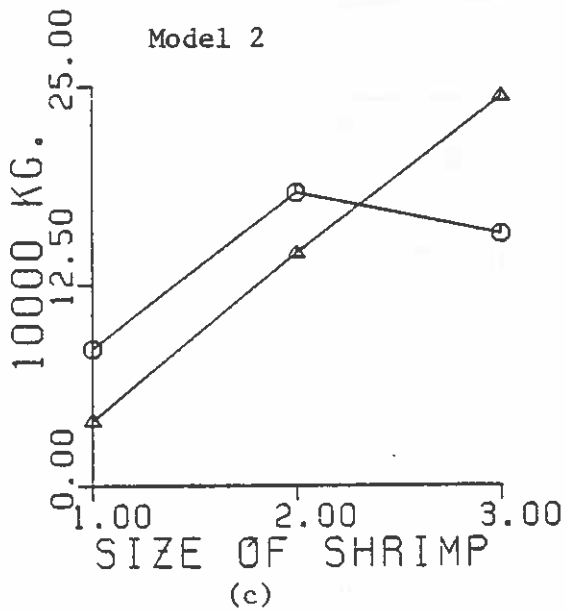
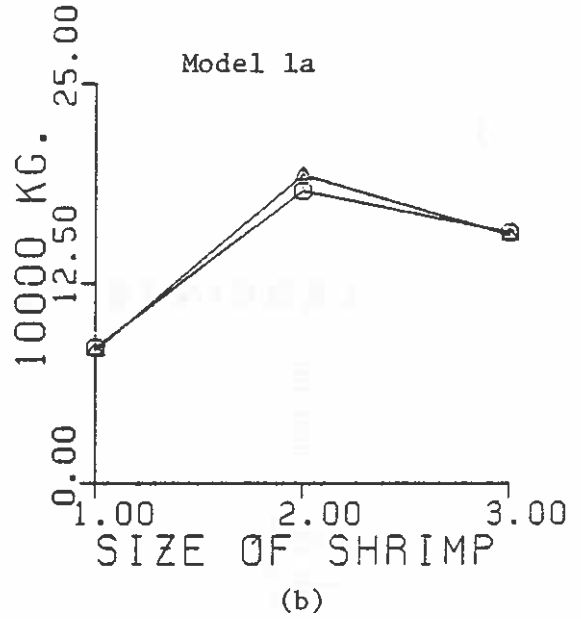
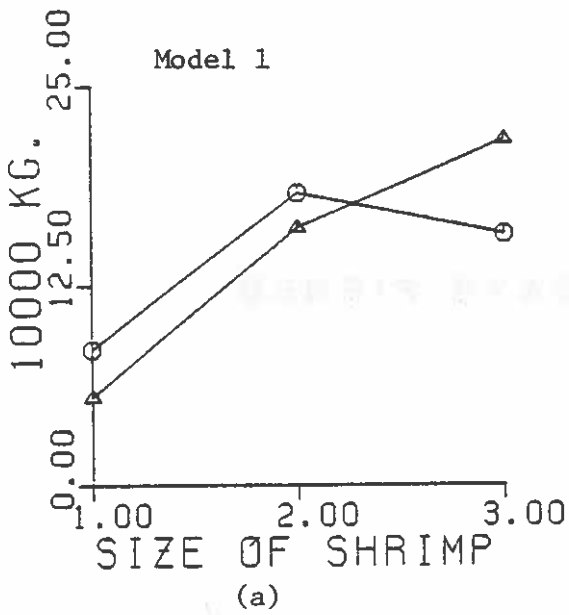


Figure 5. Actual annual landing offshore (O) compared to GBFSM estimated landing offshore (Δ).

Source: Griffin and Grant, 1982.

shore agreed closely with actual landings data (Figures 5b and 5d). In the Ivory Coast study, four models were then used (Models 1 and 2 before and after adjustment of GCOF and MSZC) in the analysis. The rationale for conducting analyses based on all four models was the lack of a satisfactory basis for resolving the discrepancy in growth rates.

#### MODEL VALIDATION

In general, model validation is a series of attempts to refute, or invalidate, a model. A model may be invalidated by: (1) refuting mechanisms or parameter values as inappropriately representing the real system, and (2) determining that the model inadequately predicts the real system's behavior. In both cases, appropriateness must be judged relative to the model's objectives.

With regard to the first type of invalidation in the Ivory Coast study, it is believed that the best information available was used to develop the mechanisms and to parameterize the models. When specific parameters could not be estimated, alternate models were developed. It was, thus, believed that the models are not easily refuted based on examination of mechanisms and parameter values.

With regard to the second type of invalidation, predicting the behavior of the real system, several comparisons can be made between predicted results, other independently constructed models, and historical data. In the Ivory Coast study (1) model predictions of total annual landings offshore at a wide range of effort levels were compared with analogous predictions of an independently constructed model (Garcia, 1978), and (2) model predictions of the magnitude and seasonal distribution of inshore and offshore

landings were compared with actual landings data.

In the Ivory Coast study, a comparison of predictions of Model 1 and Model 2 with analogous predictions of Garcia's (1978) model for total annual landings offshore at various effort levels is presented in Figure 6, which is modified from Garcia's Figure 6. Actual catch and effort levels in the fishery for the period 1969 through 1978 also are presented. The yield curves generated by both Model 1 and Model 2 have the same general shape as Garcia's curves at average effort levels or below. However, at effort levels greater than 2,230 standardized days, both of Garcia's curves increase more slowly than the Model 1 and Model 2 curves and Garcia's curve A actually indicates a decrease in yield at effort levels greater than roughly 3,500 standardized days.

In the Ivory Coast study, the magnitude and seasonal distribution of inshore and offshore landings, predicted by the four models are compared to those of actual landings data from the fishery and those for Model 1 in Figure 7. All four models were good predictors of the magnitude and general seasonal distribution of both offshore and inshore landings, although the models predicted increased inshore landings in June, that did not actually occur.

#### ESTIMATING CURVES WITH GBFSM

Yield, revenue and cost curves can be estimated with GBFSM. The adjustment factors for effort variables YR(20) and TYR(ITER, NA, NSP, NVC), along with ITER (the number of iterations per simulation run) are used to obtain values for these curves. Option A(19) must be true to develop these curves and Options A(26) and/or A(27) must be true if the number of vessels

## SIMULATION VS ACTUAL

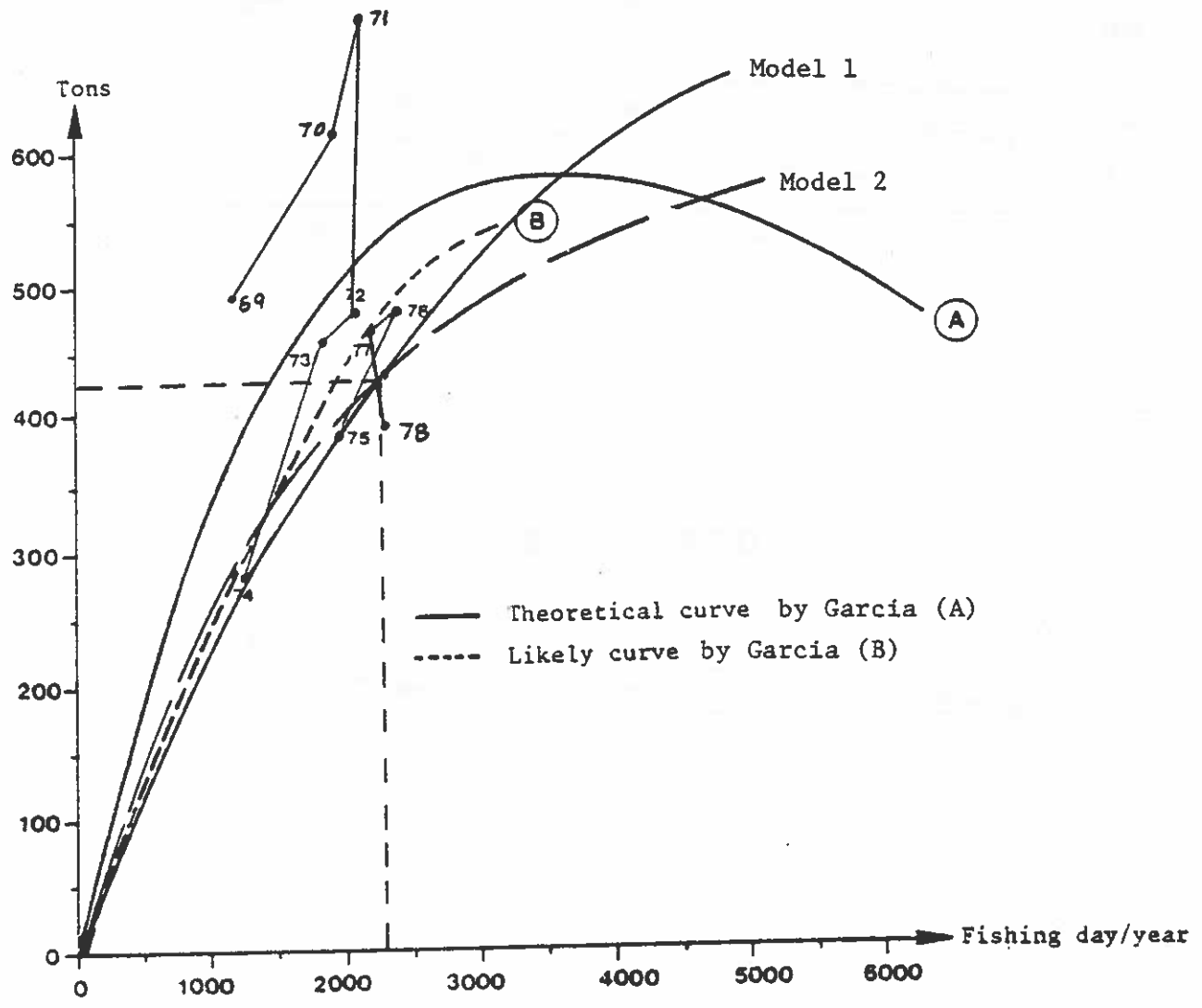


Figure 6. Comparison of GBFSM simulated curves versus Garcia's estimates and annual actual data.

## SEASONAL DISTRIBUTION

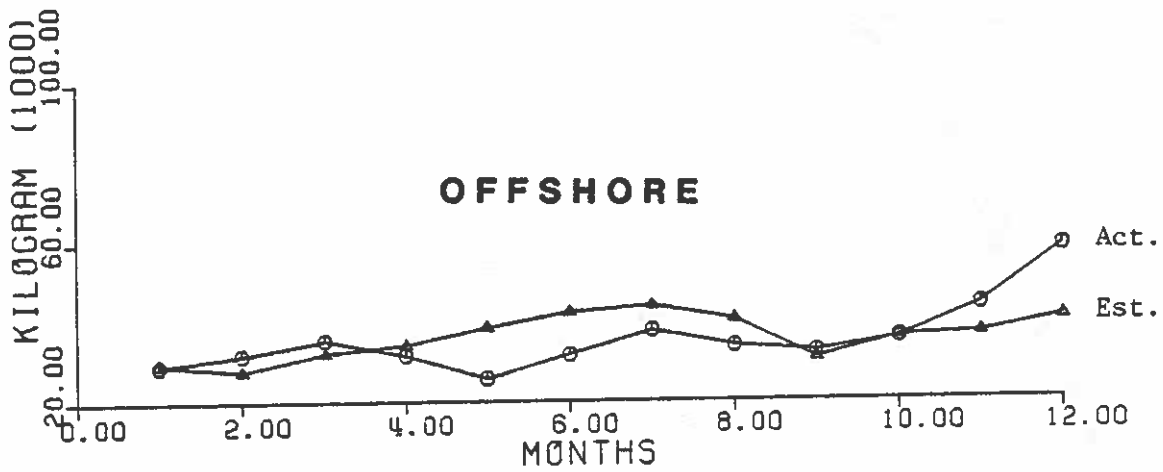
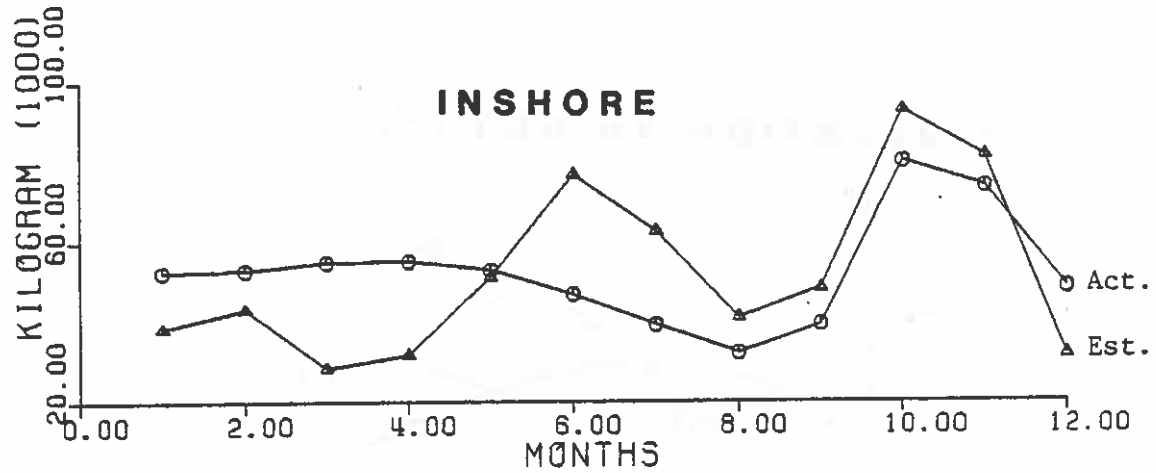


Figure 7. Comparison of GBFSM simulated data to actual data.

Source: Griffin and Grant, 1982

is changed.

The amount of effort in the model is defined by values in DFN, and YR and/or TYR let the user adjust that given level by a given percentage as defined by YR or TYR. For example, suppose curves over the range of effort from zero to 100 percent of the current effort level is desired at increments of 10 percent. First, the user must decide whether or not to vary effort to the same extent for all species and vessel classes. If so, YR can be used. If the user varies effort selectively by species and/or vessel class, then TYR is used. If YR is used, no more than 100 iterations can be made without redimensioning YR, and if TYR is used, no more than NC (number of cohorts) iterations can be made. Only the values of every other iteration is used to generate the desired curves if the species is an annual crop because in the first iteration there are no old fish GBFSM. The first iteration introduces new fish into GBFSM, and they are carried over to the second iteration as old fish. So the second iteration has both old and new fish and has only valid simulated values. When effort is incremented by 10 percent, the ODD-numbered iterations are used to adjust the old fish in GBFSM, and the EVEN-numbered iterations have the valid simulated results used in the yield, revenue and cost curves. If YR is the adjustment factor used, then:

YR(1) = 0.1	YR(7) = 0.4
YR(2) = 0.1	YR(8) = 0.4
YR(3) = 0.2	YR(9) = 0.5
YR(4) = 0.2	YR(10) = 0.5
YR(5) = 0.3	YR(11) = 0.6
YR(6) = 0.3	YR(12) = 0.6

YR(13) = 0.7

YR(17) = 0.9

YR(14) = 0.7

YR(18) = 0.9

YR(15) = 0.8

YR(19) = 1.0

YR(16) = 0.8

YR(20) = 1.0

and ITER = 20.

If the life span of a species is two years, then only every third iteration would have valid simulated results.

If the curves generated are meant to hold the number of vessels constant and change the amount of effort by changing the effort per vessel, then  $A(26) = F$  (or if TYR is used  $A(27) = F$ ). If the curves generated are meant to change effort by changing number of vessels in the fishery, then  $A(26) = T$  (or if TYR is used  $A(27) = T$ ). In either case, to adjust effort  $A(19) = T$ .

#### EXAMPLES OF POLICY ANALYSIS

GBFSM can be manipulated easily for analysis of different management strategies. Three management alternatives analyzed in other studies and their present basic results, are discussed below. Variables that must be changed in GBFSM are explained.

##### Example 1: Change Mesh Size Regulations

GBFSM was used to analyze the Ivory Coast shrimp fishery to evaluate three hypothetical fisheries management plans. One of those plans restricted the mesh size of net used in the inshore fishery. This involves changing CMESH(NSP, ND).

Notice that in row 44 (Appendix 3, SIMULAT2),  $CMESH(1) = 20.03$  mm carapace length for depth 1 (inshore), and  $CMESH(2) = 27.67$  mm carapace



length for depth 2 (offshore). If a mesh size restriction is imposed on the inshore fishery such that the smallest shrimp harvested is 23.11 mm carapace length as opposed to 20.03 mm used in the original model, Model 1 predicts a decrease in rent inshore from \$640,000 to \$502,000 and an increase in rent offshore from \$322,000 to \$943,000 (Figure 8). These changes in rent result from a decrease in inshore landings of 158,000 kg and an increase in offshore landings of 100,000 kg.

#### Example 2: Seasonal Closures

The shrimp management plan suggests closing the Fishery Conservative Zone (FCZ) off Texas in coordination with the closure of Texas waters for as long as 60 days during May, June and July. The analysis will examine the maximum duration of the closure (60 days). The values that must be entered for this closure are: GBFSM closes (ADJ = 0.0) the Texas (KAR = 1) offshore (KDT = 2) for brown shrimp (KSP = 1) during the first week (KBP = 1) of June (KBD = 6) through the last week (KEP = 5) of July (KED = 7).

Results for closing the offshore during the 60 day period of June and July are plotted in Figures 9-12. Yield curves for boats and vessels are shown in Figures 9 and 10, and cost and revenue curves are shown in Figures 11 and 12. In Figures 9 and 10, Point 1 represents the baseline equilibrium situation, Point 2 represents the situation one year after the policy change, and Point 3 represents the new equilibrium situation after the policy change.

The TFC curves in Figures 11 and 12 represent a long run total factor cost, which includes variable cost, fixed cost, and opportunity cost of labor and capital. A movement along the curve occurs by changing the num-

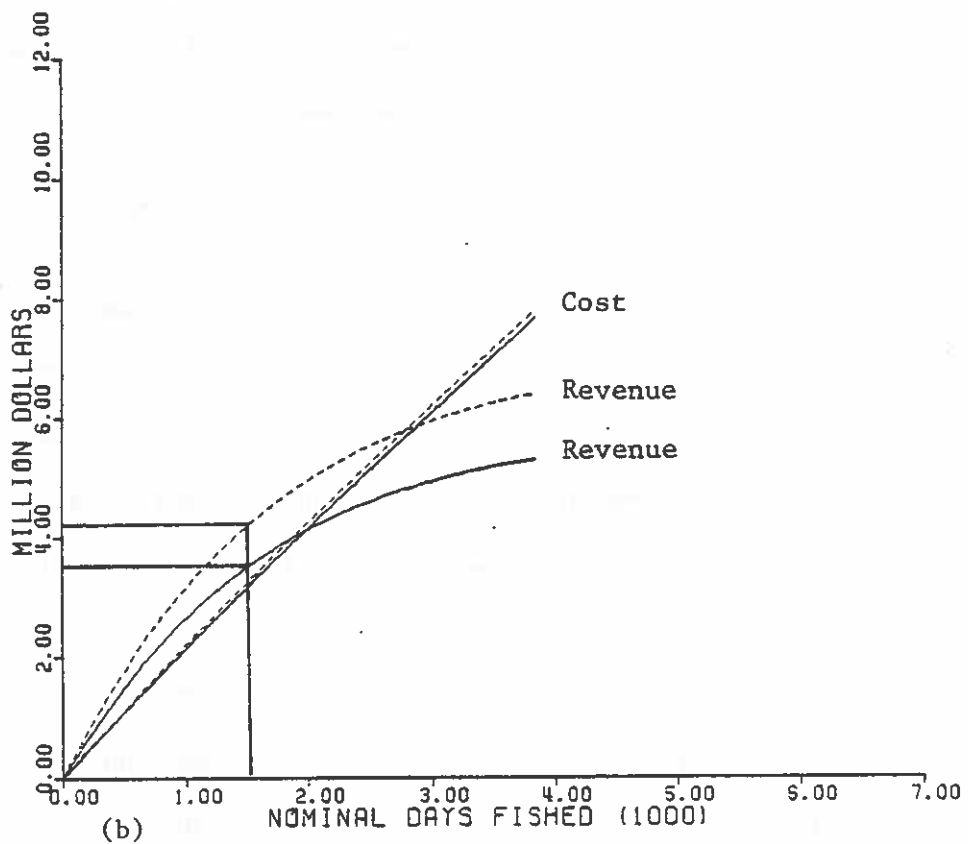
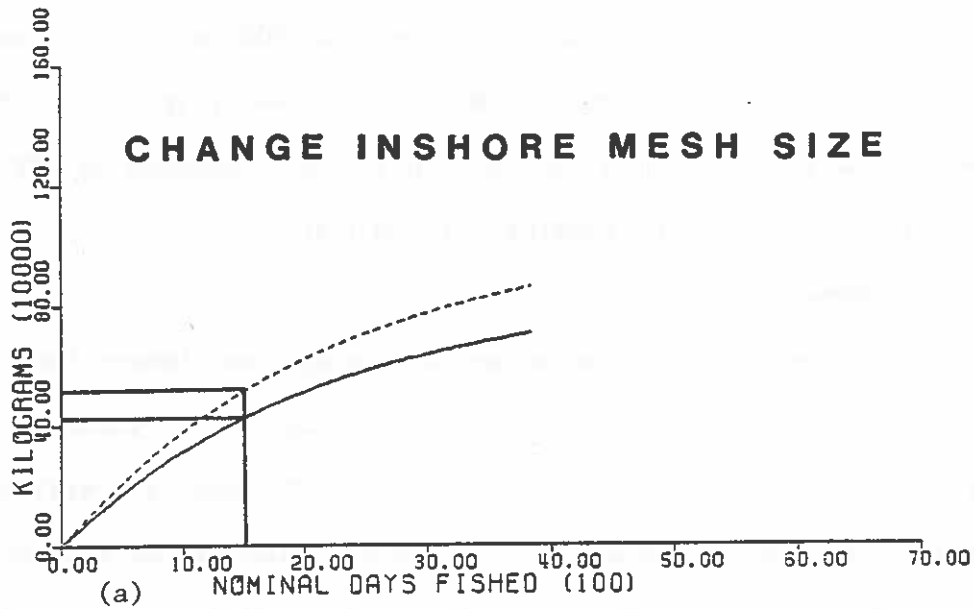


Figure 8. Comparison of annual landings offshore (a) and revenue and cost (b) for Model 1 for CMESH = 20.03 (—) and CMESH = 23.11 (---) in inshore harvesting.

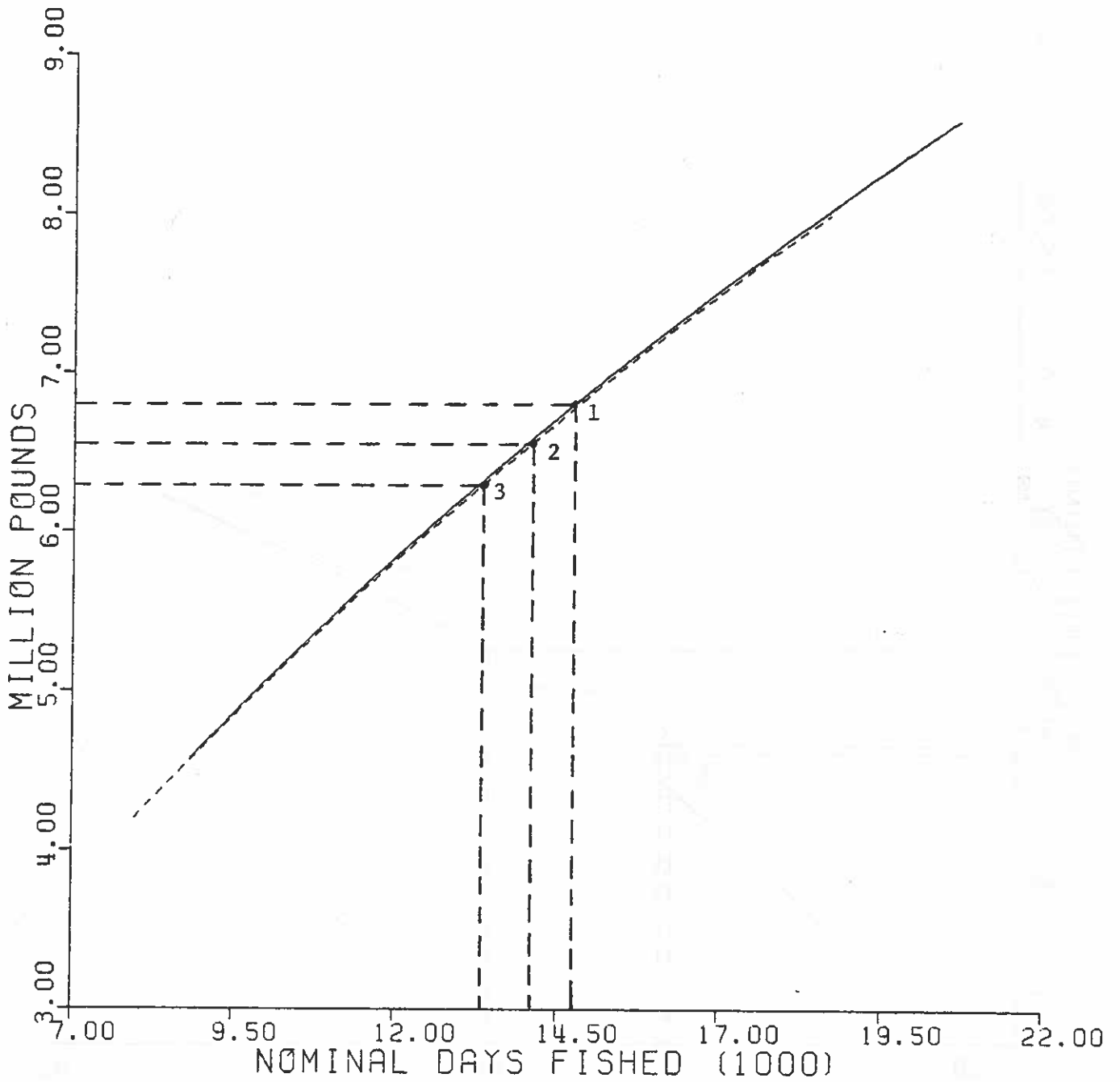


Figure 9. Yield curves of boats for baseline (solid line) and for closing the Texas offshore June and July (dashed lines).

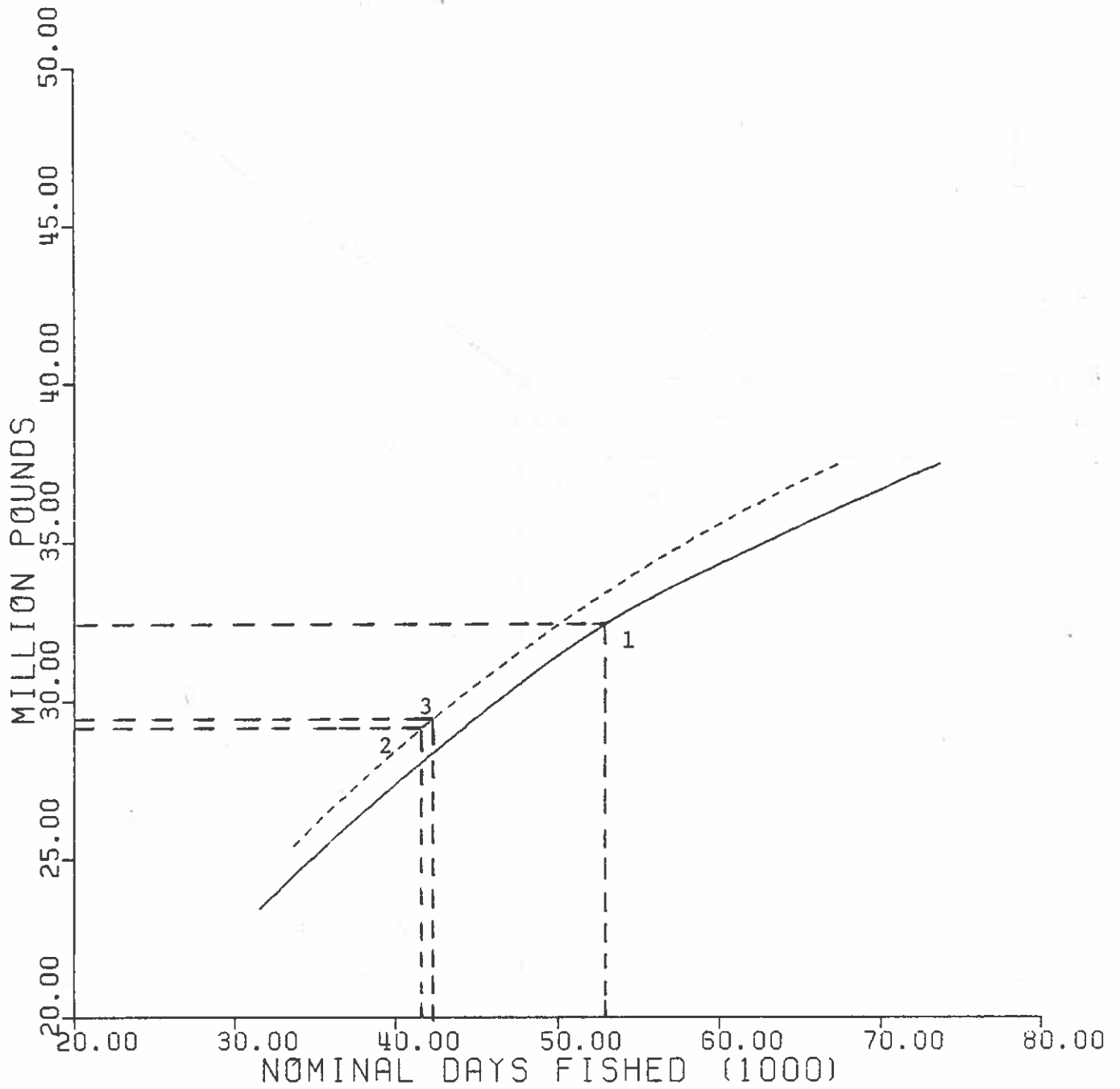


Figure 10. Yield curves of vessels for baseline (solid line) and for closing the Texas offshore June and July (dashed lines).

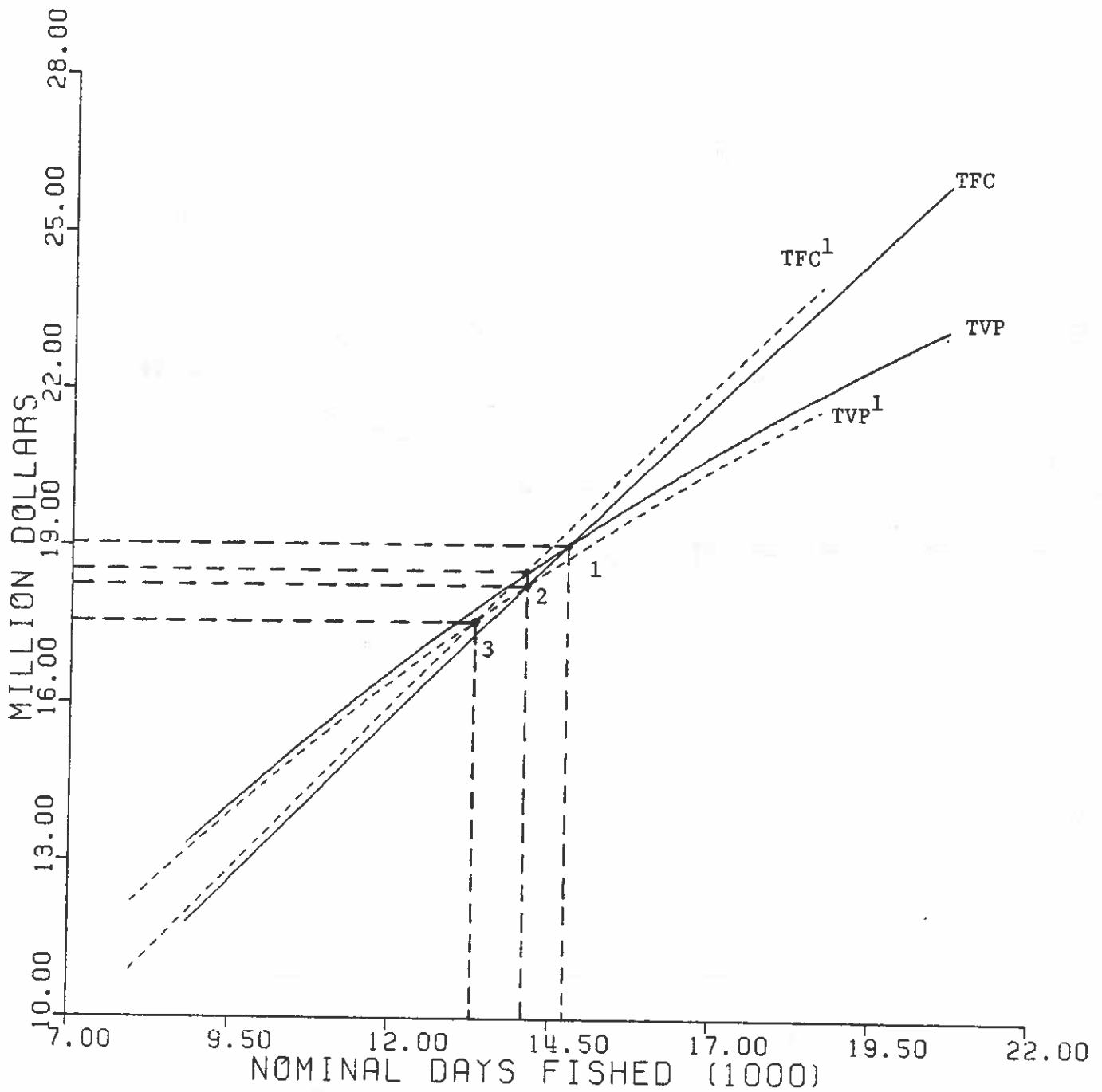


Figure 11. Total value product and total factor cost curves of boats for baseline (solid lines) and for closing the Texas offshore June and July (dashed lines).

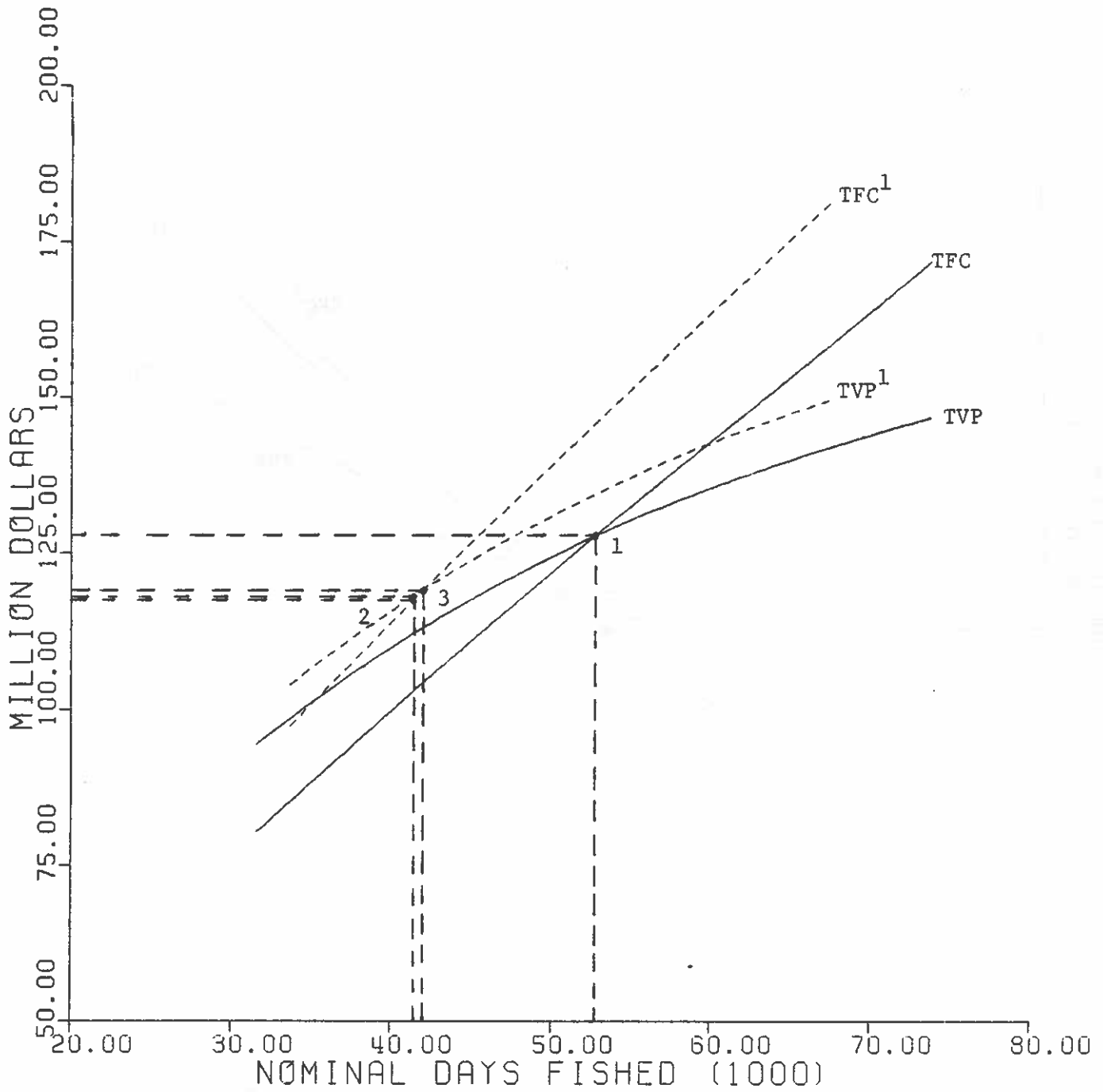


Figure 12. Total value product and total factor cost curves of vessels for baseline (solid lines) and for closing the Texas offshore June and July (dashed lines).

ber of craft and holding constant the days fished per craft in the fleet. Total value of the product curves (TVP) represent the dollar value of shrimp produced. As before, Point 1 represents the baseline equilibrium situation, the point where no excess profits are made (TVP = TFC). Point 2 represents the situation one year after the policy change. There is usually a Point 2 on the dashed TFC<sup>1</sup> curve and a Point 2 on the dashed TVP<sup>1</sup> curve because the industry is not at equilibrium, and either excess profits are being captured, or losses are being incurred. From Point 2, the industry will begin to move toward Point 3 along the dashed lines toward the new equilibrium. At Point 3, excess profits are again zero.

During the first year, closing the offshore reduces days fished for boats and vessels by 488 and 11,351, respectively, while holding the number of boats and vessels constant. This is a reduction of 3.3 percent for boats and 21.6 percent for vessels. The net effect for the first year was to reduce landings of boats by 0.18 million pounds and vessels by 3.28 million pounds, thus ultimately reducing available supplies for the first year by 3.46 million pounds.

Figures 11 and 12 show the disequilibrium position as the Point 2's for boats and vessels, respectively. The industry then adjusts toward the new equilibrium position, Point 3. Boats will move out of the industry, reducing days fished because they are incurring losses (Figure 3). Vessels will move into the industry, increasing days fished because they are making excess profits (Figure 12). Once the new equilibrium is reached, 33 full-time-equivalent boats will have left the industry and 22 full-time-equivalent vessels will have entered the industry. Excess profits will then be zero for owners and crews. There are 3.31 million pounds less

shrimp available for consumption annually, and shrimp culling is reduced by one-third.

Example 3: Eliminating Size Restriction

In the analysis of the Texas closure, one option eliminated the size restriction on shrimp. Because initially 100 percent of the undersized shrimp that are caught died, it is wasteful to discard them. The variable VKIL(NSP, ND, NM) is adjusted to eliminate culls.

VKIL(NSP, ND, NM) is the lower size limit for the smallest size-class landed, by species, depth and time division. In Appendix E, SIMULAT1, rows 55 through 60 show the values for two species, three depths and 12 months. CMESH (row 46) is 45.0 mm in length at depths and 2 for species 1. Because rows 55 and 57 have values greater than 45.0 mm in length, culls will be generated. However, vessel class 1 (bayboats) harvest 95 percent of culls; vessel class 2 (Gulf vessels) harvest 50 percent of the culls (Row 48), and these are reported in the smallest size class of legally landed shrimp. In months 8, 9 and 10, species 1 (brown shrimp) must be at least 67.6 mm in length to be landed from depth 1. Eliminating the size-count restriction on all species of shrimp at all depths in all time divisions means that the values of VKIL(NSP, ND, NM) must be changed to values less than or equal to CMESH(NSP, ND). Results are shown in Figures 13 and 14. Point 1 is the equilibrium condition before the policy change, Point 2 is the first-year impact from policy change, and Point 3 is the new equilibrium after all adjustments have taken place.

During the first year under this management alternative, the number of days fished remains constant. Landings increase the first year by the same amount that culls decrease. Vessel landings increase by 6.58 million pounds



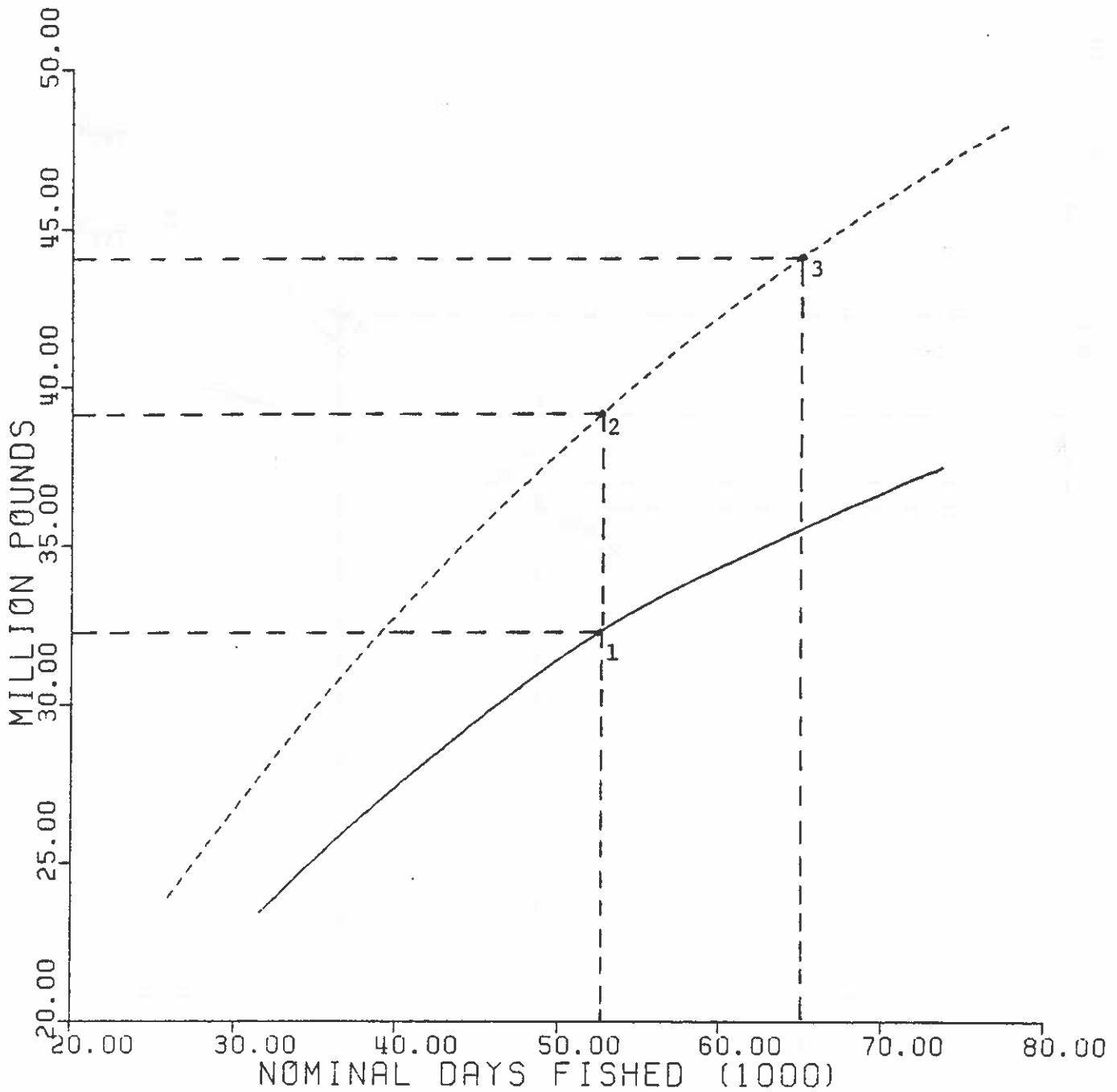


Figure 13. Yield curves of vessels for baseline (solid line) and for eliminating Texas Count Law (dashed lines).

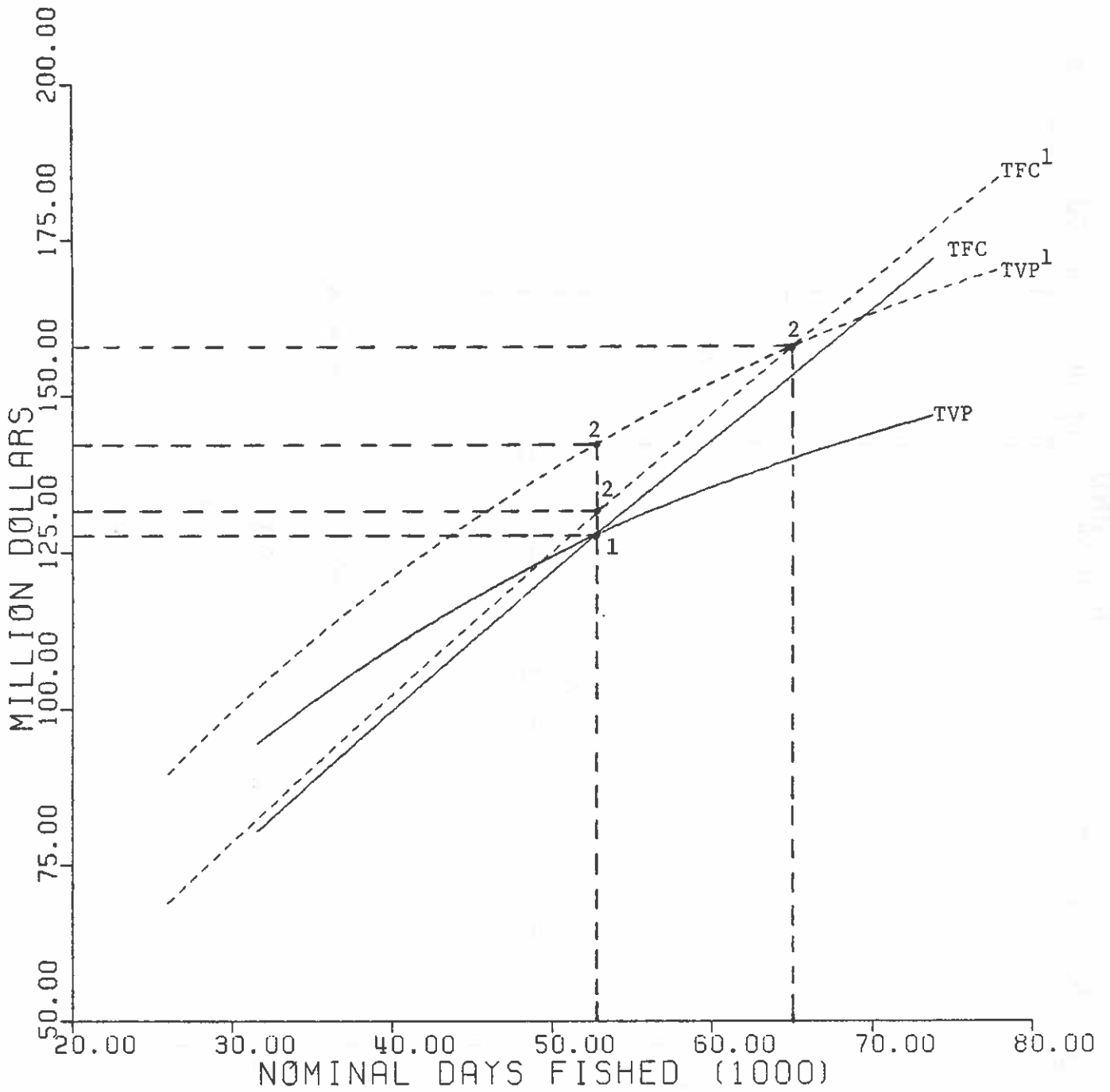


Figure 14. Total value product and total factor cost curves of vessels for baseline (solid lines) and for a Texas No Count Law (dashed lines).

(20 percent) and consumers now enjoy 6.6 million more pounds of shrimp (Figure 13). Excess profits for the first year are \$9.78 million for vessel owners and \$2.44 million for vessel crews.

To reach the new equilibrium, (Point 3 from Point 2 in Figure 14), 273 full-time-equivalent vessels will move into the industry such that fishing for shrimp increases by 12,311 days. Nine boats leave the industry because of the added fishing pressure by offshore vessels. Vessels land an additional 11.99 million pounds annually, and available supplies increase by 11.88 million pounds.

The above three examples illustrate how the models can be used to evaluate management alternatives. In addition to CMESH, Closure Parameter and VKIL, the following variables can also be used: CF, to change percentage of cull that can be landed legally; FLR and HP, to evaluate size restriction on vessels; FC, to impose a tax on vessels; COST to impose a tax on nominal days fished; PC to impose a tax on landing; TYR, to vary the number of vessels and/or the number of nominal days fished per vessel (Limited Entry); and DFN, to directly vary the number of nominal days fished. Finally, any combination of the above variables may be changed at the same time, depending on the management strategy to be analyzed.

## REFERENCES

1. Blomo, V., K. Stokes, W. Griffin, W. Grant and John Nichols, "Bio-economic Modeling of the Gulf Shrimp Fishery: An Application to Galveston Bay and Adjacent Offshore Areas," Southern Journal of Agricultural Economics, Vol. 10, No. 1, July 1978.
2. Blomo, Vito J., John P. Nichols, Wade L. Griffin and William E. Grant, "Dynamic Modeling of a Natural Resource Problem: Eastern Gulf of Mexico Shrimp Fishery," American Journal of Agricultural Economics, Vol. 64, No. 3, August 1982.
3. Grant, W.E., and W.L. Griffin, "A Bioeconomic Model of the Gulf of Mexico Shrimp Fishery," Transactions of American Fisheries Society 108: 1-13, 1979.
4. Grant, W.E., K.G. Isaakson and W.L. Griffin, "A General Bioeconomic Simulation Model for Annual-Crop Marine Fisheries," Ecological Modelling 13: 195-219, 1981.
5. Grant, W.E., W.L. Griffin and J.P. Warren, "A Management Model of the Northwest African Cephalopod Fishery," Marine Fisheries Review, November 1981.
6. Griffin, W.L. and W.E. Grant, 1981. "A Bioeconomic Analysis of the Ivory Coast Shrimp Fishery," Presented at the International Shrimp Workshop, Key West, Florida, November 1981.
7. Griffin, W.L., J.P. Warren and W.E. Grant, 1979. A Bio-Economic Model for Fish Stock Management: The Cephalopod Fishery of Northwest Africa. CECAF/TECH/79/16 (En). United Nations Development Programme, Food and Agricultural Organization of the United Nations.
8. Griffin, W.L., J.P. Nichols, W.E. Grant and J.P. Warren, 1981. Analysis of Management Alternatives for the Texas Shrimp Fishery, DIR 81-1, Staff Paper No. 1, Department of Agricultural Economics, Texas Agricultural Experiment Station, Texas A&M University, February.
9. Griffin, W.L. and J.R. Stoll, 1981, "Economic Issues Pertaining to the Gulf of Mexico Shrimp Management Plan," In Economic Analysis in Fisheries Management Plans, Lee G. Anderson (ed.), Ann Arbor Science Publishers, Inc.
10. Isaakson, K.G., W.E. Grant and W.L. Griffin, 1982. "General Bioeconomic Fisheries Simulation Model: A Detailed Model Documentation," Journal of the International Society of Ecological Modelling.
11. Warren, John P., Wade L. Griffin and William E. Grant, "Regional Fish Stock Management: A Model for Northwest Africa," Marine Policy, April, 1982.

Appendix A - TABLE

TABLE

Section 1. Model Parameter for Driver Program

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
NSP	Number of Species	I4	1-4
ND	Number of Depths. Maximum number is 10	I4	5-8
NC	Number of cohorts	I4	9-12
NA	Number of geographical areas	I4	13-16
NVC	Number of vessel classes. Maximum number is 10.	I4	17-20
NSC	Number of fish class sizes. Maximum number is 9.	I4	21-24
NM	Number of time divisions for landing statistics. For example, weeks, months quarters.	I4	25-28
NER	Number of time divisions for the phenolog factor, E, for altering recruitment.	I4	29-32
NPH	Number time divisions for the phenolog factor, GRT, for altering growth	I4	33-36
NBC	Number of species by-catch of fish. (Maximum = 8)	I4	37-40
NSX	Number of sexes of fish (Maximum = 2)	I4	41-44

Section 2. Options Available

Note: "T" means activate option; "F" means ignore option.

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
Q	Section description card for Options Available	20A4	1-80
A (1)	Print biological diagnostics for last run period	L1	4
A (2)	Print biological diagnostic after every run period. Must be used with A(1).	L1	8

Section 2. Options Available (Cont'd.)

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
A (3)	Print landing statistic after every run period by area and species. (Refer to A(3) thru A(8) and A(21) thru A(23)).	L1	12
A (4)	Print landing output by statistical collection division area and species (with bait and recreational statistics)	L1	16
A (5)	Print landings output by statistical collection division, depth, area and species	L1	20
A (6)	Print landings output by statistical collection division, depth, size class, area and species	L1	24
A (7)	Print landings by statistical collection division, depth, size class, vessel class, area and species	L1	28
A (8)	Calculate economic and print revenue variable cost, packing cost, crew shares and rent by collection division, area and species	L1	32
A (9)	Print fishing mortality diagnostic by time-step, depth area and species.	L1	36
A (10)	Printout input data	L1	40
A (11)	Uniform Distribution	L1	44
A (12)	Triangular Distribution	L1	48
A (13)	Normal Distribution	L1	52
A (14)	Stochastisizes ICOF	L1	56
A (15)	Stochastisizes ER	L1	60
A (16)	Stochastisizes NMCOF	L1	64
A (17)	Stochastisizes FMAX	L1	68
A (18)	Stochastisizes GRT	L1	72

Section 2. Options Available (Cont'd.)

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
A (19)	Adjusts DFN. Must be used in conjunction with variables YR and TYR.	L1	76
A (20)	Produces matrix of stochastic runs	L1	80
A (21)	Calculates additional economic data and print summarizes of it by vessel class by species	L1	4
A (22)	Prints summarized economic data by vessel class (Requires A(21) to be true)	L1	8
A (23)	Prints summarized economic data by species. (Requires A(21) to be true)	L1	12
A (24)	Call DEMAND	L1	16
A (25)	Print monthly values for A(3) thru A(8)	L1	20
A (26)	Adjust number of vessels by YR. (Requires A(19) to be true)	L1	24
A (27)	Adjust number of vessels by TYR (1, NVC). (Requires A(19) to be true)	L1	28
A (28)	Calculates By-Catch of fish as function of effort	L1	32
A (29)	Calculates By-Catch of fish as function of target species.	L1	36
A (30)	Print growth of individual shrimp by time-step, species, sex, cohort	L1	40
A (31)	Print price by time and size class	L1	44

Section 3. Time Parameters

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
Q	Section description card for Time Parameters	20A4	1-80
Q	Variable description card for Variables below	20A4	1-80



Section 3. Time Parameters (Con'td.)

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
IC	Number of Closures. If value is one or greater then must include third data card in Section 11.	I6	1-6
ITER	Number of periods per job run. (Note: ITER must be less than or equal to NC)	I6	7-12
FTIME	Number of time steps per period	I6	13-18
CCT	Number of time steps per cohort	F6.0	19-24
CN	Timing coefficient for NM	F6.0	25-30
CER	Timing coefficient for NER	F6.0	31-36
CPH	Timing coefficient for NPH	F6.0	37-42
C8	Minimum number for a cohort	F6.0	43-48

Section 4. Recruitment and Movement Coefficients

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
Q	Section description card for Recruitment and Movement Coefficients	20A4	1-80
Q	Variable description card for ICOF (NSP,NA)	20A4	1-80
ICOF (NSP,NA)	Gross recruitment rate by species and area.	8E10.4	1-10...
Q	Variable description card for E (NSP,NER)	20A4	1-80
E (NSP,NER)	Rate altering recruitment by species and time division for phenology factor E.	12F6.0	1-6...
Q	Variable description card for ISZ (NSP, NSX)	20A4	1-80
ISZ (NSP,NSX)	Initial size for new recruitments by species and by sex. (length in mm)	12F6.0	1-6...
Q	Variable description card for ER (NSP, ND,NA,NA)	20A4	1-80
ER (NSP,ND,NA,NA)	Movement rates (depth moved from area moved <u>from</u> . area moved <u>to</u>	12F6.0	1-6...

#### Section 4. Recruitment and Movement Coefficients

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
Q	Variable description card for SZCOF (NSP, NSX)	20A4	1-80
SZCOF (NSP,NSX)	Minimum size that shrimp begin moving from depth 1 to depth 2.	12F6.0	1-6...

#### Section 5. Growth Coefficients

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
Q	Section description card for Growth Coefficients	20A4	1-80
Q	Variable description card for GCOF (NSP,NSX)	20A4	1-80
GCOF (NSP,NSX)	Growth coefficient (growth = GCOF (MSZC - current size)	10F8.0	1-8...
Q	Variable description card for MSZC (NSP,NSX)	20A4	1-80
MSZC (NSP,NSX)	Maximum size for growth equation	10F8.0	1-8...
Q	Variable description card for GRT (NSP,NPH)	20A4	1-80
GRT (NSP,NPH)	Rate altering growth by species and time divisions for phenology factor GRT.	12F6.0	1-6...
Q	Variable description card for C3 (NSP)	20A4	1-80
C3 (NSP)	Linear coefficient in length to weight equation ( $W = C3 (\text{length})C4$ )	10F8.0	1-8
Q	Variable description card for C4 (NSP)	20A4	1-80
C4 (NSP)	Power coefficient in length to weight equation	12F6.0	1-6

#### Section 6. Natural Mortality Coefficients

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
Q	Section and variable description card for Natural Mortality Coefficients	20A4	1-80
NMCOF (NSP,ND,NSC)	Natural Mortality rates by species, depth and size class. (Read largest to smallest for NSC)	12F6.0	1-6...

Section 7. Harvest Coefficients

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
Q	Section description card for Harvest Coefficients	20A4	1-80
Q	Variable description card for Variables below.	20A4	1-80
BMESH	Bait harvest lower catchable size	F6.0	1-6
RMESH	Recreation harvest lower catchable size	F6.0	7-12
BC	Bait fishing mortality coefficient	F6.0	13-18
RC	Recreation fishing mortality coefficient	F6.0	19-24
ETM	Conversion ratio for units in landings (Example: grams to pounds)	F6.0	25-30
C5	Upper size bound for altered fishing mortality range	F6.0	31-36
C6	Lower size bound for altered fishing mortality range	F6.0	37-42
C7	Fraction killed between C5 and C6	F6.0	43-50
Q	Variable description card for CMESH (NSP,ND)	20A4	1-80
CMESH (NSP,ND)	Commercial harvest lower catchable size by species and depth.	12F6.0	1-6...
Q	Variable description card for CF (NVC, ND)	20A4	1-80
CF (NVC,ND)	Percent culls counted in legal catch by vessel class and depth.	12F6.0	1-6...
Q	Variable description card for SL (NSP,NSC)	20A4	1-80
SL (NSP,NSC)	Boundaries between size classes (Read largest to smallest NSC left to right)	12F6.0	1-6...
Q	Variable description card for FMAX (NSP,ND,NA)	20A4	1-80
FMAX (NSP,ND,NA)	Density factor for fishing mortality	10F8.0	1-8...

Section 7. Harvest Coefficients (Cont'd.)

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
Q	Variable description card for VKIL (NSP,ND,NM)	20A4	1-80
VKIL (NSP,ND,NM)	Variable lower size limit for smallest size class landed by species, depth and time division	12F6.0	1-6...
Q	Variable description card for PERF (NBC,NVC,NM)	20A4	1-80
PERF (NA,ND,NBC, NVC,NM)	Factor determining by-catch	12F6.0	1-6...

Section 8. Relative Fishing Power Coefficients

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
Q	Section description card for Relative Fishing Power Coefficients	20A4	1-80
Q	Variable description card for FRL (NVC)	20A4	1-80
FRL (NVC)	Total length of footrope of fishing net by vessel class	12F6.0	1.6...
Q	Variable description card for HP (NVC)	20A4	1-80
HP (NVC)	Horsepower of vessel by vessel class	12F6.0	1-6...
Q	Variable description card for C1 (NSP)	20A4	1-80
C1 (NSP)	Power coefficient on HP in relative fishing power equation by species	12F6.0	1-6...
Q	Variable description card for C2 (NSP)	20A4	1-80
C2 (NSP)	Power coefficient on FRL in relative fishing power equation	12F6.0	1-6...

Section 9. Economic Coefficients

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
Q	Section description card for Economic Coefficients	20A4	1-80
Q	Variable description card for CONV (NSP)	20A4	1-80

Section 9. Economic Coefficients (Cont'd.)

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
CONV (NSP)	Conversion from hold fish to product sold at dock by species (Example Heads-on to heads-off)	12F6.0	1-6...
Q	Variable description card for NV (NA,NVC)	20A4	1-80
NV (NA,NVC)	Number of vessel in each class by area and vessel class	12F6	1-6...
Q	Variable description card for C (NVC)	20A4	1-80
C (NVC)	Crew number per vessel by vessel class	12F6.0	1-6...
Q	Variable description card for FC (NVC)	20A4	1-80
FC (NVC)	Fixed cost per period (i.e. one year) per vessel by vessel class.	12F6.0	1-6...
Q	Variable description card for CVC (NVC)	20A4	1-80
Q	Variable description card for COST (NVC,ND)	20A4	1-80
COST (NVC,ND)	Vessel owner's variable cost for vessels per nominal day fished by vessel class and depth.	12F6.0	1-6...
CVC (NVC)	Crew's variable cost per nominal days fished per vessel by vessel class	12F6.0	1-6...
Q	Variable description card for OCO (NVC)	20A4	1-80
OCO (NVC)	Owner's opportunity cost per period (i.e. one year) per vessel by vessel class	12F6.0	1-6...
Q	Variable description card for OCC (NVC)	20A4	1-80
OCC (NVC)	Crew's opportunity cost per period (i.e. one year) per vessel per crew member by vessel class	12F6.0	1-6...
Q	Variable description card for PC (NVC)	20A4	1-80
PC (NVC)	Packing charge per pound of fish landed by vessel class	12F6.0	1-6...
Q	Variable description card for SHARE (NVC)	20A4	1-80
SHARE (NVC)	Crew's percent share of landings by vessel class (Enter 20% as 0.2)	12F6.0	1-6...

Section 9. Economic Coefficients (Cont'd.)

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
Q	Variable description card for PMCSP (NSP,NSC,NM)	20A4	1-80
PMCSP (NSP,NSC, NM)	Ex-vessel price of fish by species size class and division (Read largest to smallest for NSC)	12fç.0	1-6...
Q	Variable description card for PCATCH (NBC)	20A4	1-80
PCATCH (NBC)	Unit price received for by catch by species	20F6.0	1-6...

Section 10. Adjustment Factors for Effort

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
Q	Section description card for Adjustment Factors for Effort	20A4	1-80
Q	Variable description card for YR (20)	20A4	1-80
YR (20)	Coefficient for gross effort adjustment by period	20F4.0	1-4...
Q	Variable description card for TYR (ITER,NSP,NVC)	20A4	1-80
TYR (ITER,NA,NSP, NVC)	Coefficient for effort adjustment by area, species and vessel class. (Note ITER may not exceed the value of NC)	20F4.0	1-4...
Q	Variable description card for PRNT(ITER)	20A4	1-80
PRNT (ITER)	Enter a "1" in the appropriate column for each interation. Example: to print only rows 1, 5 and 8 enter a "1" in columns 1, 5 and 8. This overrides option A(3). A zero (0) in any column allows Option A(3) to work for that interation	80F1.0	1...

Section 11. Closure Parameters

Note: One data card required for each closure. Maximum of 18 closures.

Section 11. Closure Parameters (Cont'd.)

<u>Variable name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
Q	Section description card for Closure Parameters	20A4	1-80
Q	Variable description card for Variables below	20A4	1-80
KBD	Beginning division closure (i.e., first month this closure occurs)	I5	1-5
KED	Ending division closure (i.e., last month this closure occurs)	I5	6-10
KBP	Beginning time step within beginning division closure occurs (i.e., first week within first month this closure occurs)	I5	11-15
KEP	Ending time step within ending division closure occurs (i.e., last week within last month this closure occurs)	I5	16-20
KAR	Area to be closed	I5	21-25
KDT	Depth to be closed	I5	26-30
KSP	Species to be closed	I5	31-35
ADJ	Adjust factor for closure	F5.2	36-40

Section 12. Labels

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
Q	Section and variable description card for Labels	20A4	1-80
COL (NM)	Labels of time divisions of landing statistics	20A4	1-4...

Section 13. Stochastic Parameters

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
Q	Section description card for Stochastic Parameters	20A4	1-80
Q	Variable description card for IX(I), I = 1,5	20A4	1-80
IX(I), I = 1,5	Randon number seed corresponding to A(16) thru A(20)	5I6	1-6...

Section 13. Stochastic Parameters (Cont'd.)

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
Q	Variable description card for SP(I), I = 1,10	20A4	1-80
SP(I), I = 1,10	Statistical parameter corresponding to A(16) thru A(20)	10F6.0	1-6...

Section 14. Nominal Days Fished

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
Q	Section and variable description card for Nominal Days Fished	20A4	1-80
DFN (NSP,NA,ND NVC,NM)	Nominal days fished by species, area depth, number of vessel class and time division.	12F6.0	1-6...

Section 15. Demand Equation (optional)

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>	<u>Col</u>
Data Card 1:			
SIZ (1)	First boundaries for size classes in the demand equations	F8.3	1-8
SIZ (2)	Second boundary for size class in the demand equations	F8.3	9-16
TIM (1)	First boundary between seasons for demand equations	I4	17-20
TIM (2)	Second boundary between season for demand equations	I4	21-24
TIM (3)	Third boundary between season for demand equations.	I4	25-28
Data Card 2:			
DCOF (4,3,4)	Coefficients for demand equations	5E14.5	1-14...



Appendix B - SBUILD

```

1.          IMPLICIT INTEGER (A-Z)
2.          C
3.          C -- READ IN MODEL PARAMETERS (ARRAY DIMENSIONS)
4.          C
5.          READ(5,10,END=99) NSP,ND,NC,NA,NVC,NSC,NM,NER,NPH,NBC,NSX
6.          10 FORMAT(11I4)
7.          C
8.          C -- PRODUCE BLOCK DATA SUBPROGRAM
9.          C
10.         WRITE(7,1000)
11.         WRITE(7,1010)
12.         WRITE(7,1020)
13.         WRITE(7,1030)
14.         WRITE(7,1500)
15.         WRITE(7,1040)
16.         WRITE(7,1050)
17.         WRITE(7,1060)
18.         WRITE(7,1520) NSP,ND,NVC,NA,NC
19.         WRITE(7,1530) NSC,NM,NBC,NSX
20.         WRITE(7,1070)
21.         1000 FORMAT(6X,'BLOCK DATA')
22.         1010 FORMAT(6X,'COMMON/COLSE/ KBD(18),KED(18),KBP(18),KEP(18),KAR(18),
23.         + /5X,'+',T21,'KDT(18),KSP(18)')
24.         1020 FORMAT(6X,'COMMON/ADJUST/ ADJ(18)')
25.         1030 FORMAT(6X,'COMMON/WEEK/ KK(4,18)')
26.         1040 FORMAT(6X,'DATA KBD/18*O/,KED/18*O/,KBP/18*O/,KEP/18*O/,KAR/18*O/'
27.         + /5X,'+',T12,'KDT/18*O/,KSP/18*O/')
28.         1050 FORMAT(6X,'DATA ADJ/18*O.O/')
29.         1060 FORMAT(6X,'DATA KK/72*O/')
30.         1500 FORMAT(6X,'COMMON/PARAM/TIME,NSP,ND,NC,NA,NVC,NSC,NM,NSX,NBC,CN')
31.         1510 FORMAT(6X,'COMMON/SFISH/BMESH,RMESH,BC,RC,C5,C6')
32.         1520 FORMAT(6X,'DATA NSP,ND,NVC,NA,NC/',4(I2,','),I3, '/')
33.         1530 FORMAT(6X,'DATA NSC,NM,NBC,NSX/',I2,3(','),I2,
34.         + '/')
35.         1070 FORMAT(6X,'END')
36.         C
37.         C -- PRODUCE DRIVER --
38.         C
39.         WRITE(7,1010)
40.         WRITE(7,1020)
41.         WRITE(7,1030)
42.         WRITE(7,1500)
43.         WRITE(7,1510)
44.         C
45.         C -- PRODUCE DIMENSION STATEMENTS --
46.         C
47.         C
48.         C -- GENERATE 1ST REAL DIMENSIONING STATEMENT
49.         C
50.         WRITE(7,2010) NSP,NSX,NSP,NSX,NSP,NSX,NSX
51.         2010 FORMAT(6X,'REAL ISZ(',I2,',',I2,') ,SZCOF(',I2,',',I2,') ,GCOF(',
52.         1 I2,',',I2,') ,/5X,'+MSZC(',I2,',',I2,')')
53.         C
54.         C -- GENERATE 2ND REAL DIMENSIONING STATEMENT
55.         C
56.         WRITE(7,2020) NSP,NSP,NSP
57.         2020 FORMAT(6X,'REAL CONV(',I2,') ,C1(',I2,') ,C2(',I2,') ,')
58.         WRITE(7,2030) NVC,NVC,ND,NVC,NVC

```

```

59. 2030 FORMAT(5X,'1HP(',I2,')',FRL(',I2,')',TD(',I2,')',TREV(',
60. 1 I2,')',TRENT(',I2,')',)
61. WRITE(7,2040) NVC,NVC,NVC,NSP,NSC,NM,NM
62. 2040 FORMAT(5X,'+TCS(',I2,')',TOPC(',I2,')',TVC(',I2,')',SL(',I2,',',I2,
63. 1 ')',COL(',I2,')',/5X,'+TM(',I2,')',)
64. WRITE(7,2050) NSP,NER,NSP,NA,NSP,NSX,NC,NSP,NPH
65. 2050 FORMAT(5X,'3E(',I2,',',I2,')',ICOF(',I2,',',I2,')',/6X,'REAL SZ(',
66. 1 2(I2,','),I3,')',GRT(',I2,',',I2,')',)
67. WRITE(7,2060) NSP,ND,NA,NSP,ND,NA,NA,NSP,NSX,ND,NC,NA
68. 2060 FORMAT(5X,'4FMAX(',2(I2,','),I2,')',ER(',3(I2,','),I2,
69. 1 ')',/5X,'+NUM(',3(I2,','),I3,',',I2,')',)
70. WRITE(7,2070) NSP,ND,NSC,NSP,ND,NA,NVC,NSC,NM,NSP,NA,NM
71. 2070 FORMAT(5X,'5NMCDF(',2(I2,','),I2,')',CCAT(',5(I2,','),I2,
72. 1 ')',/5X,'+BCAT(',2(I2,','),I2,')',)
73. C
74. C -- GENERATE 3RD REAL DIMENSIONING STATEMENT
75. C
76. WRITE(7,3000) NSP,ND,NVC,NA,NSP,ND,NVC,NA,NSP,ND,NVC
77. 3000 FORMAT(6X,'REAL FMORT(',3(I2,','),I2,')',RFEF(',
78. 1 3(I2,','),I2,')',/5X,'+RFP(',2(I2,','),I2,')',)
79. WRITE(7,3010) NSP,ND,NA,NVC,NM,NSP,NA,ND,NVC,NM
80. 3010 FORMAT(5X,'1CULL(',4(I2,','),I2,')',/5X,'+DFN(',4(I2,','),I2,
81. 1 ')',)
82. WRITE(7,3020) NSP,NA,NM,NVC,NSP,NA,NM,NVC,NSP,NA,NM,NVC
83. 3020 FORMAT(5X,'2RENT(',3(I2,','),I2,')',/5X,'+CS(',3(I2,','),I2,
84. 1 ')',/5X,'+OPC(',3(I2,','),I2,')',)
85. C
86. C -- GENERATE 4TH REAL DIMENSIONING STATEMENT
87. C
88. WRITE(7,3030) NSP,NA,NM,NVC,NSP,NA,NM,NVC,NSP,NSC,NM
89. 3030 FORMAT(6X,'REAL VC(',3(I2,','),I2,')',/5X,'+REV(',3(I2,','),I2,
90. 1 ')',PMCS(',2(I2,','),I2,')',)
91. WRITE(7,3040) ND,NM,ND,NSC,NM
92. 3040 FORMAT(5X,'+ TDM(',I2,',',I2,')',TDSM(',2(I2,','),I2,')',)
93. WRITE(7,3050) ND,NSC,NVC,NM,ND,NSC,NM
94. 3050 FORMAT(5X,'+ TDSVM(',3(I2,','),I2,')',,
95. + 'TDSV(',2(I2,','),I2,')',)
96. WRITE(7,3060) ND,NSC,NSP,NSC,NM
97. 3060 FORMAT(5X,'+ TDS(',I2,',',I2,
98. + ')',PMCS(',2(I2,','),I2,')',)
99. C
100. C -- PRODUCE 5TH DIMENSIONING STATEMENT
101. C
102. WRITE(7,4000) NA,NSP,ND,NM,NSP,NA,NM,NA,NVC,NVC
103. 4000 FORMAT(6X,'REAL VKIL(',3(I2,','),I2,')',RCAT(',2(I2,',')
104. 1 ',I2,')',/5X,'+NV(',I2,',',I2,')',FC(',I2,')',)
105. WRITE(7,4010) NVC,NVC,NVC,NVC,NA,NSP,NVC,NA,NSP,NVC
106. 4010 FORMAT(5X,'10CD(',I2,')',OCC(',I2,')',C(',I2,')',CVC(',I2,
107. 1 ')',TCOST(',2(I2,','),I2,')',/5X,'+CCOST(',2(I2,','),I2
108. 1 ')',)
109. C
110. C -- FINISH 5TH DIMENSIONING STATEMENT AND
111. C -- PRODUCE 6TH DIMENSIONING STATEMENT
112. C
113. WRITE(7,4020) NSP,NA,ND,NVC,NM,NVC,NVC,ND,NVC
114. 4020 FORMAT(5X,'2DFNI(',4(I2,','),I2,')',PC(',I2,')',SHARE(',
115. 1 I2,')',/6X,'REAL COST(',I2,',',I2,')',)
116. WRITE(7,4030) ND,NVC,NSP,ND,NSP,NSP,NVC,NA,NVC
117. 4030 FORMAT(5X,'3CF(',I2,',',I2,')',CMESH(',I2,',',I2,')',C3(',
118. 1 I2,')',C4(',I2,')',VESTO(',I2,')',/5X,'+TDF(',I2,',',I2,')',)
119. WRITE(7,4040) NA,NSP,NVC,NA,NVC,NA,ND,NSP,NBC,NVC,NM

```

```

120. 4040 FORMAT(5X,'5TDFN(',2(I2,','),I2,')',XNV(',I2,',',I2,')',
121. 1      ,BCATCH(',5(I2,','),I2,')')
122.      WRITE(7,4050) NA,ND,NSP,NBC,NVC,NM,NA,ND,NBC,NVC,NM,NBC
123. 4050 FORMAT(5X,'4VCATCH(',5(I2,','),I2,')',
124. 1      ,PERF(',4(I2,','),I2,')',PCATCH(',I2,')')
125.      WRITE(7,4060) NC,NA,NSP,NVC,NC
126. 4060 FORMAT(5X,'6TYR(',I3,',',.2(I2,','),I2,')',PRNT(',I3,')')
127. C
128. C -- PRODUCE ASSIGNMENT STATEMENTS FOR NER AND NPH
129. C
130.      WRITE(7,5010) NER,NPH
131. 5010 FORMAT(6X,'NER=',I2/6X,'NPH=',I2)
132. C
133. C -- PRODUCE CALL TO SUBPROGRAMS
134. C
135.      WRITE(7,5000)
136. 5000 FORMAT(6X,
137. + 'CALL MANE(ISZ,SZCOF,GCDF,MSZC,CONV,C1,C2,COST,'
138. + /5X,'+ SL,CDL,TM,E,ICOF,SZ,GRT,'
139. + /5X,'+FMAX,ER,NUM,NMCOF,CCAT,BCAT,RCAT,FMORT,RFEF,RFP,CULL,DFN,'
140. + /5X,'+DFNI,RENT,CS,OPC,VC,REV,PMCS,TDM,TDSM,TDSVM,'
141. + /5X,'+TDSV,TDS,PMCS,VKIL,TYR,NV,'
142. + /5X,'+TCOST,CCDST,CF,CMESH,C3,C4,TDF,TDFN,XNV,'
143. + /5X,'+BCATCH,PERF,VCATCH,'
144. + /5X,'+NER,NPH,HP,FRL,TD,TREV,TRENT,TCS,'
145. + /5X,'+',TOPC,TVC,FC,OCO,OCC,C,CVC,PC,SHARE,VESTO,PCATCH,'
146. + /5X,'+',PRNT,TAIL')
147. C
148. C -- PRODUCE STOP/END
149. C
150.      WRITE(7,6000)
151. 6000 FORMAT(6X,'STOP'/6X,'END')
152. 99      STOP
153.      END

```

Appendix C - SORIGIN

```

1.      BLOCK DATA
2.      COMMON/COLSE/ KBD(18),KED(18),KBP(18),KEP(18),KAR(18),
3.      +      KDT(18),KSP(18)
4.      COMMON/ADJUST/ ADJ(18)
5.      COMMON/WEEK/ KK(4,18)
6.      COMMON/PARAM/TIME,NSP,ND,NC,NA,NVC,NSC,NM,NSX,NBC,CN
7.      DATA KBD/18*O/,KED/18*O/,KBP/18*O/,KEP/18*O/,KAR/18*O/
8.      +      ,KDT/18*O/,KSP/18*O/
9.      DATA ADJ/18*O.O/
10.     DATA KK/72*O/
11.     DATA NSP,ND,NVC,NA,NC/ 2, 3, 2, 1, 96/
12.     DATA NSC,NM,NBC,NSX/ 8,12, 1, 2/
13.     END
14.     COMMON/COLSE/ KBD(18),KED(18),KBP(18),KEP(18),KAR(18),
15.     +      KDT(18),KSP(18)
16.     COMMON/ADJUST/ ADJ(18)
17.     COMMON/WEEK/ KK(4,18)
18.     COMMON/PARAM/TIME,NSP,ND,NC,NA,NVC,NSC,NM,NSX,NBC,CN
19.     COMMON/SFISH/BMESH,RMESH,BC,RC,C5,C6
20.     REAL ISZ( 2, 2),SZCOF( 2, 2),GCOF( 2, 2),
21.     +MSZC( 2, 2)
22.     REAL CONV( 2),C1( 2),C2( 2),
23.     1HP( 2),FRL( 2),TD( 3),TREV( 2),TRENT( 2),
24.     +TCS( 2),TOPC( 2),TVC( 2),SL( 2, 8),COL(12),
25.     +TM(12),
26.     3E( 2,48),ICOF( 2, 1)
27.     REAL SZ( 2, 2, 96),GRT( 2,12),
28.     4FMAX( 2, 3, 1),ER( 2, 3, 1, 1),
29.     +NUM( 2, 2, 3, 96, 1),
30.     5NMCOF( 2, 3, 8),CCAT( 2, 3, 1, 2, 8,12),
31.     +BCAT( 2, 1,12)
32.     REAL FMORT( 2, 3, 2, 1),RFEF( 2, 3, 2, 1),
33.     +RFP( 2, 3, 2),
34.     1CULL( 2, 3, 1, 2,12),
35.     +DFN( 2, 1, 3, 2,12),
36.     2RENT( 2, 1,12, 2),
37.     +CS( 2, 1,12, 2),
38.     +OPC( 2, 1,12, 2)
39.     REAL VC( 2, 1,12, 2),
40.     +REV( 2, 1,12, 2),PMCS( 2, 8,12),
41.     + TDM( 3,12),TDSM( 3, 8,12),
42.     + TDSVM( 3, 8, 2,12),TDSV( 3, 8,12),
43.     + TDS( 3, 8),PMCS( 2, 8,12)
44.     REAL VKIL( 1, 2, 3,12),RCAT( 2, 1,12),
45.     +NV( 1, 2),FC( 2),
46.     1OCO( 2),OCC( 2),C( 2),CVC( 2),TCOST( 1, 2, 2),
47.     +CCOST( 1, 2, 2),
48.     2DFNI( 2, 1, 3, 2,12),PC( 2),SHARE( 2)
49.     REAL COST( 3, 2),
50.     3CF( 3, 2),CMESH( 2, 3),C3( 2),C4( 2),VESTO( 2),
51.     +TDF( 1, 2),
52.     5TDFN( 1, 2, 2),XNV( 1, 2),BCATCH( 1, 3, 2, 1, 2,12),
53.     4VCATCH( 1, 3, 2, 1, 2,12),PERF( 1, 3, 1, 2,12),PCATCH( 1),
54.     6TYR( 96, 1, 2, 2),PRNT( 96)
55.     NER=48
56.     NPH=12
57.     CALL MANE(ISZ,SZCOF,GCOF,MSZC,CONV,C1,C2,COST,
58.     + SL,COL,TM,E,ICOF,SZ,GRT,

```

```

59.      +FMAX, ER, NUM, NMCOF, CCAT, BCAT, RCAT, FMORT, RFEF, RFP, CULL, DFN,
60.      +DFNI, RENT, CS, OPC, VC, REV, PMCS, TDM, TDSM, TDSVM,
61.      +TDSV, TDS, PMCSP, VKIL, TYR, NV,
62.      +TCOST, CCOST, CF, CMESH, C3, C4, TDF, TDFN, XNV,
63.      +BCATCH, PERF, VCATCH,
64.      +NER, NPH, HP, FRL, TD, TREV, TRENT, TCS,
65.      +TOPC, TVC, FC, OCO, OCC, C, CVC, PC, SHARE, VESTO, PCATCH,
66.      +PRNT, TAIL)
67.      STOP
68.      END
69.      C
70.      C
71.      SUBROUTINE MANE( ISZ, SZCOF, GCOF, MSZC, CONV, C1, C2, COST,
72.      1SL, COL, TM, E, ICOF, SZ, GRT, FMAX, ER
73.      2, NUM, NMCOF, CCAT, BCAT, RCAT, FMORT, RFEF, RFP, CULL, DFN, DFNI, RENT, CS,
74.      3OPC, VC, REV, PMCS, TDM, TDSM, TDSVM, TDSV, TDS,
75.      4PMCSP, VKIL, TYR, NV, TCOST, CCOST, CF, CMESH,
76.      5C3, C4, TDF, TDFN, XNV, BCATCH, PERF, VCATCH,
77.      +NER, NPH, HP, FRL, TD, TREV, TRENT, TCS,
78.      +TOPC, TVC, FC, OCO, OCC, C, CVC, PC, SHARE, VESTO, PCATCH,
79.      + PRNT, TAIL)
80.      INTEGER TIME, FTIME
81.      DIMENSION Q(20)
82.      COMMON/COLSE/ KBD( 18 ), KED( 18 ), KBP( 18 ), KEP( 18 ), KAR( 18 ), KDT( 18 ),
83.      +
84.      KSP( 18 )
85.      C
86.      COMMON/ADJUST/ ADJ( 18 )
87.      C
88.      COMMON/WEEK/ KK( 4, 18 )
89.      COMMON/PARAM/TIME, NSP, ND, NC, NA, NVC, NSC, NM, NSX, NBC, CN
90.      COMMON/SFISH/BMESH, RMESH, BC, RC, C5, C6
91.      INTEGER IX( 5 )
92.      REAL ISZ( NSP, NSX ), SZCOF( NSP, NSX ), GCOF( NSP, NSX ), MSZC( NSP, NSX )
93.      REAL CONV( NSP ), C1( NSP ), C2( NSP ),
94.      1HP( NVC ), FRL( NVC ), TD( ND ), TREV( NVC ), TRENT( NVC ),
95.      2TCS( NVC ), TOPC( NVC ), TVC( NVC ), SL( NSP, NSC ), COL( NM ), TM( NM ), E( NS
96.      3P, NER ), ICOF( NSP, NA ), SZ( NSP, NSX, NC ), GRT( NSP, NPH ), FMAX( NSP, ND, NA ),
97.      4ER( NSP, ND, NA, NA ), NUM( NSP, NSX, ND, NC, NA ), NMCOF( NSP, ND, NSC ),
98.      5CCAT( NSP, ND, NA, NVC, NSC, NM ), BCAT( NSP, NA, NM )
99.      REAL FMORT( NSP, ND, NVC, NA ), RFEF( NSP, ND, NVC, NA ), RFP( NSP, ND, NVC ), CULL
100.     1( NSP, ND, NA, NVC, NM ), DFN( NSP, NA, ND, NVC, NM ), RENT( NSP, NA, NM, NVC ), CS( NS
101.     2P, NA, NM, NVC ), OPC( NSP, NA, NM, NVC ), VC( NSP, NA, NM, NVC ), REV( NSP, NA, NM, NV
102.     3C ), PMCS( NSP, NSC, NM ), TDM( ND, NM ), TDSM( ND, NSC, NM ),
103.     4TDSVM( ND, NSC, NVC, NM ), TDSV( ND, NSC, NM ),
104.     5TDS( ND, NSC ), PMCSP( NSP, NSC, NM ), YR( 100 )
105.     REAL VKIL( NA, NSP, ND, NM ), RCAT( NSP, NA, NM ), NV( NA, NVC ), FC( NVC ), OCO( NVC
106.     1 ), OCC( NVC ), C( NVC ), CVC( NVC ), TCOST( NA, NSP, NVC ), CCOST( NA, NSP, NVC ), N
107.     2U( 5 ), SP( 10 ), DFNI( NSP, NA, ND, NVC, NM ), PC( NVC ), SHARE( NVC ), COST( ND, NVC )
108.     3, CF( ND, NVC ), CMESH( NSP, ND ), C3( NSP ), C4( NSP ), VESTO( NVC ), TDF( NA, NVC ),
109.     4TDFN( NA, NSP, NVC ), XNV( NA, NVC ), BCATCH( NA, ND, NSP, NBC, NVC, NM ), VCATCH
110.     5( NA, ND, NSP, NBC, NVC, NM ), PERF( NA, ND, NBC, NVC, NM ), PCATCH( NBC )
111.     REAL TYR( NC, NA, NSP, NVC ), PRNT( NC )
112.     LOGICAL A( 40 )
113.     C
114.     C... INITIALIZE ARRAY NUM
115.     C
116.     DO 7004 I=1, NSP
117.         DO 7014 J=1, NSX
118.             DO 7024 K=1, NC
119.                 SZ( I, J, K ) = 0
120.                 DO 7003 L = 1, ND

```

```

120.                                DO 7013 M =1,NA
121.                                NUM(I,J,L,K,M) = 0
122.    7013                        CONTINUE
123.    7003                        CONTINUE
124.    7024                        CONTINUE
125.    7014                        CONTINUE
126.    7004 CONTINUE
127.    C
128.    C...READ IN VARIOUS COEFFICIENTS (OR PARAMETERS)
129.    C
130.        CALL INOUT(A, ITER, ISZ, SZCOF, GCOF, MSZC, CONVP, C1, C2, COST,
131.        1SL, COL, TM, E, ICOF, SZ, GRT, FMAX, ER
132.        2, NUM, NMCOF, CCAT, BCAT, RCAT, FMORT, RFEF, RFP, CULL, DFN, DFNI, RENT, CS,
133.        3OPC, VC, REV, PMCS, TDM, TDSM, TDSVM, TDSV, TDS, CPH, FTIME,
134.        4PMCS, SP, VKIL, TYR, YR, NV, TCOST, CCOST, CF, CMESH, IC, CCT, CER,
135.        5C3, C4, TDF, TDFN, XNV, BCATCH, PERF, VCATCH,
136.        +NER, NPH, HP, FRL, TD, TREV, TRENT, TCS, C7, C8, ETM, IX,
137.        +TOPC, TVC, FC, OCO, OCC, C, CVC, PC, SHARE, VESTO, PCATCH, TVALUE,
138.        + PRNT, TAIL)
139.    C
140.        COUNT = ITER
141.    C
142.    C...CALCULATE RELATIVE CATCHABILITY OF VESSELS PRESENT
143.    C
144.        CALL RELCAT(RFP, HP, FRL, C1, C2)
145.    C
146.    C...INITIALIZATION SECTION
147.    C
148.        200 CONTINUE
149.        IRUN= COUNT+1-ITER
150.        NCHRT= 0
151.    C...ALTER EFFORT
152.    C
153.        CALL EFFORT(ITER, COUNT, A, NV, XNV, YR, IRUN, TYR, DFN, DFNI)
154.    C
155.        199 CONTINUE
156.    C
157.        CALL INTIAL(TDFN, TCOST, CCOST, PMCS, PMCS, BCATCH, VCATCH, RENT,
158.        +          CS, VC, OPC, REV, FMORT, RFEF, RCAT, BCAT, CULL, CCAT, IC,
159.        +          TDF, NU)
160.    C
161.        216 CONTINUE
162.        TIME= 0
163.        IF(.NOT.(A(11).OR.A(12).OR.A(13))) GO TO 301
164.    C
165.    C...STOCHASTIC SECTION
166.    C
167.        CALL STOCHA(A, SP, IX, NU)
168.    C
169.        301 CONTINUE
170.    C-----
171.    C MAIN LOOP
172.    C-----
173.        400 TIME= TIME+1
174.    C INTRODUCE FRY
175.        CALL INTRO(E, NUM, ICOF, NCHRT, ISZ, CCT, CER, NER, SZ, NU(1), A)
176.    C MOVE SHRIMP ACCORDING TO EMIGRATION RATES
177.        CALL MOVE(NUM, SZ, ER, SZCOF, C8, NU(2))
178.    C GROW SHRIMP
179.        CALL GROW(SZ, GRT, GCOF, MSZC, CPH, NPH, NUM, NU(3), A)
180.    C KILL SHRIMP DUE TO NATURAL MORTALITY

```



```

181.          CALL NMORT(NUM,SZ,NMCOF,SL,C8,NU(4))
182. C HARVEST SHRIMP
183.          CALL FISH(NUM,SZ,CCAT,CONV,FMAX,FMORT,RFP,DFN,RFEF,C3,C4,ETM,SL,CU
184.          2LL,CF,C7,C8,A,NU(5),VKIL,CMESH,TDF,TDFN,IC,BCATCH,PERF)
185. C CONDUCT BAIT FISHING
186.          IF(BC.EQ.O.O) GO TO 477
187.          CALL BAIT(NUM,SZ,BCAT,C3,C4,ETM,CONV)
188.          477 CONTINUE
189. C CONDUCT RECREATIONAL FISHING
190.          IF(RC.EQ.O.O) GO TO 478
191.          CALL RECF(NUM,SZ,FMORT,RCAT,C3,C4,ETM,CONV)
192.          478 CONTINUE
193. C CALCULATE ECONOMICS
194.          IF(.NOT.A(8)) GO TO 420
195.          CALL ECON(RENT,CS,VC,OPC,REV,COST,DFN,CCAT,PMCS,SHARE,PC,SL,IRUN,C
196.          1VC,TCOST,CCDST,A,IC)
197.          420 CONTINUE
198.          IF(TIME.LT.FTIME) GO TO 400
199. C-----
200. C END LOOP
201. C-----
202.          ITER= ITER-1
203.          IF(.NOT.A(1)) GO TO 401
204.          IF((ITER.NE.O).AND.(.NOT.A(2))) GO TO 401
205. C
206.          CALL NDLOOP(IRUN,A,SZ,NUM)
207. C
208.          401 CONTINUE
209.          IF(ITER.EQ.O) GO TO 996
210.          IF(PRNT(IRUN),NE,O.O) GO TO 995
211.          996 IF((ITER.NE.O).AND.(.NOT.A(3))) GO TO 409
212. C
213. C...PRINT OUTPUT (FOR SELECTED OPTIONS)
214. C
215.          995 CALL WRITUP(CCAT,BCAT,RCAT,RENT,REV,OPC,CS,VC,CULL,COL,
216.          +              A,TM,TDM,TDSM,TDSVM,TDSV,
217.          +              TDS,TD,TREV,TRENT,
218.          +              TCS,TOPC,TVC,TCOST,CCOST,IRUN,COUNT,OCO,
219.          +              OCC,C,NV,FC,DFN,VESTO,
220.          +              TDF,TDFN,BCATCH,VCATCH,PCATCH,SHARE,TVALUE)
221.          409 CONTINUE
222.          IF(ITER.NE.O) GO TO 200
223.          RETURN
224.          END
225. C-----
226. C SUBROUTINES
227. C-----
228.          SUBROUTINE INTRO(E,NUM,ICOF,NCHRT,ISZ,CCT,CER,NER,SZ,ZB,A)
229.          INTEGER TIME
230.          LOGICAL A(40)
231.          COMMON/PARAM/TIME,NSP,ND,NC,NA,NVC,NSC,NM,NSX,NBC,CN
232.          REAL E(NSP,NER),ICOF(NSP,NA),NUM(NSP,NSX,ND,NC,NA),ISZ(NSP,NSX),
233.          1SZ(NSP,NSX,NC)
234.          IF(NCHRT.GE.NC) NCHRT= 1
235.          IF((TIME/CCT).EQ.(FLOAT(TIME)/CCT)) NCHRT= NCHRT+1
236.          NCT= (TIME/CER)+1
237.          IF(NCT.GT.NER) NCT= NCT-((NCT/NER)*NER)
238.          K= NCHRT
239.          DO 10 I= 1,NSP
240.          DO 30 J=1,NSX
241.          SZ(I,J,K)= ISZ(I,J)

```

```

242.          DO 20 L= 1,NA
243.          NUM(I,J,1,K,L)= E(I,NCT)*(ICOF(I,L)+(ZB*ICOF(I,L)))
244.          20 CONTINUE
245.          30 CONTINUE
246.          10 CONTINUE
247.          RETURN
248.          END
249.
250.          C
251.          C
251.          SUBROUTINE MOVE(NUM,SZ,ER,SZCOF,C8,ZB)
252.          COMMON/PARAM/TIME,NSP,ND,NC,NA,NVC,NSC,NM,NSX,NBC,CN
253.          REAL NUM(NSP,NSX,ND,NC,NA),SZ(NSP,NSX,NC),ER(NSP,ND,NA,NA),
254.          1SZCOF(NSP,NSX)
255.          DO 10 I= 1,NSP
256.             DO 11 J7=1,NSX
257.                DO 20 L= 1,NA
258.                   DO 30 K= 1,NC
259.                      IF(SZ(I,J7,K).LE.SZCOF(I,J7)) GO TO 60
260.                      A= (ER(I,1,L,L)+(ZB*ER(I,1,L,L)))*NUM(I,J7,1,K,L)
261.                      DO 40 J1= 2,ND
262.                         J=J1
263.                         IF(ND.EQ.1) J= 1
264.                         EDEPTH= A-(ER(I,J,L,L)+(ZB*ER(I,J,L,L)))*
265.                         NUM(I,J7,J,K,L)
266.                         EAREA= 0.0
267.                         A= (ER(I,J,L,L)+(ZB*ER(I,J,L,L)))*
268.                         NUM(I,J7,J,K,L)
269.                         IF(NA.LE.2) GO TO 70
270.                         IF(L.EQ.NA) GO TO 80
271.                         IF(L.EQ.1) GO TO 90
272.                         EAREA= (ER(I,J,L-1,L)+(ZB*ER(I,J,L-1,L)))*
273.                         NUM(I,J7,J,K,L-1)+(ER(I,
274.                         J,L+1,L)+(ZB*ER(I,J,L+1,L)))*
275.                         NUM(I,J7,J,K,L+1)-(ER(I,J,L,L-1)+(ZB*
276.                         ER(I,L,L,L-1)))*NUM(I,J7,J,K,L)-
277.                         (ER(I,J,L,L+1)+(ZB*ER(I,J,L,L+1)))
278.                         *NUM(I,J7,J,K,L)
279.                         GO TO 100
280.                         80
280.                         CONTINUE
281.                         EAREA= (ER(I,J,L-1,L)+(ZB*ER(I,J,L-1,L)))*
282.                         NUM(I,J7,J,K,L-1)-(ER(I,J,
283.                         L,L-1)+(ZB*ER(I,J,L,L-1)))*NUM(I,J7,J,K,L)
284.                         GO TO 110
285.                         90
285.                         CONTINUE
286.                         EAREA= (ER(I,J,L+1,L)+(ZB*ER(I,J,L+1,L)))*
287.                         NUM(I,J7,J,K,L+1)-(ER(I,J,L,L+1)+
288.                         (ZB*ER(I,J,L,L+1)))*NUM(I,J7,J,K,L)
289.                         110
289.                         CONTINUE
290.                         100
290.                         CONTINUE
291.                         70
291.                         CONTINUE
292.                         NUM(I,J7,J,K,L)= NUM(I,J7,J,K,L)+EDEPTH+EAREA
293.                         IF(NUM(I,J7,J,K,L).LT.C8) NUM(I,J7,J,K,L)= 0
294.                         40
294.                         CONTINUE
295.                         NUM(I,J7,1,K,L)= NUM(I,J7,1,K,L)-(ER(I,1,L,L)+
296.                         (ZB*ER(I,1,L,L)))*NU
297.                         M(I,J7,1,K,L)
298.                         60
298.                         CONTINUE
299.                         30
299.                         CONTINUE
300.                         20
300.                         CONTINUE
301.                         11
301.                         CONTINUE
302.                         10
302.                         CONTINUE

```

```

303.         RETURN
304.         END
305.     C
306.     C
307.         SUBROUTINE GROW(SZ,GRT,GCOF,MSZC,CPH,NPH,NUM,ZB,A)
308.         INTEGER TIME
309.         LOGICAL A(40)
310.         COMMON/PARAM/TIME,NSP,ND,NC,NA,NVC,NSC,NM,NSX,NBC,CN
311.         REAL SZ(NSP,NSX,NC),GCOF(NSP,NSX),MSZC(NSP,NSX)
312.         REAL NUM(NSP,NSX,ND,NC,NA),GRT(NSP,NPH)
313.         NCT= (TIME/CPH)+1
314.         IF(NCT.GT.NPH) NCT= NCT-((NCT/NPH)*NPH)
315.         DO 10 I= 1,NSP
316.         DO 11 J7=1,NSX
317.         DO 20 K= 1,NC
318.         DO 30 J= 1,ND
319.         DO 40 L= 1,NA
320.         IF(NUM(I,J7,J,K,L).GE.1) GO TO 50
321.     40 CONTINUE
322.     30 CONTINUE
323.         GO TO 60
324.     50 CONTINUE
325.         SZ(I,J7,K)= SZ(I,J7,K)+GCOF(I,J7)*(GRT(I,NCT)+(ZB*GRT(I,NCT)))*
326.         1(MSZC(I,J7)-SZ(I,J7,K))
327.         IF(K.NE.1) GO TO 60
328.         IF(A(30)) WRITE(6,100)TIME,I,J7,K,SZ(I,J7,K)
329.     100 FORMAT(' GROW   TIME=',I3,' SP=',I1,' SEX=',I1,' COHORT=',I3,
330.         +      ' SZ=',F5.1)
331.         60 CONTINUE
332.         20 CONTINUE
333.         11 CONTINUE
334.         10 CONTINUE
335.         RETURN
336.         END
337.     C
338.     C
339.         SUBROUTINE NMORT(NUM,SZ,NMCOF,SL,C8,ZB)
340.         COMMON/PARAM/TIME,NSP,ND,NC,NA,NVC,NSC,NM,NSX,NBC,CN
341.         REAL NUM(NSP,NSX,ND,NC,NA),SZ(NSP,NSX,NC),NMCOF(NSP,ND,NSC),
342.         1SL(NSP,NSC)
343.         M1= NSC-1
344.         DO 10 I= 1,NSP
345.         DO 11 J7=1,NSX
346.         DO 20 J= 1,ND
347.         DO 30 L= 1,NA
348.         DO 40 K= 1,NC
349.         IF(NUM(I,J7,J,K,L).LT.1) GO TO 50
350.         M= 0
351.         IF(SZ(I,J7,K).GE.SL(I,1)) GO TO 60
352.         IF(NSC.EQ.1) GO TO 90
353.         DO 70 M= 1,M1
354.         IF(SZ(I,J7,K).LT.SL(I,M).AND.SZ(I,J7,K).GE.SL(I,M+1)) GO TO 60
355.     70 CONTINUE
356.         NUM(I,J7,J,K,L)= NUM(I,J7,J,K,L)-NUM(I,J7,J,K,L)*(NMCOF(I,J,NSC)+
357.         1(ZB*NMCOF(I,J,NSC)))
358.         GO TO 80
359.     90 CONTINUE
360.     60 CONTINUE
361.         NUM(I,J7,J,K,L)= NUM(I,J7,J,K,L)-NUM(I,J7,J,K,L)*(NMCOF(I,J,M+1)+
362.         1(ZB*NMCOF(I,J,M+1)))
363.     80 CONTINUE

```

```

364.         IF(NUM(I,J7,J,K,L).LT.CB) NUM(I,J7,J,K,L)= 0
365.         50 CONTINUE
366.         40 CONTINUE
367.         30 CONTINUE
368.         20 CONTINUE
369.         11 CONTINUE
370.         10 CONTINUE
371.         RETURN
372.         END
373.
374. C
375. C
376.         SUBROUTINE FISH(NUM,SZ,CCAT,CONV,FMAX,FMORT,RFP,DFN,RFEF,C3,C4,ETM
377.         2,SL,CULL,CF,C7,C8,A,ZB,VKIL,CMESH,TDF,TDFN,IC,BCATCH,PERF)
378.         INTEGER TIME
379.         COMMON/PARAM/TIME,NSP,ND,NC,NA,NVC,NSC,NM,NSX,NBC,CN
380.         REAL NUM(NSP,NSX,ND,NC,NA),SZ(NSP,NSX,NC),CCAT(NSP,ND,NA,NVC,NSC,N
381.         2M),CONV(NSP),FMAX(NSP,ND,NA),FMORT(NSP,ND,NVC,NA),RFEF(NSP,ND,NVC,
382.         3NA),RFP(NSP,ND,NVC),DFN(NSP,NA,ND,NVC,NM),SL(NSP,NSC),CULL(NSP,ND,
383.         4NA,NVC,NM),VKIL(NA,NSP,ND,NM),KILLED,CF(ND,NVC),CMESH(NSP,ND),
384.         5C3(NSP),C4(NSP),TDF(NA,NVC),TDFN(NA,NSP,NVC)
385.         REAL PERF(NA,ND,NBC,NVC,NM),BCATCH(NA,ND,NSP,NBC,NVC,NM)
386.         LOGICAL A(40)
387.         COMMON/SFISH/BMESH,RMESH,BC,RC,C5,C6
388.         CALL FEF(RFP,DFN,RFEF,TDF,TDFN,IC)
389.         N2= NSC-2
390.         M= (TIME/CN)+1
391.         IF(M.GT.NM) M= M-((M/NM)*NM)
392.         DO 10 I= 1,NSP
393.         DO 11 J7=1,NSX
394.         DO 20 J= 1,ND
395.         DO 60 L= 1,NA
396.         TFMORT= 0
397.         DO 30 M1= 1,NVC
398. C...CALCULATE BY-CATCH OF FISH
399.         IF(.NOT.A(28)) GO TO 201
400.         DO 200 IOS=1,NBC
401.         BCATCH(L,J,I,IOS,M1,M)=BCATCH(L,J,I,IOS,M1,M)+RFEF(I,J,M1,L)*
402.         1PERF(L,J,IOS,M1,M)
403.         200 CONTINUE
404.         201 CONTINUE
405. C.....
406.         FMORT(I,J,M1,L)=RFEF(I,J,M1,L)*(FMAX(I,J,L)+(ZB*FMAX(I,J,L)))
407.         TFMORT= TFMORT+FMORT(I,J,M1,L)
408.         30 CONTINUE
409.         IF(.NOT.A(9)) GO TO 190
410.         X= TFMORT*100
411.         WRITE(6,31)TIME,J,L,X
412.         31 FORMAT(' TIME=',I3,2X,'DEPTH',1X,I2,2X,'AREA',1X,I2,2X,'FMORT=',F1
413.         10.4)
414.         190 CONTINUE
415.         IF(TFMORT.LE.1) GO TO 40
416.         X= TFMORT*100
417.         DO 50 M1= 1,NVC
418.         FMORT(I,J,M1,L)= FMORT(I,J,M1,L)/TFMORT
419.         50 CONTINUE
420.         40 CONTINUE
421.         DO 100 N= 1,NVC
422.         IF(FMORT(I,J,N,L).LE.0.0) GO TO 170
423.         DO 70 K= 1,NC
424.         CATCH = 0.0
425.         IF(NUM(I,J7,J,K,L).LT.1) GO TO 80

```

```

425.         IF(SZ(I,J7,K).LT.CMESH(I,J)) GO TO 90
426.         KILLED= 1.0
427.         IF(SZ(I,J7,K).LE.C5. AND .SZ(I,J7,K).GE.C6) KILLED= C7
428.         CATCH= FMORT(I,J,N,L)*KILLED*NUM(I,J7,J,K,L)*C3(I)*SZ(I,J7,K)**C4(
429.         1I)/ETM/CONV(I)
430.         NUM(I,J7,J,K,L)= NUM(I,J7,J,K,L)-FMORT(I,J,N,L)*KILLED*NUM(I,J7,J,
431.         1K,L)
432.         IF(NUM(I,J7,J,K,L).LT.C8) NUM(I,J7,J,K,L)= 0
433.         IF(SZ(I,J7,K).GE.VKIL(L,I,J,M)) GO TO 110
434.         CULL(I,J,L,N,M)= CULL(I,J,L,N,M)+CATCH*(1-CF(J,N))
435.         CCAT(I,J,L,N,NSC,M)= CCAT(I,J,L,N,NSC,M)+CATCH*CF(J,N)
436.         IF(.NOT.A(29)) GO TO 203
437.         DO 202 IOS=1,NBC
438.         202 BCATCH(L,J,I,IOS,N,M)=BCATCH(L,J,I,IOS,N,M) + CATCH*CF(J,N)
439.         1*PERF(L,J,IOS,N,M)
440.         203 CONTINUE
441.         GO TO 150
442.         110 CONTINUE
443.         N1= 0
444.         IF(SZ(I,J7,K).GE.SL(I,1)) GO TO 120
445.         IF(NSC.EQ.1) GO TO 160
446.         DO 130 N1= 1,N2
447.         IF(SZ(I,J7,K).LT.SL(I,N1). AND .SZ(I,J7,K).GE.SL(I,N1+1))GO TO 140
448.         130 CONTINUE
449.         CCAT(I,J,L,N,NSC,M)= CCAT(I,J,L,N,NSC,M)+CATCH
450.         IF(.NOT.A(29)) GO TO 205
451.         DO 204 IOS=1,NBC
452.         204 BCATCH(L,J,I,IOS,N,M)=BCATCH(L,J,I,IOS,N,M) + CATCH
453.         1*PERF(L,J,IOS,N,M)
454.         205 CONTINUE
455.         GO TO 180
456.         160 CONTINUE
457.         140 CONTINUE
458.         120 CONTINUE
459.         CCAT(I,J,L,N,N1+1,M)= CCAT(I,J,L,N,N1+1,M)+CATCH
460.         IF(.NOT.A(29)) GO TO 207
461.         DO 206 IOS=1,NBC
462.         206 BCATCH(L,J,I,IOS,N,M)=BCATCH(L,J,I,IOS,N,M) + CATCH
463.         1*PERF(L,J,IOS,N,M)
464.         207 CONTINUE
465.         180 CONTINUE
466.         150 CONTINUE
467.         90 CONTINUE
468.         80 CONTINUE
469.         70 CONTINUE
470.         170 CONTINUE
471.         100 CONTINUE
472.         60 CONTINUE
473.         20 CONTINUE
474.         11 CONTINUE
475.         10 CONTINUE
476.         RETURN
477.         END
478.     C
479.     C
480.         SUBROUTINE ECON(RENT,CS,VC,OPC,REV,COST,DFN,CCAT,PMCS,SHARE,PC,SL,
481.         1IRUN,CVC,TCOST,CCOST,A,IC)
482.         INTEGER TIME
483.         COMMON/COLSE/ KBD(18),KED(18),KBP(18),KEP(18),KAR(18),KDT(18),
484.         + KSP(18)
485.     C

```

```

486.          COMMON/ADJUST/ ADJ(18)
487.
488.          C          COMMON/WEEK/ KK(4,18)
489.          COMMON/PARAM/TIME,NSP,ND,NC,NA,NVC,NSC,NM,NSX,NBC,CN
490.          REAL RENT(NSP,NA,NM,NVC),CS(NSP,NA,NM,NVC),OPC(NSP,NA,NM,NVC),VC(N
491.          2SP,NA,NM,NVC),REV(NSP,NA,NM,NVC),DFN(NSP,NA,ND,NVC,NM),PM
492.          3CS(NSP,NSC,NM),CCAT(NSP,ND,NA,NVC,NSC,NM),SL(NSP,NSC),TCOST(NA,NSP
493.          4,NVC),CCOST(NA,NSP,NVC),CVC(NVC),PC(NVC),SHARE(NVC),COST(ND,NVC)
494.          LOGICAL A(40)
495.          J= (TIME/CN)+1
496.          IF(J.GT.NM) J= J-((J/NM)*NM)
497.          IF((TIME+1).LE.(J*CN)) GO TO 80
498.          IF(.NOT.A(24)) GO TO 81
499.          IF(IRUN.EQ.1) GO TO 81
500.          CALL DEMAND(PMCS,CCAT,J,SL,IRUN,A)
501.          81 CONTINUE
502.          80 CONTINUE
503.          DO 10 I= 1,NSP
504.          DO 20 L= 1,NVC
505.          DO 30 M= 1,NA
506.          DO 40 K1= 1,ND
507.          RATE = 1.0
508.          IF(IC.EQ.0) GO TO 2000
509.          DO 2001 IZZ=1,IC
510.          IZ=IC+1-IZZ
511.          IF(M.EQ.KAR(IZ) .AND. K1.EQ.KDT(IZ) .AND. I.EQ.KSP(IZ) .AND.
512.          + J.GE.KBD(IZ) .AND. J.LE.KED(IZ) .AND. KK(3,IZ).GE.KBP(IZ) .AND.
513.          + KK(4,IZ).EQ.KEP(IZ)) RATE=ADJ(IZ)
514.          IF(J.EQ.KBD(IZ) .AND. L.NE.1 .AND. M.EQ.KAR(IZ)
515.          1 .AND. K1.EQ.KDT(IZ) .AND. I.EQ.KSP(IZ))
516.          2 KK(3,IZ) = KK(3,IZ) + 1
517.          IF(J.EQ.KED(IZ) .AND. L.NE.1 .AND. M.EQ.KAR(IZ)
518.          1 .AND. K1.EQ.KDT(IZ) .AND. I.EQ.KSP(IZ))
519.          2 KK(4,IZ) = KK(4,IZ) + 1
520.          2001 CONTINUE
521.          2000 CONTINUE
522.          VC(I,M,J,L)= VC(I,M,J,L)+COST(K1,L)*DFN(I,M,K1,L,J)*RATE
523.          CCOST(M,I,L)= CCOST(M,I,L)+CVC(L)*DFN(I,M,K1,L,J)*RATE
524.          40 CONTINUE
525.          IF((TIME+1).LE.(J*CN)) GO TO 60
526.          DO 70 K1= 1,ND
527.          DO 50 K= 1,NSC
528.          OPC(I,M,J,L)= OPC(I,M,J,L)+PC(L)*(1-SHARE(L))*CCAT(I,K1,M,L,K,J)
529.          REV(I,M,J,L)= REV(I,M,J,L)+CCAT(I,K1,M,L,K,J)*PMCS(I,K,J)
530.          50 CONTINUE
531.          70 CONTINUE
532.          CS(I,M,J,L)= REV(I,M,J,L)*SHARE(L)
533.          CST= OPC(I,M,J,L)+VC(I,M,J,L)+CS(I,M,J,L)
534.          TCOST(M,I,L)= TCOST(M,I,L)+CST
535.          CCOST(M,I,L)= CCOST(M,I,L)+OPC(I,M,J,L)*SHARE(L)/(1-SHARE(L))
536.          RENT(I,M,J,L)= REV(I,M,J,L)-CST
537.          60 CONTINUE
538.          30 CONTINUE
539.          20 CONTINUE
540.          10 CONTINUE
541.          RETURN
542.          END
543.
544.          C
545.          C          SUBROUTINE FEF(RFP,DFN,RFEF,TDF,TDFN,IC)
546.          INTEGER TIME

```

```

547.      COMMON/COLSE/ KBD(18),KED(18),KBP(18),KEP(18),KAR(18),KDT(18),
548.      +          KSP(18)
549.      C
550.      COMMON/ADJUST/ ADJ(18)
551.      C
552.      COMMON/WEEK/ KK(4,18)
553.      COMMON/PARAM/TIME,NSP,ND,NC,NA,NVC,NSC,NM,NSX,NBC,CN
554.      REAL RFEF(NSP,ND,NVC,NA),DFN(NSP,NA,ND,NVC,NM),RFP(NSP,ND,NVC)
555.      1,TDF(NA,NVC),TDFN(NA,NSP,NVC)
556.      K= (TIME/CN)+1
557.      IF(K.GT.NM) K= K-((K/NM)*NM)
558.      DO 10 J= 1,ND
559.      DO 20 L= 1,NA
560.      DO 40 I= 1,NSP
561.      RATE = 1.0
562.      IF(IC.EQ.0) GO TO 2000
563.      DO 2001 IZZ=1,IC
564.      IZ=IC+1-IZZ
565.      IF(L.EQ.KAR(IZ) .AND. J.EQ.KDT(IZ) .AND. I.EQ.KSP(IZ) .AND.
566.      + K.GE.KBD(IZ) .AND. K.LE.KED(IZ) .AND. KK(1,IZ).GE.KBP(IZ)
567.      + .AND. KK(2,IZ).LE.KEP(IZ)) RATE=ADJ(IZ)
568.      IF(K.EQ.KBD(IZ) .AND. L.EQ.KAR(IZ) .AND. J.EQ.KDT(IZ)
569.      1 .AND. I.EQ.KSP(IZ)) KK(1,IZ) = KK(1,IZ) + 1
570.      IF(K.EQ.KED(IZ) .AND. L.EQ.KAR(IZ) .AND. J.EQ.KDT(IZ)
571.      1 .AND. I.EQ.KSP(IZ)) KK(2,IZ) = KK(2,IZ) + 1
572.      2001 CONTINUE
573.      2000 CONTINUE
574.      DO 30 M= 1,NVC
575.      XXX = DFN(I,L,J,M,K)*RATE
576.      RFEF(I,J,M,L)= RFP(I,J,M)*XXX
577.      TDF(L,M) = TDF(L,M) + XXX
578.      TDFN(L,I,M) = TDFN(L,I,M) + XXX
579.      30 CONTINUE
580.      40 CONTINUE
581.      20 CONTINUE
582.      10 CONTINUE
583.      RETURN
584.      END
585.      C
586.      C
587.      SUBROUTINE RELCAT(RFP,HP,FRL,C1,C2)
588.      COMMON/PARAM/TIME,NSP,ND,NC,NA,NVC,NSC,NM,NSX,NBC,CN
589.      REAL RFP(NSP,ND,NVC),HP(NVC),FRL(NVC),C1(NSP),C2(NSP)
590.      DO 10 I= 1,NSP
591.      SFP= (HP(NVC)**C1(I))*(FRL(NVC)**C2(I))
592.      DO 20 J= 1,ND
593.      DO 30 K= 1,NVC
594.      RFP(I,J,K)= (HP(K)**C1(I))*(FRL(K)**C2(I))/SFP
595.      30 CONTINUE
596.      20 CONTINUE
597.      10 CONTINUE
598.      RETURN
599.      END
600.      C
601.      C
602.      SUBROUTINE BAIT(NUM,SZ,BCAT,C3,C4,ETM,CONV)
603.      INTEGER TIME
604.      COMMON/PARAM/TIME,NSP,ND,NC,NA,NVC,NSC,NM,NSX,NBC,CN
605.      COMMON/SFISH/BMESH,RMESH,BC,RC,C5,C6
606.      REAL NUM(NSP,NSX,ND,NC,NA),SZ(NSP,NSX,NC),BCAT(NSP,NA,NM),CONV(NSP
607.      1),C3(NSP),C4(NSP)

```

```

608.          J= 1
609.          M= (TIME/CN)+1
610.          IF(M.GT.NM) M= M-((M/NM)*NM)
611.          DO 10 I= 1,NSP
612.          DO 11 J7=1,NSX
613.          DO 20 K= 1,NC
614.          DO 30 L= 1,NA
615.          IF(NUM(I,J7,J,K,L).LT.1) GO TO 40
616.          IF(SZ(I,J7,K).LT.BMESH) GO TO 50
617.          BCAT(I,L,M)= BCAT(I,L,M)+(BC*NUM(I,J7,J,K,L)*C3(I)*SZ(I,J7,K)**
618.          2C4(I)/ETM/CONV(I))
619.          NUM(I,J7,J,K,L)= NUM(I,J7,J,K,L)-(BC*NUM(I,J7,J,K,L))
620.          50 CONTINUE
621.          40 CONTINUE
622.          30 CONTINUE
623.          20 CONTINUE
624.          11 CONTINUE
625.          10 CONTINUE
626.          RETURN
627.          END
628.          C
629.          C
630.          SUBROUTINE RECF(NUM,SZ,FMORT,RCAT,C3,C4,ETM,CONV)
631.          INTEGER TIME
632.          COMMON/PARAM/TIME,NSP,ND,NC,NA,NVC,NSC,NM,NSX,NBC,CN
633.          COMMON/SFISH/BMESH,RMESH,BC,RC,C5,C6
634.          REAL NUM(NSP,NSX,ND,NC,NA),SZ(NSP,NSX,NC),RCAT(NSP,NA,NM),
635.          1FMORT(NSP,ND,NVC),CONV(NSP),C3(NSP),C4(NSP)
636.          J= 1
637.          M= (TIME/CN)+1
638.          IF(M.GT.NM) M= M-((M/NM)*NM)
639.          DO 10 I= 1,NSP
640.          DO 11 J7=1,NSX
641.          DO 20 K= 1,NC
642.          DO 30 L= 1,NA
643.          IF(NUM(I,J7,J,K,L).LT.1) GO TO 40
644.          IF(SZ(I,J7,K).LT.RMESH) GO TO 50
645.          DO 60 M1= 1,NVC
646.          RCAT(I,L,M)= RCAT(I,L,M)+(RC*FMORT(I,J,M1)*NUM(I,J7,J,K,L)*C3(I)*
647.          1SZ(I,J7,K)**C4(I)/ETM/CONV(I))
648.          NUM(I,J7,J,K,L)=NUM(I,J7,J,K,L)-(RC*FMORT(I,J,M1)*NUM(I,J7,J,K,L))
649.          60 CONTINUE
650.          50 CONTINUE
651.          40 CONTINUE
652.          30 CONTINUE
653.          20 CONTINUE
654.          11 CONTINUE
655.          10 CONTINUE
656.          RETURN
657.          END
658.          C
659.          C
660.          SUBROUTINE WRITUP(CCAT,BCAT,RCAT,RENT,REV,OPC,CS,VC,CULL,COL,A,TM,
661.          1TDM,TDSM,TDSVM,TDSV,TDS,TD,TREV
662.          2,TRENT,TCS,TGPC,TVC,TCOST,CCOST,IRUN,COUNT,OCO,OCC,CC,NV,FC,DFN,
663.          3VESTO,TDF,TDFN,BCATCH,VCATCH,PCATCH,SHARE,TVALUE)
664.          COMMON/PARAM/TIME,NSP,ND,NC,NA,NVC,NSC,NM,NSX,NBC,CN
665.          REAL CCAT(NSP,ND,NA,NVC,NSC,NM),BCAT(NSP,NA,NM),RCAT(NSP,NA,NM),RE
666.          1NT(NSP,NA,NM,NVC),REV(NSP,NA,NM,NVC),OPC(NSP,NA,NM,NVC),CS(NSP,NA,
667.          2NM,NVC),VC(NSP,NA,NM,NVC),CULL(NSP,ND,NA,NVC,NM),COL(NM),TM(NM),
668.          3TDM(ND,NM),TDSM(ND,NSC,NM),TDSVM(N

```



```

669.      4D,NSC,NVC,NM),TDSV(ND,NSC,NVC),
670.      5TDS(ND,NSC),TD(ND),TREV(NVC),TRENT(NVC)
671.      REAL TCS(NVC),TOPC(NVC),TVC(NVC),TCOST(NA,NSP,NVC),CCOST(NA,NSP,
672.      +NVC),OCO(NVC),OCC(NVC),NV(NA,NVC),CC(NVC),FC(NVC),DFN(NSP,NA,ND,
673.      2NVC,NM),VESTO(NVC),TDF(NA,NVC),TDFN(NA,NSP,NVC),BCATCH(NA,ND,NSP,
674.      3NBC,NVC,NM),VCATCH(NA,ND,NSP,NBC,NVC,NM),PCATCH(NBC),SHARE(NVC)
675.      4,TS(5),TSV(5,5),TV(5),DFS(5,5),FIXX(5),OPPX(5),DDF(5)
676.      INTEGER IRUN
677.      LOGICAL A(40)
678.      DATA C/' TOT'//,TCC/O.O/,TC/O.O/,RVC/O.O/,RV/O.O/
679.      DO 1 L= 1,NA
680.      DO 94 M=1,NVC
681.      94 TV(M)=O
682.      DO 2 I= 1,NSP
683.      C...INITIALIZATION SECTION
684.      TOT= O
685.      TRCAT= O
686.      TBCAT= O
687.      TS(I)=O
688.      DO 60 M= 1,NVC
689.      TREV(M)= O
690.      TRENT(M)= O
691.      TOPC(M)= O.O
692.      TCS(M)= O
693.      TVC(M)= O
694.      TSV(I,M)=O
695.      60 CONTINUE
696.      DO 3 J= 1,ND
697.      TD(J)= O
698.      DO 4 K= 1,NSC
699.      TDS(J,K)= O
700.      DO 5 M= 1,NVC
701.      TDSV(J,K,M)= O
702.      5 CONTINUE
703.      4 CONTINUE
704.      3 CONTINUE
705.      DO 6 N= 1,NM
706.      TM(N)= O
707.      DO 7 J= 1,ND
708.      TDM(J,N)= O
709.      DO 8 K= 1,NSC
710.      TDSM(J,K,N)= O
711.      DO 9 M= 1,NVC
712.      TDSVM(J,K,M,N)= O
713.      9 CONTINUE
714.      8 CONTINUE
715.      7 CONTINUE
716.      C...SUMMATION SECTION
717.      DO 10 M= 1,NVC
718.      TREV(M)= TREV(M)+REV(I,L,N,M)
719.      TOPC(M)= TOPC(M)+OPC(I,L,N,M)
720.      TCS(M)= TCS(M)+CS(I,L,N,M)
721.      TVC(M)= TVC(M)+VC(I,L,N,M)
722.      TRENT(M)= TRENT(M)+RENT(I,L,N,M)
723.      DO 11 J= 1,ND
724.      DO 12 K= 1,NSC
725.      Z= CCAT(I,J,L,M,K,N)
726.      TOT= TOT+Z
727.      TM(N)= TM(N)+Z
728.      TD(J)= TD(J)+Z
729.      TDM(J,N)= TDM(J,N)+Z

```

```

730.         TDS(J,K)= TDS(J,K)+Z
731.         TDSM(J,K,N)= TDSM(J,K,N)+Z
732.         TDSV(J,K,M)= TDSV(J,K,M)+Z
733.         TDSVM(J,K,M,N)= TDSVM(J,K,M,N)+Z
734.         TSV(I,M)=TSV(I,M)+Z
735.         TV(M)=TV(M)+Z
736.         TS(I)=TS(I)+Z
737.         12 CONTINUE
738.         11 CONTINUE
739.         10 CONTINUE
740.         6 CONTINUE
741. C...PRINT OUT SECTION
742.         WRITE(6,101)IRUN,L,I
743.         101 FORMAT(1H0,' RUN',I4,2X,'AREA',I2,2X,'SPECIES',I2)
744.         WRITE(6,757)
745.         757 FORMAT(' ',T25,'P R E D I C T E D V A L U E S')
746.         IF (.NOT.A(4)) GO TO 14
747.         IF(A(25) .AND. A(20) .AND. IRUN.EQ.COUNT)WRITE(8,701)
748.         701 FORMAT(' ',-----')
749.         IF(A(25) .AND. A(20) .AND. IRUN.EQ.COUNT)WRITE(8,700)
750.         700 FORMAT(' LANDINGS IN THOUSANDS BY SPECIES AND MONTH'/
751.         1' SPEC MTH MEAN SD CIL CIU')
752. C...BY SCP
753.         WRITE(6,102)
754.         102 FORMAT(1H0,' ',25('*'),'TOTAL CATCH BY SCP',26('*'))
755.         WRITE(6,107)
756.         WRITE(6,103)
757.         103 FORMAT(' ',T9,'COMMERCIAL',12X,'RECREATION',12X,'BAIT')
758.         WRITE(6,107)
759.         DO 15 N= 1,NM
760.         TRCAT= TRCAT+RCAT(I,L,N)
761.         TBCAT= TBCAT+BCAT(I,L,N)
762.         IF(.NOT.A(25)) GO TO 500
763.         WRITE(6,104)COL(N),TM(N),RCAT(I,L,N),BCAT(I,L,N)
764.         104 FORMAT(' ',2X,A4,2X,3(F10.0,12X))
765.         KCOUNT=1
766.         IF(A(20) .AND. IRUN.GT.1)CALL SASOUT(TD, TM, TOT, TDM, TDS, CTH, RV,
767.         1TREVE, TCSTS, RRENT, TVALUE, A, IRUN, KCOUNT, COUNT, I, J, K, M, N)
768.         500 CONTINUE
769.         15 CONTINUE
770.         IF(A(20) .AND. IRUN.EQ.COUNT)WRITE(8,701)
771.         IF(A(20) .AND. IRUN.EQ.COUNT)WRITE(8,702)
772.         702 FORMAT(' LANDINGS IN THOUSANDS BY SPECIES '//
773.         1' SPEC MEAN SD CIL CIU')
774.         WRITE(6,104)C, TOT, TRCAT, TBCAT
775.         KCOUNT=2
776.         IF(A(20) .AND. IRUN.GT.1)CALL SASOUT(TD, TM, TOT, TDM, TDS, CTH, RV,
777.         1TREVE, TCSTS, RRENT, TVALUE, A, IRUN, KCOUNT, COUNT, I, J, K, M, N)
778.         14 CONTINUE
779.         IF(.NOT.A(5)) GO TO 289
780. C...BY DEPTH AND SCP
781.         WRITE(6,105)
782.         105 FORMAT(1H0,' ',15('*'),'TOTAL COMMERCIAL CATCH BY DEPTH AND SCP',
783.         115('*'))
784.         WRITE(6,107)
785.         WRITE(6,106)(J, J= 1, ND)
786.         106 FORMAT(' ', 'DEPTH-',7X,10(I2,10X))
787.         WRITE(6,107)
788.         107 FORMAT(' ',-----)
789.         1-----')
790.         IF(.NOT.A(25)) GO TO 501

```

```

791.         IF(A(25) .AND. A(20) .AND. IRUN.EQ.COUNT)WRITE(8,701)
792.         IF(A(25) .AND. A(20) .AND. IRUN.EQ.COUNT)WRITE(8,703)
793.     703 FORMAT(' LANDINGS IN THOUSANDS BY SPECIES, DEPTH AND MONTH'/
794.     1' SPEC DEP MTH MEAN SD CIL CIU')
795.         DO 17 N= 1,NM
796.         WRITE(6,108)COL(N),(TDM(J,N),J= 1,ND)
797.     108 FORMAT(' ',2X,A4,2X,10(F10.0,2X))
798.         17 CONTINUE
799.         IF(.NOT.A(20)) GO TO 501
800.         KCOUNT=3
801.         DO 170 J=1,ND
802.         DO 170 N=1,NM
803.         IF(A(20) .AND. IRUN.GT.1)CALL SASOUT(TD, TM, TOT, TDM, TDS, CTH, RV,
804.         1TREVE, TCSTS, RRENT, TVALUE, A, IRUN, KCOUNT, COUNT, I, J, K, M, N)
805.     170 CONTINUE
806.     501 CONTINUE
807.         IF(A(20) .AND. IRUN.EQ.COUNT)WRITE(8,701)
808.         IF(A(20) .AND. IRUN.EQ.COUNT)WRITE(8,704)
809.     704 FORMAT(' LANDINGS IN THOUSANDS BY SPECIES AND DEPTH'/
810.     1' SPEC DEP MEAN SD CIL CIU')
811.         WRITE(6,108)C,(TD(J),J= 1,ND)
812.         KCOUNT=4
813.         DO 16 J=1,ND
814.         IF(A(20) .AND. IRUN.GT.1)CALL SASOUT(TD, TM, TOT, TDM, TDS, CTH, RV,
815.         1TREVE, TCSTS, RRENT, TVALUE, A, IRUN, KCOUNT, COUNT, I, J, K, M, N)
816.         16 CONTINUE
817.     289 CONTINUE
818.         IF(.NOT.A(6)) GO TO 18
819.     C...BY DEPTH, SIZE CLASS, AND SCP
820.         WRITE(6,109)
821.     109 FORMAT(1H0,' ',9('*'),'TOTAL COMMERCIAL CATCH BY DEPTH,SIZE CLASS,
822.     1AND SCP',10('*'))
823.         DO 19 J= 1,ND
824.         WRITE(6,110)J
825.     110 FORMAT(' ', 'DEPTH', I3)
826.         WRITE(6,107)
827.         WRITE(6,111)(N,N= 1,NSC)
828.     111 FORMAT(' ', 'SIZE CLASS-', 1X, 'CULLS', 7X, 9(I2, 10X))
829.         WRITE(6, 107)
830.         TT= 0
831.         DO 20 N= 1,NM
832.         T= 0
833.         DO 21 M= 1,NVC
834.         T= T+CULL(I, J, L, M, N)
835.     21 CONTINUE
836.         TT= TT+T
837.         IF(.NOT.A(25)) GO TO 502
838.         WRITE(6,108)COL(N),T,(TDSM(J,M,N),M= 1,NSC)
839.     502 CONTINUE
840.     20 CONTINUE
841.         IF(A(20) .AND. IRUN.EQ.COUNT)WRITE(8,701)
842.         IF(A(20) .AND. IRUN.EQ.COUNT)WRITE(8,705)
843.     705 FORMAT(' LANDINGS IN THOUSANDS BY SPECIES, DEPTH AND SIZE'/
844.     1' SPEC DEP SIZ MEAN SD CIL CIU')
845.         WRITE(6,108)C,TT,(TDS(J,M),M= 1,NSC)
846.         KCOUNT=5
847.         DO 19 M=1,NSC
848.         IF(A(20) .AND. IRUN.GT.1)CALL SASOUT(TD, TM, TOT, TDM, TDS, CTH, RV,
849.         1TREVE, TCSTS, RRENT, TVALUE, A, IRUN, KCOUNT, COUNT, I, J, K, M, N)
850.     19 CONTINUE
851.     18 CONTINUE

```

```

852.         IF(.NOT.A(7)) GO TO 22
853. C...BY DEPTH, SIZE CLASS, VESSEL CLASS, AND SCP
854.         WRITE(6,112)
855.         112 FORMAT(1HO,' ',3('*'),'TOTAL COMMERCIAL CATCH BY DEPTH,SIZE CLASS,
856.           1VESSEL CLASS, AND SCP',2('*'))
857.         DO 507 K=1,NVC
858.           VESTO(K) =0.0
859.         507 CONTINUE
860.         DO 23 J= 1,ND
861.         DO 24 K= 1,NVC
862.           XXX=0.
863.         DO 504 M=1,NSC
864.           YYY = TDSV(J,M,K)
865.           XXX = XXX+YYY
866.           VESTO(K) = VESTO(K) + YYY
867.         504 CONTINUE
868.         WRITE(6,110)J
869.         WRITE(6,113)K
870.         113 FORMAT(' ', 'VESSEL CLASS',I3)
871.         WRITE(6,107)
872.         WRITE(6,111)(N,N= 1,NSC)
873.         WRITE(6,107)
874.         T= 0
875.         DO 25 N= 1,NM
876.         T= T+CULL(I,J,L,K,N)
877.         IF(.NOT.A(25)) GO TO 503
878.         WRITE(6,108)COL(N),CULL(I,J,L,K,N),(TDSVM(J,M,K,N),M= 1,NSC)
879.         503 CONTINUE
880.         25 CONTINUE
881.         WRITE(6,108)C,T,(TDSV(J,M,K),M= 1,NSC)
882.         WRITE(6,108)C,XXX
883.         24 CONTINUE
884.         23 CONTINUE
885.         DO 509 K=1,NVC
886.           WRITE(6,508)K,VESTO(K)
887.         509 CONTINUE
888.         508 FORMAT(' TOTAL LANDINGS FOR VESSEL CLASS',I2,' IS'.F10.0)
889.         22 CONTINUE
890.         IF(.NOT.A(8)) GO TO 26
891. C...ECONOMICS
892.         115 FORMAT(' ', 'VESS. CLASS-',10(I2,10X))
893.         WRITE(6,116)
894.         116 FORMAT(' ',/,T32, 'COSTS & RETURNS')
895.         DO 28 K= 1,NVC
896.           WRITE(6,113)K
897.           WRITE(6,107)
898.           WRITE(6,117)
899.         117 FORMAT(' ',14X,'REV',10X,'VC',8X,'PACK',6X,'SHARES',8X,'RENT')
900.         WRITE(6,107)
901.         IF(.NOT.A(25)) GO TO 505
902.         DO 29 N= 1,NM
903.           WRITE(6,108)COL(N),REV(I,L,N,K),VC(I,L,N,K),OPC(I,L,N,K),
904.             1CS(I,L,N,K),RENT(I,L,N,K)
905.         29 CONTINUE
906.         505 CONTINUE
907.         WRITE(6,108)C,TREV(K),TVC(K),TOPC(K),TCS(K),TRENT(K)
908.         28 CONTINUE
909.         26 CONTINUE
910.         2 CONTINUE
911.         1 CONTINUE
912.         IF(.NOT.A(28).AND..NOT.A(29)) GO TO 200

```

```

913.      DO 306 I=1,NSP
914.      DO 306 K=1,ND
915.      DO 306 L=1,NA
916.      DO 242 IOS=1,NBC
917.      DO 242 J=1,NVC
918.      DO 242 JOS=1,NM
919.      VCATCH(L,K,I,IOS,J,JOS)= BCATCH(L,K,I,IOS,J,JOS)*PCATCH(IOS)
920.      242 CONTINUE
921.      WRITE(6,203)
922.      203 FORMAT(/' ***BY-CATCH BY VESSEL CLASS, SCP, SPECIES, AREA, DEPTH'
923.      1,' AND SPECIES OF FISH')
924.      DO 202 L3=1,NBC
925.      WRITE(6,243)L3,I,K,L
926.      243 FORMAT(' BY-CATCH',I2,' SHRIMP SPECIES',I2,' DEPTH',I2,' AREA',
927.      1I2)
928.      WRITE(6,107)
929.      WRITE(6,115)(N,N=1,NVC)
930.      WRITE(6,107)
931.      DO 204 N=1,NM
932.      IF(.NOT.A(25)) GO TO 309
933.      WRITE(6,108)COL(N),(BCATCH(L,K,I,L3,L4,N),L4=1,NVC)
934.      309 CONTINUE
935.      DO 308 L4=1,NVC
936.      IF(N.EQ.1 .AND. L3.EQ.1) GO TO 308
937.      BCATCH(L,K,I,1,L4,1) = BCATCH(L,K,I,1,L4,1)+BCATCH(L,K,I,L3,L4,N)
938.      308 CONTINUE
939.      204 CONTINUE
940.      WRITE(6,108)C,(BCATCH(L,K,I,1,L4,1),L4=1,NVC)
941.      202 CONTINUE
942.      WRITE(6,303)
943.      303 FORMAT(/' ***VALUE BY-CATCH BY VESSEL CLASS, SCP, SPECIES, AREA,'
944.      1,' DEPTH, AND SPECIES OF FISH')
945.      DO 302 L3=1,NBC
946.      WRITE(6,243)L3,I,K,L
947.      WRITE(6,107)
948.      WRITE(6,115)(N,N=1,NVC)
949.      WRITE(6,107)
950.      DO 304 N=1,NM
951.      IF(.NOT.A(25)) GO TO 310
952.      WRITE(6,108)COL(N),(VCATCH(L,K,I,L3,L4,N),L4=1,NVC)
953.      310 CONTINUE
954.      DO 305 L4=1,NVC
955.      IF(N.EQ.1 .AND. L3.EQ.1) GO TO 305
956.      VCATCH(L,K,I,1,L4,1) = VCATCH(L,K,I,1,L4,1)+VCATCH(L,K,I,L3,L4,N)
957.      305 CONTINUE
958.      304 CONTINUE
959.      WRITE(6,108)C,(VCATCH(L,K,I,1,L4,1),L4=1,NVC)
960.      DO 311 L4=1,NVC
961.      IF(L.EQ.1 .AND. K.EQ.1) GO TO 311
962.      VCATCH(1,1,I,1,L4,1)=VCATCH(1,1,I,1,L4,1)+VCATCH(L,K,I,1,L4,1)
963.      311 CONTINUE
964.      302 CONTINUE
965.      306 CONTINUE
966.      200 CONTINUE
967.      IF(A(8) .AND. A(20) .AND. IRUN.EQ.COUNT)WRITE(8,701)
968.      IF(A(8) .AND. A(20) .AND. IRUN.EQ.COUNT)WRITE(8,706)
969.      706 FORMAT(' LANDINGS IN THOUSANDS BY SPECIES AND VESSEL CLASS'/
970.      1' SPEC VES MEAN SD CIL CIU')
971.      GTDF=0.0
972.      DO 600 K=1,NA
973.      DO 510 M =1,NVC

```

```

974.          WRITE(6,601)K
975.    601  FORMAT(////' ECONOMIC DATA FOR AREA',I3)
976.          WRITE(6,511)M,TDF(K,M)
977.          WRITE(6,512)M, NV(K,M)
978.    510  CONTINUE
979.    511  FORMAT('/' TOTAL DAYS FISHED FOR VESSEL CLASS',I2,' IS',F10.0)
980.    512  FORMAT(' NUMBER OF VESSELS IN VESSEL CLASS',I2,' IS',F10.0)
981.  C.... TOTAL ECONOMIC DATA
982.          IF(.NOT.A(21)) GO TO 57
983.  C.... APPORTION VESSELS
984.          DO 150 II=1,NSP
985.            FIXX(II)=0.0
986.            OPPX(II)=0.0
987.            DDF(II)=0.0
988.            DO 150 JJ=1,NVC
989.              DFS(II,JJ)=0.0
990.    150  CONTINUE
991.          DO 70 I= 1,NSP
992.            DO 71 J= 1,NVC
993.              DFS(I,J)=DFS(I,J)+TDFN(K,I,J)/TDF(K,J)
994.    74  CONTINUE
995.    71  CONTINUE
996.    70  CONTINUE
997.  C.... OWNERS COSTS
998.          WRITE(6,129)
999.    129  FORMAT(' ',/,T21,'TOTAL OWNERS COSTS *1000!')
1000.          WRITE(6,107)
1001.          WRITE(6,125)
1002.    125  FORMAT(' VC SP      DAYF $/LB      CATCH      REV-F      REV-B',
1003.    1'      REV-T',7X,'TVC',8X,'FC',8X,'OC',5X,'TCOST',6X,'RENT')
1004.          WRITE(6,107)
1005.          DO 59 I= 1,NSP
1006.            DO 78 J=1,NVC
1007.              DDF(I)=DDF(I)+TDFN(K,I,J)
1008.              IF(NM.EQ.1) GO TO 89
1009.              L= 1
1010.              IF(NM.GT.1) L= 2
1011.    62  CONTINUE
1012.              REV(I,K,1,J)= REV(I,K,1,J)+REV(I,K,L,J)
1013.              CS(I,K,1,J)= CS(I,K,1,J)+CS(I,K,L,J)
1014.              L= L+1
1015.              IF(L.GT.NM) GO TO 61
1016.              GO TO 62
1017.    61  CONTINUE
1018.    89  CONTINUE
1019.              RV=REV(I,K,1,J)/1000.
1020.              CTH=TSV(I,J)/1000.
1021.              PRICE=RV/CTH
1022.              VCAT=VCATCH(K,1,I,1,J,1)/1000.
1023.              FIX= FC(J)*NV(K,J)*DFS(I,J)/1000.
1024.              OPP= OCO(J)*NV(K,J)*DFS(I,J)/1000.
1025.              FIXX(I)=FIXX(I)+FIX
1026.              OPPX(I)=OPPX(I)+OPP
1027.              TREVE=RV+VCAT
1028.              TVCX=TCOST(K,I,J)/1000.
1029.              TCSTS= TVCX + FIX + OPP
1030.              RRENT= TREVE-TCSTS
1031.              WRITE(6,124)J,I,TDFN(K,I,J),PRICE,CTH,RV,VCAT,TREVE,
1032.    1 TVCX,FIX,OPP,TCSTS,RRENT
1033.              KCOUNT=6
1034.              IF(A(20) .AND. IRUN.GT.1)CALL SASOUT(TD,TM,TOT,TDM,TDS,CTH,RV,

```

```

1035.      1TREVE,TCSTS,RRENT,TVALUE,A,IRUN,KCOUNT,COUNT,I,J,K,M,N)
1036. 124 FORMAT(' ',I2,I3,F9.0,F6.2,9F10.0)
1037.      78 CONTINUE
1038.      59 CONTINUE
1039.      FIX= 0.0
1040.      OPP= 0.0
1041.      IF(.NOT.A(22)) GO TO 68
1042.      WRITE(6,107)
1043.      WRITE(6,131)
1044. 131 FORMAT(' ', 'TOT VS. CL. ')
1045.      TC=0.0
1046.      RV=0.0
1047.      CTH = 0.0
1048.      VCA = 0.0
1049.      DO 64 I= 1,NVC
1050.      CTH=TV(I)/1000.
1051.      DO 65 J= 1,NSP
1052.      RV= RV+REV(J,K,1,I)/1000.
1053.      TC= TC+TCOST(K,J,I)/1000.
1054.      VCA = VCA + VCATCH(K,1,J,1,I,1)/1000.
1055.      65 CONTINUE
1056.      FIX= FC(I)*NV(K,I)/1000.
1057.      OPP= OCD(I)*NV(K,I)/1000.
1058.      TREVE = RV + VCA
1059.      TCSTS = TC + FIX + OPP
1060.      RRENT = TREVE - TCSTS
1061.      PRICE = RV/CTH
1062.      WRITE(6,130)I,TDF(K,I),PRICE,CTH,RV,VCA,TREVE,TC,FIX,OPP,TCSTS,
1063.      1RRENT
1064. 130 FORMAT(' ',I5,F9.0,F6.2,9F10.0)
1065.      VCA=0.0
1066.      TC= 0.0
1067.      RV= 0.0
1068.      64 CONTINUE
1069.      68 CONTINUE
1070.      FIX= 0.0
1071.      OPP= 0.0
1072.      IF(.NOT.A(23)) GO TO 58
1073.      WRITE(6,107)
1074.      WRITE(6,132)
1075. 132 FORMAT(' ', 'TOT SP ')
1076.      TC=0.0
1077.      RV=0.0
1078.      GTDF=0.0
1079.      DO 66 J= 1,NSP
1080.      GTDF=GTDF+DDF(J)
1081.      CTH=TS(J)/1000.
1082.      DO 67 I= 1,NVC
1083.      RV= RV+REV(J,K,1,I)/1000.
1084.      TC= TC+TCOST(K,J,I)/1000.
1085.      VCA = VCA + VCATCH(K,1,J,1,I,1)/1000.
1086.      67 CONTINUE
1087.      FIX=FIXX(J)
1088.      OPP=OPPX(J)
1089.      PRICE=RV/CTH
1090.      TREVE = RV + VCA
1091.      TCSTS = TC + FIX + OPP
1092.      RRENT = TREVE - TCSTS
1093.      WRITE(6,130)J,DDF(J),PRICE,CTH,RV,VCA,TREVE,TC,FIX,OPP,TCSTS,
1094.      1RRENT
1095.      FIX= 0.0

```

```

1096.      OPP= 0.0
1097.      TC= 0.0
1098.      RV= 0.0
1099.      VCA=0.0
1100.      66 CONTINUE
1101.      58 CONTINUE
1102.      WRITE(6,107)
1103.      CTH=0.0
1104.      TC=0.0
1105.      RV=0.0
1106.      DO 75 I= 1,NVC
1107.      DO 69 J= 1,NSP
1108.      OPPX(J)=0.0
1109.      CTH= CTH + TSV(J,I)/1000.
1110.      RV= RV+REV(J,K,1,I)/1000.
1111.      TC= TC+TCOST(K,J,I)/1000.
1112.      VCA = VCA + VCATCH(K,1,J,1,I,1)/1000.
1113.      69 CONTINUE
1114.      FIX= FIX+FC(I)*NV(K,I)/1000.
1115.      OPP= OPP+OCO(I)*NV(K,I)/1000.
1116.      75 CONTINUE
1117.      PRICE=RV/CTH
1118.      TREVE = RV + VCA
1119.      TCSTS = TC + FIX + OPP
1120.      RRENT = TREVE - TCSTS
1121.      WRITE(6,133)C,GTDF,PRICE,CTH,RV,VCA,TREVE,TC,FIX,OPP,TCSTS,RRENT
1122.      VCA=0.0
1123.      RVC=0.0
1124.      TCC=0.0
1125.      133 FORMAT(' ',A4,1X,F9.0,F6.2,9F10.0)
1126.      C.....CREWS COSTS
1127.      WRITE(6,127)
1128.      127 FORMAT(' ',/,T19,'TOTAL CREWS COSTS $1000!')
1129.      WRITE(6,107)
1130.      WRITE(6,126)
1131.      126 FORMAT(' ', 'VC SP',5X,'REV-F',5X,'REV-B',5X,'REV-T',7X,
1132.      1'TVC',8X,'OC',5X,'TCOST',6X,'RENT')
1133.      WRITE(6,107)
1134.      DO 76 I= 1,NSP
1135.      DO 77 J= 1,NVC
1136.      RVC=CS(I,K,1,J)/1000.
1137.      TCC=CCOST(K,I,J)/1000.
1138.      OPPC= OCC(J)*NV(K,J)*DFS(I,J)*CC(J)/1000.
1139.      OPPX(J)=OPPX(J)+OPPC
1140.      VCAC=VCATCH(K,1,I,1,J,1)*SHARE(J)/1000.
1141.      TREVE = CS(I,K,1,J)/1000. + VCAC
1142.      TCSTS=TCC + OPPC
1143.      RRENTC = TREVE - TCSTS
1144.      WRITE(6,91)J,I,RVC,VCAC,TREVE,TCC,OPPC,TCSTS
1145.      + ,RRENTC
1146.      91 FORMAT(' ',I2,I3,7F10.0)
1147.      VCAC=0.0
1148.      77 CONTINUE
1149.      76 CONTINUE
1150.      OPPC= 0.0
1151.      IF(.NOT.A(22)) GO TO 80
1152.      WRITE(6,107)
1153.      WRITE(6,131)
1154.      RVC=0.0
1155.      TCC=0.0
1156.      DO 81 I= 1,NVC

```



```

1157.      DO 82 J= 1,NSP
1158.          RVC= RVC+CS(J,K,1,I)/1000.
1159.          TCC= TCC+CCOST(K,J,I)/1000.
1160.          VCAC=VCATCH(K,1,I,1,J,1)*SHARE(J)/1000.
1161.      82 CONTINUE
1162.          OPPC= OCC(I)*NV(K,I)*CC(I)/1000.
1163.          TREVE = RVC + VCAC
1164.          TCSTS = TCC + OPPC
1165.          RRENTC = TREVE - TCSTS
1166.          WRITE(6,92)I,RVC,VCAC,TREVE,TCC,OPPC,TCSTS,RRENTC
1167.      92 FORMAT(' ',I5,7F10.0)
1168.          OPPC= 0.0
1169.          TCC= 0.0
1170.          RVC= 0.0
1171.          VCAC=0.0
1172.      81 CONTINUE
1173.      80 CONTINUE
1174.          IF(.NOT.A(23)) GO TO 83
1175.          WRITE(6,107)
1176.          WRITE(6,132)
1177.          RVC=0.0
1178.          TCC=0.0
1179.          DO 84 J= 1,NSP
1180.              DO 85 I= 1,NVC
1181.                  RVC= RVC+CS(J,K,1,I)/1000.
1182.                  TCC= TCC+CCOST(K,J,I)/1000.
1183.                  OPPC= OPPC+OCC(I)*NV(K,I)*VC(J,K,1,1)*CC(I)/1000.
1184.                  VCAC=VCAC+VCATCH(K,1,I,1,J,1)*SHARE(J)/1000.
1185.              85 CONTINUE
1186.                  OPPC=OPPX(J)
1187.                  TREVE = RVC + VCAC
1188.                  TCSTS = TCC + OPPC
1189.                  RRENTC = TREVE - TCSTS
1190.                  WRITE(6,92)J,RVC,VCAC,TREVE,TCC,OPPC,TCSTS,RRENTC
1191.                  OPPC= 0.0
1192.                  TCC= 0.0
1193.                  RVC= 0.0
1194.                  VCAC=0.0
1195.              84 CONTINUE
1196.          83 CONTINUE
1197.          WRITE(6,107)
1198.          DO 87 I= 1,NVC
1199.              DO 86 J= 1,NSP
1200.                  RVC= RVC+CS(J,K,1,I)/1000.
1201.                  TCC= TCC+CCOST(K,J,I)/1000.
1202.                  VCAC=VCAC+VCATCH(K,1,I,1,J,1)*SHARE(J)/1000.
1203.              86 CONTINUE
1204.                  OPPC= OPPC+OCC(I)*NV(K,I)*CC(I)/1000.
1205.              87 CONTINUE
1206.                  TREVE = RVC + VCAC
1207.                  TCSTS = TCC + OPPC
1208.                  RRENTC = TREVE - TCSTS
1209.                  WRITE(6,93)C,RVC,VCAC,TREVE,TCC,OPPC,TCSTS,RRENTC
1210.          93 FORMAT(' ',A4,1X,8F10.0)
1211.          VCAC=0.0
1212.      C.....COMBINED COSTS
1213.          WRITE(6,134)
1214.      134 FORMAT(' ',/,T28,'COMBINED COSTS')
1215.          WRITE(6,107)
1216.          WRITE(6,135)
1217.      135 FORMAT(' ',9X,'REV-F',5X,'REV-B',5X,'REV-T',7X,'TVC',8X,'FC',

```

```

1218.      18X,'DC',5X,'TCOST',6X,'RENT')
1219.      WRITE(6,107)
1220.      RV= RV+RVC
1221.      VCA=VCA+VCAC
1222.      TC= TC+TCC
1223.      OPP= OPP+OPPC
1224.      TREVE = RV + VCA
1225.      TCSTS = TC + OPP
1226.      RRENT= RRENT+RRENTC
1227.      WRITE(6,93)C,RV,VCA,TREVE,TC,FIX,OPP,TCSTS,RRENT
1228.      57 CONTINUE
1229.      600 CONTINUE
1230.      RETURN
1231.      END
1232.      C
1233.      C
1234.      SUBROUTINE DEMAND(PMCS,CCAT,J,SL,IRUN,A)
1235.      INTEGER TIME
1236.      DIMENSION Q(20)
1237.      COMMON/PARAM/TIME,NSP,ND,NC,NA,NVC,NSC,NM,NSX,NBC,CN
1238.      REAL PMCS(NSP,NSC,NM),CCAT(NSP,ND,NA,NVC,NSC,NM),SL(NSP,NSC),
1239.      +PBAR(8,12),XBAR(8,12),XLB(8,12),FLEX(10)
1240.      LOGICAL A(40)
1241.      DATA PBAR/96*0.0/
1242.      C      NOTE TO USER: THE ABOVE FIXED DIMENSIONED VARIABLES ARRAY
1243.      C      SIZES WILL NEED TO BE CHANGED IF THEIR ARRAY
1244.      C      SIZES ARE EXCEEDED. THE PARAMETERS FOR THE
1245.      C      VARIABLES ARE FLEX(NSC), XBAR(NSC,NM),
1246.      C      PBAR(NSC,NM), XLB(NSC,NM).
1247.      DO 1 K=1,NSC
1248.      XLB(K,J)=0.0
1249.      1 CONTINUE
1250.      IF((J.GT.1).OR.(IRUN.GT.2)) GO TO 10
1251.      READ(5,100)Q
1252.      IF(A(10))WRITE(6,1001)Q
1253.      100 FORMAT(20A4)
1254.      1001 FORMAT(' ',20A4)
1255.      READ(5,100)Q
1256.      IF(A(10))WRITE(6,1001)Q
1257.      READ(5,101)(FLEX(L),L=1,NSC)
1258.      IF(A(10))WRITE(6,101)(FLEX(L),L=1,NSC)
1259.      101 FORMAT(12F6.4)
1260.      READ(5,100)Q
1261.      IF(A(10))WRITE(6,1001)Q
1262.      READ(5,102)((XBAR(K,JJ),JJ=1,NM),K=1,NSC)
1263.      IF(A(10))WRITE(6,102)((XBAR(K,JJ),JJ=1,NM),K=1,NSC)
1264.      102 FORMAT(8F10.0)
1265.      DO 2 K=1,NSC
1266.      DO 2 JJ=1,NM
1267.      2 PBAR(K,JJ)=PMCS(1,K,JJ)
1268.      10 CONTINUE
1269.      DO 20 K=1,NSC
1270.      DO 20 L=1,NA
1271.      DO 20 M=1,NVC
1272.      DO 20 N=1,ND
1273.      DO 20 I=1,NSP
1274.      XLB(K,J)=XLB(K,J)+CCAT(I,N,L,M,K,J)
1275.      20 CONTINUE
1276.      DO 30 I=1,NSP
1277.      DO 30 K=1,NSC
1278.      PMCS(I,K,J)=PBAR(K,J)*(1.0-FLEX(K))*(XBAR(K,J)-XLB(K,J))

```

```

1279.      + /XBAR(K,J))
1280.      IF(A(31) .AND. I.EQ.1) WRITE(6,103)J,K,PMCS(I,K,J)
1281.      103 FORMAT(' MONTH=',I3,'      SIZE=',I2,'      PRICE=',F6.2)
1282.      30 CONTINUE
1283.      RETURN
1284.      END
1285.      C
1286.      C
1287.      SUBROUTINE TRID(IX,SP1,SP2,Y)
1288.      CALL RANDU(IX,IY,RN)
1289.      IX = IY
1290.      H= 2/(SP1+SP2)
1291.      Y= 2*RN/H
1292.      IF(Y.GT.SP1) GO TO 10
1293.      Y= -(SP1-(2*RN/H))
1294.      10 CONTINUE
1295.      RETURN
1296.      END
1297.      C
1298.      C
1299.      SUBROUTINE UNI(IX,SP1,SP2,Y)
1300.      CALL RANDU(IX,IY,RN)
1301.      IX = IY
1302.      Y= RN*(SP1+SP2)
1303.      IF(Y.LT.SP1) Y= -(SP1-Y)
1304.      IF(Y.GE.SP1) Y= Y-SP1
1305.      RETURN
1306.      END
1307.      C
1308.      C
1309.      SUBROUTINE GAUSS(IX,SP1,SP2,Y)
1310.      Z=0
1311.      DO 1 I=1,12
1312.          CALL RANDU(IX,IY,RN)
1313.          IX = IY
1314.          Z=Z+RN
1315.      1 CONTINUE
1316.      Z=(Z-6.0)/6.0
1317.      Y=SP1+SP2*Z
1318.      RETURN
1319.      END
1320.      C
1321.      C
1322.      SUBROUTINE SASOUT(TD, TM, TOT, TDM, TDS, CTH, RV,
1323.      1TREV, TCSTS, RRENT, TVALUE, A, IRUN, KCOUNT, COUNT, I, J, K, M, N)
1324.      COMMON/PARAM/TIME, NSP, ND, NC, NA, NVC, NSC, NM, NSX, NBC, CN
1325.      REAL TM(NM), TOT, TDM(ND, NM), TD(ND), TDS(ND, NSC)
1326.      REAL STM(2, 12), S2TM(2, 12), STOT(2), S2TOT(2)
1327.      REAL STD(2, 3, 12), S2TDM(2, 3, 12), STD(2, 3), S2TD(2, 3)
1328.      REAL STDS(2, 3, 8), S2TDS(2, 3, 8), SCTH(2, 2), S2CTH(2, 2)
1329.      REAL SRV(2, 2), S2RV(2, 2), STREVE(2, 2), S2TREV(2, 2)
1330.      REAL STCSTS(2, 2), S2TCST(2, 2), SRRENT(2, 2), S2RREN(2, 2)
1331.      INTEGER IRUN, KCOUNT, I, J, K, N
1332.      LOGICAL A(40)
1333.      DATA STM/24*0.0/, S2TM/24*0.0/, STOT/2*0.0/, S2TOT/2*0.0/
1334.      DATA STD/72*0.0/, S2TDM/72*0.0/, STD/6*0.0/, S2TD/6*0.0/
1335.      DATA STDS/48*0.0/, S2TDS/48*0.0/, SCTH/4*0.0/, S2CTH/4*0.0/
1336.      DATA SRV/4*0.0/, S2RV/4*0.0/, STREVE/4*0.0/, S2TREV/4*0.0/
1337.      DATA STCSTS/4*0.0/, S2TCST/4*0.0/, SRRENT/4*0.0/, S2RREN/4*0.0/
1338.      C      NOTE TO USER: THE ABOVE FIXED DIMENSIONED VARIABLES ARRAY
1339.      C      SIZES WILL NEED TO BE CHANGED IF THEIR ARRAY

```

```

1340.      C          SIZES ARE EXCEEDED. THE PARAMETERS FOR THE
1341.      C          VARIABLES ARE STM(NSP,NM), S2TM(NSP,NM),
1342.      C          STOT(NSP), S2TOT(NSP), STDM(NSP,ND,NM),
1343.      C          STD(NSP,ND), S2TD(NSP,ND), STDS(NSP,ND,NSC),
1344.      C          S2TDS(NSP,ND,NSC), SCTH(NSP,NVC), S2CTH(NSP,NVC),
1345.      C          SRV(NSP,NVC), S2RV(NSP,NVC), STREVE(NSP,NVC),
1346.      C          STCSTS(NSP,NVC), S2TCST(NSP,NVC), SRRENT(NSP,NVC),
1347.      C          S2RREN(NSP,NVC).
1348.          IF(KCOUNT.EQ.1) GO TO 1
1349.          IF(KCOUNT.EQ.2) GO TO 2
1350.          IF(KCOUNT.EQ.3) GO TO 3
1351.          IF(KCOUNT.EQ.4) GO TO 4
1352.          IF(KCOUNT.EQ.5) GO TO 5
1353.          IF(KCOUNT.EQ.6) GO TO 6
1354.      1 CONTINUE
1355.          STM(I,N)=STM(I,N) + TM(N)/1000.
1356.          S2TM(I,N)=S2TM(I,N)+(TM(N)/1000. )**2
1357.          IF(IRUN.LT.COUNT) GO TO 99
1358.          INN=COUNT-1
1359.          JNN=INN-1
1360.          S2TM(I,N)=(((S2TM(I,N)-((STM(I,N)**2)/INN))/JNN)**0.5)
1361.          STM(I,N)=STM(I,N)/INN
1362.          CIU=STM(I,N)+TVALUE*S2TM(I,N)
1363.          CIL=STM(I,N)-TVALUE*S2TM(I,N)
1364.          WRITE(8,101)I,N,STM(I,N),S2TM(I,N),CIL,CIU
1365.      101 FORMAT(1X,2I5,4F8.0)
1366.          GO TO 99
1367.      2 CONTINUE
1368.          STOT(I)=STOT(I) + TOT/1000.
1369.          S2TOT(I)=S2TOT(I)+(TOT/1000. )**2
1370.          IF(IRUN.LT.COUNT) GO TO 99
1371.          INN=COUNT-1
1372.          JNN=INN-1
1373.          S2TOT(I)=(((S2TOT(I)-((STOT(I)**2)/INN))/JNN)**0.5)
1374.          STOT(I)=STOT(I)/INN
1375.          CIU=STOT(I)+TVALUE*S2TOT(I)
1376.          CIL=STOT(I)-TVALUE*S2TOT(I)
1377.          WRITE(8,102)I,STOT(I),S2TOT(I),CIL,CIU
1378.      102 FORMAT(1X,I5,4F8.0)
1379.          GO TO 99
1380.      3 CONTINUE
1381.          STDM(I,J,N)=STDM(I,J,N) + TDM(J,N)/1000.
1382.          S2TDM(I,J,N)=S2TDM(I,J,N)+(TDM(J,N)/1000. )**2
1383.          IF(IRUN.LT.COUNT) GO TO 99
1384.          INN=COUNT-1
1385.          JNN=INN-1
1386.          S2TDM(I,J,N)=(((S2TDM(I,J,N)-((STDM(I,J,N)**2)/INN))/JNN)**0.5)
1387.          STDM(I,J,N)=STDM(I,J,N)/INN
1388.          CIU=STDM(I,J,N)+TVALUE*S2TDM(I,J,N)
1389.          CIL=STDM(I,J,N)-TVALUE*S2TDM(I,J,N)
1390.          WRITE(8,103)I,J,N,STDM(I,J,N),S2TDM(I,J,N),CIL,CIU
1391.      103 FORMAT(1X,3I5,4F8.0)
1392.          GO TO 99
1393.      4 CONTINUE
1394.          STD(I,J)=STD(I,J) + TD(J)/1000.
1395.          S2TD(I,J)=S2TD(I,J)+(TD(J)/1000. )**2
1396.          IF(IRUN.LT.COUNT) GO TO 99
1397.          INN=COUNT-1
1398.          JNN=INN-1
1399.          S2TD(I,J)=(((S2TD(I,J)-((STD(I,J)**2)/INN))/JNN)**0.5)
1400.          STD(I,J)=STD(I,J)/INN

```

```

1401.      CIU=STD(I,J)+TVALUE*S2TD(I,J)
1402.      CIL=STD(I,J)-TVALUE*S2TD(I,J)
1403.      WRITE(8,104)I,J,STD(I,J),S2TD(I,J),CIL,CIU
1404. 104  FORMAT(1X,2I5,4F8.0)
1405.      GO TO 99
1406.      5  CONTINUE
1407.      STDS(I,J,M)=STDS(I,J,M) + TDS(J,M)/1000.
1408.      S2TDS(I,J,M)=S2TDS(I,J,M)+(TDS(J,M)/1000. )**2
1409.      IF(IRUN.LT.COUNT) GO TO 99
1410.      INN=COUNT-1
1411.      JNN=INN-1
1412.      S2TDS(I,J,M)=(((S2TDS(I,J,M)-((STDS(I,J,M)**2)/INN))/JNN)**0.5)
1413.      STDS(I,J,M)=STDS(I,J,M)/INN
1414.      CIU=STDS(I,J,M)+TVALUE*S2TDS(I,J,M)
1415.      CIL=STDS(I,J,M)-TVALUE*S2TDS(I,J,M)
1416.      WRITE(8,105)I,J,M,STDS(I,J,M),S2TDS(I,J,M),CIL,CIU
1417. 105  FORMAT(1X,3I5,4F8.0)
1418.      GO TO 99
1419.      6  CONTINUE
1420.      SCTH(I,J)=SCTH(I,J) + CTH
1421.      S2CTH(I,J)=S2CTH(I,J)+(CTH)**2
1422.      IF(IRUN.LT.COUNT) GO TO 99
1423.      INN=COUNT-1
1424.      JNN=INN-1
1425.      S2CTH(I,J)=(((S2CTH(I,J)-((SCTH(I,J)**2)/INN))/JNN)**0.5)
1426.      SCTH(I,J)=SCTH(I,J)/INN
1427.      CIU=SCTH(I,J)+TVALUE*S2CTH(I,J)
1428.      CIL=SCTH(I,J)-TVALUE*S2CTH(I,J)
1429.      WRITE(8,106)I,J,SCTH(I,J),S2CTH(I,J),CIL,CIU
1430. 106  FORMAT(1X,2I5,4F8.0,' LANDINGS IN THOUSANDS')
1431.      SRV(I,J)=SRV(I,J) + RV
1432.      S2RV(I,J)=S2RV(I,J)+(RV)**2
1433.      S2RV(I,J)=(((S2RV(I,J)-((SRV(I,J)**2)/INN))/JNN)**0.5)
1434.      SRV(I,J)=SRV(I,J)/INN
1435.      CIU=SRV(I,J)+TVALUE*S2RV(I,J)
1436.      CIL=SRV(I,J)-TVALUE*S2RV(I,J)
1437.      WRITE(8,107)SRV(I,J),S2RV(I,J),CIL,CIU
1438. 107  FORMAT(1X,4F8.0,' SHRIMP REVENUE IN THOUSANDS')
1439.      STREVE(I,J)=STREVE(I,J) + TREVE
1440.      S2TREV(I,J)=S2TREV(I,J)+(TREVE)**2
1441.      S2TM(I,J)=(((S2TM(I,J)-((STM(I,J)**2)/INN))/JNN)**0.5)
1442.      STREVE(I,J)=STREVE(I,J)/INN
1443.      CIU=STREVE(I,J)+TVALUE*S2TREV(I,J)
1444.      CIL=STREVE(I,J)-TVALUE*S2TREV(I,J)
1445.      WRITE(8,108)STREVE(I,J),S2TREV(I,J),CIL,CIU
1446. 108  FORMAT(1X,4F8.0,' TOTAL REVENUE IN THOUSANDS')
1447.      STCSTS(I,J)=STCSTS(I,J) + TCSTS
1448.      S2TCST(I,J)=S2TCST(I,J)+(TCSTS)**2
1449.      S2TCST(I,J)=(((S2TCST(I,J)-((STCSTS(I,J)**2)/INN))/JNN)**0.5)
1450.      STCSTS(I,J)=STCSTS(I,J)/INN
1451.      CIU=STCSTS(I,J)+TVALUE*S2TCST(I,J)
1452.      CIL=STCSTS(I,J)-TVALUE*S2TCST(I,J)
1453.      WRITE(8,109)STCSTS(I,J),S2TCST(I,J),CIL,CIU
1454. 109  FORMAT(1X,4F8.0,' TOTAL COST IN THOUSANDS')
1455.      SRRENT(I,J)=SRRENT(I,J) + RRENT
1456.      S2RREN(I,J)=S2RREN(I,J)+(RRENT)**2
1457.      S2RREN(I,J)=(((S2RREN(I,J)-((SRRENT(I,J)**2)/INN))/JNN)**0.5)
1458.      SRRENT(I,J)=SRRENT(I,J)/INN
1459.      CIU=SRRENT(I,J)+TVALUE*S2RREN(I,J)
1460.      CIL=SRRENT(I,J)-TVALUE*S2RREN(I,J)
1461.      WRITE(8,110)SRRENT(I,J),S2RREN(I,J),CIL,CIU

```

```

1462.      110 FORMAT(11X,4F8.0,' RENT IN THOUSANDS')
1463.      99 CONTINUE
1464.      RETURN
1465.      END
1466.      SUBROUTINE INOUT(A,ITER,ISZ,SZCOF,GCOF,MSZC,CONV,C1,C2,COST,
1467.      1SL,COL,TM,E,ICOF,SZ,GRT,FMAX,ER
1468.      2,NUM,NMCOF,CCAT,BCAT,RCAT,FMORT,RFEF,RFP,CULL,DFN,DFNI,RENT,CS,
1469.      3OPC,VC,REV,PMCS,TDM,TDSM,TDSVM,TDSV,TDS,CPH,FTIME,
1470.      4PMCS,SP,VKIL,TYR,YR,NV,TCOST,CCOST,CF,CMESH,IC,CCT,CER,
1471.      5C3,C4,TDF,TDFN,XNV,BCATCH,PERF,VCATCH,
1472.      +NER,NPH,HP,FRL,TD,TREV,TRENT,TCS,C7,C8,ETM,IX,
1473.      +TOPC,TVC,FC,OCO,OCO,C,CVC,PC,SHARE,VESTO,PCATCH,TVALUE,
1474.      + PRNT,TAIL)
1475.      INTEGER TIME,FTIME
1476.      DIMENSION Q(20)
1477.      COMMON/COLSE/ KBD(18),KED(18),KBP(18),KEP(18),KAR(18),KDT(18),
1478.      + KSP(18)
1479.      C
1480.      COMMON/ADJUST/ ADJ(18)
1481.      C
1482.      COMMON/WEEK/ KK(4,18)
1483.      COMMON/PARAM/TIME,NSP,ND,NC,NA,NVC,NSC,NM,NSX,NBC,CN
1484.      COMMON/SFISH/BMESH,RMESH,BC,RC,C5,C6
1485.      INTEGER IX(5)
1486.      REAL ISZ(NSP,NSX),SZCOF(NSP,NSX),GCOF(NSP,NSX),MSZC(NSP,NSX)
1487.      REAL CONV(NSP),C1(NSP),C2(NSP),
1488.      1HP(NVC),FRL(NVC),TD(ND),TREV(NVC),TRENT(NVC),
1489.      2TCS(NVC),TOPC(NVC),TVC(NVC),SL(NSP,NSC),COL(NM),TM(NM),E(NS
1490.      3P,NER),ICOF(NSP,NA),SZ(NSP,NSX,NC),GRT(NSP,NPH),FMAX(NSP,ND,NA),
1491.      4ER(NSP,ND,NA,NA),NUM(NSP,NSX,ND,NC,NA),NMCOF(NSP,ND,NSC),
1492.      5CCAT(NSP,ND,NA,NVC,NSC,NM),BCAT(NSP,NA,NM)
1493.      REAL FMORT(NSP,ND,NVC,NA),RFEF(NSP,ND,NVC,NA),RFP(NSP,ND,NVC),CULL
1494.      1(NSP,ND,NA,NVC,NM),DFN(NSP,NA,ND,NVC,NM),RENT(NSP,NA,NM,NVC),CS(NS
1495.      2P,NA,NM,NVC),OPC(NSP,NA,NM,NVC),VC(NSP,NA,NM,NVC),REV(NSP,NA,NM,NV
1496.      3C),PMCS(NSP,NSC,NM),TDM(ND,NM),TDSM(ND,NSC,NM),
1497.      4TDSVM(ND,NSC,NVC,NM),TDSV(ND,NSC,NM),
1498.      5TDS(ND,NSC),PMCS(NSP,NSC,NM),YR(100)
1499.      REAL VKIL(NA,NSP,ND,NM),RCAT(NSP,NA,NM),NV(NA,NVC),FC(NVC),OCO(NVC
1500.      1),OCO(NVC),C(NVC),CVC(NVC),TCOST(NA,NSP,NVC),CCOST(NA,NSP,NVC),
1501.      2SP(10),DFNI(NSP,NA,ND,NVC,NM),PC(NVC),SHARE(NVC),COST(ND,NVC)
1502.      3,CF(ND,NVC),CMESH(NSP,ND),C3(NSP),C4(NSP),VESTO(NVC),TDF(NA,NVC),
1503.      4TDFN(NA,NSP,NVC),XNV(NA,NVC),BCATCH(NA,ND,NSP,NBC,NVC,NM),VCATCH
1504.      5(NA,ND,NSP,NBC,NVC,NM),PERF(NA,ND,NBC,NVC,NM),PCATCH(NBC)
1505.      REAL TYR(NC,NA,NSP,NVC),PRNT(NC)
1506.      LOGICAL A(40)
1507.      C
1508.      C...READ IN OPTIONS AVAILABLE
1509.      113 FORMAT(' NSP ND NC NA NVC NSC NM NER NPH NBC NSX'/' ',11I4)
1510.      READ(5,105) Q
1511.      105 FORMAT(20A4)
1512.      1051 FORMAT(' ',20A4)
1513.      1052 FORMAT('/' ',20A4)
1514.      READ(5,102) A
1515.      IF(A(10))WRITE(6,113)NSP,ND,NC,NA,NVC,NSC,NM,NER,NPH,NBC,NSX
1516.      IF(A(10)) WRITE(6,1051)Q
1517.      IF(A(10))WRITE(6,1021)A
1518.      102 FORMAT(20L4)
1519.      1021 FORMAT(' ',20L4/)
1520.      C...READ IN TIME PARAMETERS
1521.      READ(5,105)Q
1522.      IF(A(10)) WRITE(6,1052)Q

```

```

1523.      READ(5,105)Q
1524.      IF(A(10)) WRITE(6,1051)Q
1525.      READ(5,101) IC,ITER,FTIME,CCT,CN,CER,CPH,C8
1526.      IF(A(10)) WRITE(6,9011) IC,ITER,FTIME,CCT,CN,CER,CPH,C8
1527.      101 FORMAT(3I6,5F6.0)
1528.      9011 FORMAT(' ',3I6,4F6.1,F8.0)
1529.      C...READ IN RECRUITMENT AND MOVEMENT COEFFICIENTS
1530.      READ(5,105)Q
1531.      IF(A(10)) WRITE(6,1052)Q
1532.      READ(5,105)Q
1533.      IF(A(10)) WRITE(6,1051)Q
1534.      READ(5,106)((ICOF(I,J),J= 1,NA),I= 1,NSP)
1535.      IF(A(10))WRITE(6,1061)((ICOF(I,J),J= 1,NA),I= 1,NSP)
1536.      106 FORMAT(8E10.4)
1537.      1061 FORMAT(' ',8E12.4)
1538.      READ(5,105)Q
1539.      IF(A(10)) WRITE(6,1051)Q
1540.      READ(5,103)((E(I,J),J= 1,NER),I= 1,NSP)
1541.      IF(A(10))WRITE(6,1031)((E(I,J),J= 1,NER),I= 1,NSP)
1542.      READ(5,105)Q
1543.      IF(A(10)) WRITE(6,1051)Q
1544.      READ(5,103)((ISZ(I,J),J=1,NSX),I=1,NSP)
1545.      IF(A(10))WRITE(6,1031)((ISZ(I,J),J=1,NSX),I=1,NSP)
1546.      READ(5,105)Q
1547.      IF(A(10)) WRITE(6,1051)Q
1548.      READ(5,103)((ER(I,J,K,L),L= 1,NA),K= 1,NA),J= 1,ND),I= 1,NSP)
1549.      IF(A(10))WRITE(6,1031)((ER(I,J,K,L),L= 1,NA),K= 1,NA),J= 1,ND)
1550.      + ,I=1,NSP)
1551.      103 FORMAT(12F6.0)
1552.      READ(5,105)Q
1553.      IF(A(10)) WRITE(6,1051)Q
1554.      READ(5,103)((SZCOF(I,J),J=1,NSX),I=1,NSP)
1555.      IF(A(10))WRITE(6,1031)((SZCOF(I,J),J=1,NSX),I=1,NSP)
1556.      1031 FORMAT(' ',12F8.2)
1557.      1032 FORMAT(' ',12F6.2)
1558.      1033 FORMAT(' ',12F6.0)
1559.      1034 FORMAT(' ',12F8.4)
1560.      C...READ IN GROWTH COEFFICIENTS
1561.      READ(5,105)Q
1562.      IF(A(10)) WRITE(6,1052)Q
1563.      READ(5,105)Q
1564.      IF(A(10)) WRITE(6,1051)Q
1565.      READ(5,104)((GCOF(I,J),J=1,NSX),I=1,NSP)
1566.      IF(A(10))WRITE(6,1043)((GCOF(I,J),J=1,NSX),I=1,NSP)
1567.      READ(5,105)Q
1568.      IF(A(10)) WRITE(6,1051)Q
1569.      READ(5,104)((MSZC(I,J),J=1,NSX),I=1,NSP)
1570.      IF(A(10))WRITE(6,1041)((MSZC(I,J),J=1,NSX),I=1,NSP)
1571.      READ(5,105)Q
1572.      IF(A(10)) WRITE(6,1051)Q
1573.      READ(5,103)((GRT(I,J),J= 1,NPH),I= 1,NSP)
1574.      IF(A(10))WRITE(6,1031)((GRT(I,J),J= 1,NPH),I= 1,NSP)
1575.      READ(5,105)Q
1576.      IF(A(10)) WRITE(6,1051)Q
1577.      READ(5,104)(C3(I),I=1,NSP)
1578.      IF(A(10))WRITE(6,1042)(C3(I),I=1,NSP)
1579.      READ(5,105)Q
1580.      IF(A(10)) WRITE(6,1051)Q
1581.      READ(5,103)(C4(I),I=1,NSP)
1582.      IF(A(10))WRITE(6,1031)(C4(I),I=1,NSP)
1583.      C...READ IN NATURAL MORTALITY COEFFICIENTS

```

```

1584.      READ(5,105)Q
1585.      IF(A(10)) WRITE(6,1052)Q
1586.      READ(5,103)((NMCOF(I,J,K),K= 1,NSC),J= 1,ND),I= 1,NSP)
1587.      IF(A(10))WRITE(6,1031)((NMCOF(I,J,K),K= 1,NSC),J= 1,ND),I= 1,NSP)
1588. C...READ IN HARVEST COEFFICIENTS
1589.      READ(5,105)Q
1590.      IF(A(10)) WRITE(6,1052)Q
1591.      READ(5,105)Q
1592.      IF(A(10)) WRITE(6,1051)Q
1593.      READ(5,103)BMESH,RMESH,BC,RC,ETM,C5,C6,C7
1594.      IF(A(10))WRITE(6,1031)BMESH,RMESH,BC,RC,ETM,C5,C6,C7
1595.      READ(5,105)Q
1596.      IF(A(10)) WRITE(6,1051)Q
1597.      READ(5,103)((CMESH(I,J),J=1,ND),I=1,NSP)
1598.      IF(A(10))WRITE(6,1031)((CMESH(I,J),J=1,ND),I=1,NSP)
1599.      READ(5,105)Q
1600.      IF(A(10)) WRITE(6,1051)Q
1601.      READ(5,103)((CF(I,J),I= 1,ND),J= 1,NVC)
1602.      IF(A(10))WRITE(6,1031)((CF(I,J),I= 1,ND),J= 1,NVC)
1603.      READ(5,105)Q
1604.      IF(A(10)) WRITE(6,1051)Q
1605.      READ(5,103)((SL(I,J),J= 1,NSC),I= 1,NSP)
1606.      IF(A(10))WRITE(6,1031)((SL(I,J),J= 1,NSC),I= 1,NSP)
1607.      READ(5,105)Q
1608.      IF(A(10)) WRITE(6,1051)Q
1609.      READ(5,104)((FMAX(I,J,K),K= 1,NA),J= 1,ND),I= 1,NSP)
1610.      IF(A(10))WRITE(6,1042)((FMAX(I,J,K),K= 1,NA),J= 1,ND),I= 1,NSP)
1611.      READ(5,105)Q
1612.      IF(A(10)) WRITE(6,1051)Q
1613.      READ(5,103)((VKIL(L,I,J,K),K= 1,NM),J= 1,ND),I= 1,NSP),L=1,NA)
1614.      IF(A(10))WRITE(6,1031)((VKIL(L,I,J,K),K= 1,NM),J= 1,ND),I= 1,NSP
1615.      1),L=1,NA)
1616.      READ(5,105)Q
1617.      IF(A(10)) WRITE(6,1051)Q
1618.      READ(5,103)((PERF(L,M,I,J,K),K=1,NM),J=1,NVC),I=1,NBC),M=1,ND)
1619.      1,L=1,NA)
1620.      IF(A(10))WRITE(6,1033)((PERF(L,M,I,J,K),K=1,NM),J=1,NVC)
1621.      1,I=1,NBC),M=1,ND),L=1,NA)
1622.      104 FORMAT(10F8.0)
1623.      1041 FORMAT(' ',10F10.2)
1624.      1042 FORMAT(' ',10F12.8)
1625.      1043 FORMAT(' ',10F10.4)
1626. C...READ IN RELATIVE FISHING POWER COEFFICIENTS
1627.      READ(5,105)Q
1628.      IF(A(10)) WRITE(6,1052)Q
1629.      READ(5,105)Q
1630.      IF(A(10)) WRITE(6,1051)Q
1631.      READ(5,103)(FRL(I),I=1,NVC)
1632.      IF(A(10))WRITE(6,1031)(FRL(I),I=1,NVC)
1633.      READ(5,105)Q
1634.      IF(A(10)) WRITE(6,1051)Q
1635.      READ(5,103)(HP(I),I=1,NVC)
1636.      IF(A(10))WRITE(6,1031)(HP(I),I=1,NVC)
1637.      READ(5,105)Q
1638.      IF(A(10)) WRITE(6,1051)Q
1639.      READ(5,103)(C1(I),I=1,NSP)
1640.      IF(A(10))WRITE(6,1034)(C1(I),I=1,NSP)
1641.      READ(5,105)Q
1642.      IF(A(10)) WRITE(6,1051)Q
1643.      READ(5,103)(C2(I),I=1,NSP)
1644.      IF(A(10))WRITE(6,1034)(C2(I),I=1,NSP)

```



```

1645. C...READ IN ECONOMIC COEFFICIENTS
1646. READ(5,105)Q
1647. IF(A(10)) WRITE(6,1052)Q
1648. READ(5,105)Q
1649. IF(A(10)) WRITE(6,1051)Q
1650. READ(5,103)(CONV(I),I=1,NSP)
1651. IF(A(10))WRITE(6,1031)(CONV(I),I=1,NSP)
1652. READ(5,105)Q
1653. IF(A(10)) WRITE(6,1051)Q
1654. READ(5,103)((XNV(J,I),I=1,NVC),J=1,NA)
1655. IF(A(10))WRITE(6,1031)((XNV(J,I),I=1,NVC),J=1,NA)
1656. READ(5,105)Q
1657. IF(A(10)) WRITE(6,1051)Q
1658. READ(5,103)(C(I),I=1,NVC)
1659. IF(A(10))WRITE(6,1031)(C(I),I=1,NVC)
1660. READ(5,105)Q
1661. IF(A(10)) WRITE(6,1051)Q
1662. READ(5,103)(FC(I),I=1,NVC)
1663. IF(A(10))WRITE(6,1031)(FC(I),I=1,NVC)
1664. READ(5,105)Q
1665. IF(A(10)) WRITE(6,1051)Q
1666. READ(5,103)((COST(I,J),I= 1,ND),J= 1,NVC)
1667. IF(A(10))WRITE(6,1031)((COST(I,J),I= 1,ND),J= 1,NVC)
1668. READ(5,105)Q
1669. IF(A(10)) WRITE(6,1051)Q
1670. READ(5,103)(CVC(I),I=1,NVC)
1671. IF(A(10))WRITE(6,1031)(CVC(I),I=1,NVC)
1672. READ(5,105)Q
1673. IF(A(10)) WRITE(6,1051)Q
1674. READ(5,103)(OCO(I),I=1,NVC)
1675. IF(A(10))WRITE(6,1031)(OCO(I),I=1,NVC)
1676. READ(5,105)Q
1677. IF(A(10)) WRITE(6,1051)Q
1678. READ(5,103)(DCC(I),I=1,NVC)
1679. IF(A(10))WRITE(6,1031)(DCC(I),I=1,NVC)
1680. READ(5,105)Q
1681. IF(A(10)) WRITE(6,1051)Q
1682. READ(5,103)(PC(I),I=1,NVC)
1683. IF(A(10))WRITE(6,1031)(PC(I),I=1,NVC)
1684. READ(5,105)Q
1685. IF(A(10)) WRITE(6,1051)Q
1686. READ(5,103)(SHARE(I),I=1,NVC)
1687. IF(A(10))WRITE(6,1031)(SHARE(I),I=1,NVC)
1688. READ(5,105)Q
1689. IF(A(10)) WRITE(6,1051)Q
1690. READ(5,103)((PM CSP(I,J,K),K= 1,NM),J= 1,NSC),I= 1,NSP)
1691. IF(A(10))WRITE(6,1031)((PM CSP(I,J,K),K= 1,NM),J= 1,NSC),I= 1,NSP)
1692. READ(5,105)Q
1693. IF(A(10)) WRITE(6,1051)Q
1694. READ(5,103)(PCATCH(I),I=1,NBC)
1695. IF(A(10))WRITE(6,1031)(PCATCH(I),I=1,NBC)
1696. C...READ IN ADJUSTMENT FACTORS FOR EFFORT
1697. READ(5,105)Q
1698. IF(A(10)) WRITE(6,1052)Q
1699. READ(5,105)Q
1700. IF(A(10)) WRITE(6,1051)Q
1701. READ(5,108)(YR(I),I= 1,ITER)
1702. IF(A(10))WRITE(6,1081)(YR(I),I= 1,ITER)
1703. READ(5,105)Q
1704. IF(A(10)) WRITE(6,1051)Q
1705. READ(5,108)((TYR(K,L,I,J),J= 1,NVC),I= 1,NSP),L=1,NA),K=1,ITER)

```

```

1706.         IF(A(10))WRITE(6,1041)((((TYR(K,L,I,J),J= 1,NVC),I= 1,NSP)
1707.         1,L=1,NA),K=1,ITER)
1708.         108 FORMAT(20F4.0)
1709.         1081 FORMAT(20F5.1)
1710.         READ(5,105)Q
1711.         IF(A(10))WRITE(6,1051)Q
1712.         READ(5,112)(PRNT(I),I=1,ITER)
1713.         IF(A(10))WRITE(6,1120)(PRNT(I),I=1,ITER)
1714.         112 FORMAT(80F1.0)
1715.         1120 FORMAT(' ',40F2.0)
1716.         C...READ IN CLOSURE PARAMETERS
1717.         READ(5,105)Q
1718.         IF(A(10)) WRITE(6,1052)Q
1719.         READ(5,105)Q
1720.         IF(A(10)) WRITE(6,1051)Q
1721.         IF (IC.EQ.0) GO TO 1000
1722.         DO 1001 IZZ=1,IC
1723.             IZ=IC+1-IZZ
1724.             READ(5,111) KBD(IZ),KED(IZ),KBP(IZ),KEP(IZ),KAR(IZ),KDT(IZ),
1725.             +         KSP(IZ),ADJ(IZ)
1726.             IF (A(10)) WRITE(6,1111) KBD(IZ),KED(IZ),KBP(IZ),KEP(IZ),
1727.             +         KAR(IZ),KDT(IZ),KSP(IZ),ADJ(IZ)
1728.         1001 CONTINUE
1729.         1000 CONTINUE
1730.         111 FORMAT(7I5,F5.2)
1731.         1111 FORMAT(' ',7I5,F6.2)
1732.         C...READ IN LABELS
1733.         READ(5,105)Q
1734.         IF(A(10)) WRITE(6,1052)Q
1735.         READ(5,105)(COL(I),I= 1,NM)
1736.         IF(A(10))WRITE(6,1051)(COL(I),I= 1,NM)
1737.         C...READ IN STOCHASTIC PARAMETERS
1738.         READ(5,105)Q
1739.         IF(A(10)) WRITE(6,1052)Q
1740.         READ(5,105)Q
1741.         IF(A(10)) WRITE(6,1051)Q
1742.         READ(5,109)(IX(I),I=1,5)
1743.         IF(A(10))WRITE(6,1091)(IX(I),I=1,5)
1744.         READ(5,105)Q
1745.         IF(A(10)) WRITE(6,1051)Q
1746.         READ(5,103)(SP(I),I=1,10)
1747.         IF(A(10))WRITE(6,1032)(SP(I),I=1,10)
1748.         READ(5,105)Q
1749.         IF(A(10))WRITE(6,1052)Q
1750.         READ(5,103)TVALUE
1751.         IF(A(10))WRITE(6,1034)TVALUE
1752.         109 FORMAT(5I6)
1753.         1091 FORMAT(' ',5I6)
1754.         C...READ IN NOMINAL DAYS FISHED
1755.         READ(5,105)Q
1756.         IF(A(10)) WRITE(6,1052)Q
1757.         DO 110 M= 1,NSP
1758.             READ(5,103)((((DFNI(M,L,I,K,J),J= 1,NM),K= 1,NVC),
1759.             +         I=1,ND),L=1,NA)
1760.             IF(A(10))WRITE(6,1033)((((DFNI(M,L,I,K,J),J= 1,NM),
1761.             +         K=1,NVC),I=1,ND),L= 1,NA)
1762.         110 CONTINUE
1763.         RETURN
1764.         END
1765.         C
1766.         SUBROUTINE EFFORT(ITER,COUNT,A,NV,XNV,YR,IRUN,TYR,DFN,DFNI)

```

```

1767.      COMMON/PARAM/TIME, NSP, ND, NC, NA, NVC, NSC, NM, NSX, NBC, CN
1768.      C
1769.      REAL NV(NA, NVC), XNV(NA, NVC), YR(100), TYR(NC, NA, NSP, NVC),
1770.      +      DFN(NSP, NA, ND, NVC, NM), DFNI(NSP, NA, ND, NVC, NM)
1771.      LOGICAL A(40)
1772.      C
1773.      IF((ITER.NE.COUNT).AND.(.NOT.A(19))) RETURN
1774.      DO 193 L= 1, NVC
1775.          DO 191 J= 1, NA
1776.              NV(J, L) = XNV(J, L)
1777.              IF(A(26)) NV(J, L) = XNV(J, L)*YR(IRUN)
1778.              DO 190 I= 1, NSP
1779.                  IF(I.EQ.1 .AND. (A(27)))
1780.                      +      NV(J, L)=NV(J, L)*TYR(IRUN, J, I, L)
1781.                      DO 192 K= 1, ND
1782.                          DO 194 M= 1, NM
1783.                              DFN(I, J, K, L, M) =
1784.                              +      DFNI(I, J, K, L, M)*TYR(IRUN, J, I, L)*YR(IRUN)
1785.          194      CONTINUE
1786.      192      CONTINUE
1787.      190      CONTINUE
1788.      191      CONTINUE
1789.      193      CONTINUE
1790.      RETURN
1791.      C
1792.      END
1793.      C
1794.      SUBROUTINE INTIAL(TDFN, TCOST, CCOST, PMCS, PMCSP, BCATCH, VCATCH, RENT,
1795.      +      CS, VC, OPC, REV, FMORT, RFEF, RCAT, BCAT, CULL, CCAT, IC,
1796.      +      TDF, NU)
1797.      COMMON/PARAM/TIME, NSP, ND, NC, NA, NVC, NSC, NM, NSX, NBC, CN
1798.      COMMON/WEEK/ KK(4, 18)
1799.      C
1800.      REAL TDFN(NA, NSP, NVC), TCOST(NA, NSP, NVC), CCOST(NA, NSP, NVC),
1801.      +      PMCS(NSP, NSC, NM), PMCSP(NSP, NSC, NM), BCATCH(NA, ND, NSP, NBC,
1802.      +      NVC, NM), VCATCH(NA, ND, NSP, NBC, NVC, NM), RENT(NSP, NA, NM, NVC),
1803.      +      CS(NSP, NA, NM, NVC), VC(NSP, NA, NM, NVC), OPC(NSP, NA, NM, NVC),
1804.      +      REV(NSP, NA, NM, NVC), FMORT(NSP, ND, NVC, NA), RFEF(NSP, ND, NVC, NA),
1805.      +      RCAT(NSP, NA, NM), BCAT(NSP, NA, NM), CULL(NSP, ND, NA, NVC, NM),
1806.      +      CCAT(NSP, ND, NA, NVC, NSC, NM), TDF(NA, NVC), NU(5)
1807.      C
1808.      DO 201 I= 1, NSP
1809.      215      CONTINUE
1810.          DO 218 J= 1, NVC
1811.              DO 1218 L=1, NA
1812.                  TDFN(L, I, J) = 0.0
1813.                  TCOST(L, I, J)= 0.0
1814.                  CCOST(L, I, J)= 0.0
1815.      1218      CONTINUE
1816.      218      CONTINUE
1817.          DO 202 J= 1, NM
1818.              DO 271 L= 1, NSC
1819.                  PMCS(I, L, J)= PMCSP(I, L, J)
1820.      271      CONTINUE
1821.          DO 203 K= 1, NVC
1822.              DO 219 L=1, NA
1823.                  DO 1219 J3=1, ND
1824.                      DO 2219 J2=1, NBC
1825.                          BCATCH(L, J3, I, J2, K, J)=0.0
1826.                          VCATCH(L, J3, I, J2, K, J)=0.0
1827.      2219      CONTINUE

```

```

1828.      1219      CONTINUE
1829.      219      CONTINUE
1830.      DO 213 M= 1,NA
1831.          RENT(I,M,J,K)= 0
1832.          CS(I,M,J,K)= 0
1833.          VC(I,M,J,K)= 0
1834.          OPC(I,M,J,K)= 0
1835.          REV(I,M,J,K)= 0
1836.      213      CONTINUE
1837.      203      CONTINUE
1838.      202      CONTINUE
1839.          DO 205 J= 1,ND
1840.              DO 207 L= 1,NA
1841.                  DO 206 K= 1,NVC
1842.                      FMORT(I,J,K,L)= 0
1843.                      RFEF(I,J,K,L)= 0
1844.      206      CONTINUE
1845.                  DO 210 M= 1,NM
1846.                      RCAT(I,L,M)= 0
1847.                      BCAT(I,L,M)= 0
1848.                      DO 211 N1= 1,NVC
1849.                          CULL(I,J,L,N1,M)= 0
1850.                          DO 212 N= 1,NSC
1851.                              CCAT(I,J,L,N1,N,M)= 0
1852.      212      CONTINUE
1853.      211      CONTINUE
1854.      210      CONTINUE
1855.      207      CONTINUE
1856.      205      CONTINUE
1857.      201 CONTINUE
1858.  C
1859.  C
1860.      IF (IC.EQ.0) GO TO 2000
1861.      DO 2001 IZZ=1,IC
1862.          IZ=IC+1-IZZ
1863.          DO 2002 IZY=1,4
1864.              KK(IZY,IZ)=1
1865.      2002      CONTINUE
1866.      2001 CONTINUE
1867.      2000 CONTINUE
1868.          DO 550 I=1,NVC
1869.              DO 500 L=1,NA
1870.                  TDF(L,I) = 0.0
1871.      500      CONTINUE
1872.      550 CONTINUE
1873.          DO 216 I= 1,5
1874.              NU(I)= 0.0
1875.      216 CONTINUE
1876.      RETURN
1877.      END
1878.  C
1879.      SUBROUTINE NDLOOP(IRUN,A,SZ,NUM)
1880.  C
1881.      COMMON/PARAM/TIME,NSP,ND,NC,NA,NVC,NSC,NM,NSX,NBC,CN
1882.  C
1883.      REAL SZ(NSP,NSX,NC),NUM(NSP,NSX,ND,NC,NA)
1884.      LOGICAL A(40)
1885.  C
1886.      DO 403 I= 1,NSP
1887.          DO 404 J= 1,ND
1888.              DO 405 L= 1,NA

```

```

1889.          DO 408 J7=1,NSX
1890.          WRITE(6,406)IRUN,I,J,L,J7
1891.          406      FORMAT('O','ITERATION ',I3,3X,'SPECIES ',
1892.          +          I2,3X,'DEPTH ',I2,3X,'AREA ',I2,3X,'SEX',
1893.          +          I2/,'O',T8,'COHORT',3X,'NUMBERS',14X,'SIZE')
1894.          WRITE(6,407)(K,NUM(I,J7,J,K,L),SZ(I,J7,K),
1895.          +          K= 1,NC)
1896.          407      FORMAT(' ',T12,I3,2X,F10.0,10X,F10.6)
1897.          408      CONTINUE
1898.          405      CONTINUE
1899.          404      CONTINUE
1900.          403      CONTINUE
1901.          RETURN
1902.          END
1903.      C
1904.          SUBROUTINE STOCHA(A,SP,IX,NU)
1905.      C
1906.          REAL SP(10),NU(5)
1907.          INTEGER IX(5)
1908.          LOGICAL A(40)
1909.      C
1910.          DO 302 I= 1,5
1911.              IF(.NOT.A(I+13)) GO TO 303
1912.              J= 2*I
1913.              K= 2*I-1
1914.              IF(A(11)) CALL UNI(IX(I),SP(K),SP(J),NU(I))
1915.              IF(A(12)) CALL TRID(IX(I),SP(K),SP(J),NU(I))
1916.              IF(A(13)) CALL GAUSS(IX(I),SP(K),SP(J),NU(I))
1917.          303      CONTINUE
1918.          302      CONTINUE
1919.          RETURN
1920.          END

```

Appendix D - JCL

Set 1 - JCL to create a load moduls from SBUILD called NEW(LMNBUILD).

```
1. //JCLBUILD JOB (L679,004D,02,003,CB), 'GRIFFIN',MSGLEVEL=(2,0)
2. //*LEVEL 1
3. // EXEC COMPRESS,LIBRARY='USR.L679.CB.NEW'
4. // EXEC FORTXCL,REGION=256K
5. //FORT.SYSIN DD DSN=USR.L679.CB.SBUILD,DISP=SHR
6. //LKED.SYSLMOD DD DSN=USR.L679.CB.NEW(LMNBUILD),DISP=OLD
7. //*END
```

Set 2 - JCL to create a load moduls from SORIGIN called NEW(LMSYSTEM).

```
1. //JCLSETUP JOB (L679,004D,02,003,CB), 'GRIFFIN',MSGLEVEL=(2,0)
2. //*LEVEL 1
3. // EXEC FORTXCL,REGION=256K
4. //FORT.SYSIN DD DSN=USR.L679.CB.SORIGIN,DISP=OLD
5. //LKED.SYSLMOD DD DSN=USR.L679.CB.NEW(LMNSYSTEM),DISP=OLD
6. //*END
```

Set 3 - JCL that accesses NEW(LMNBUILD) and is used to set the array size for the model parameters and to change the array sizes when needed.

```
1. //JCLMODEL JOB (L679,004D,02,003,CB), 'GRIFFIN',MSGLEVEL=(2,0)
2. //*LEVEL 1
3. // EXEC COMPRESS,LIBRARY='USR.L679.CB.NEW'
4. // EXEC PGM=LMNBUILD
5. //STEPLIB DD DSN=USR.L679.CB.NEW,DISP=SHR
6. //FTO6FOO1 DD SYSOUT=A
7. //FTO7FOO1 DD DSN=&&DRIVER,SPACE=(TRK,(10,10)),DISP=(NEW,PASS),
8. // DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160),UNIT=SYSDA
9. //FTO5FOO1 DD *
10. 2 3 96 1 2 8 12 48 12 1 2
11. // EXEC FORTXCL
12. //FORT.SYSIN DD DSN=&&DRIVER,DISP=OLD
13. //LKED.SYSLMOD DD DSN=USR.L679.CB.NEW(LMNSYSTEM),DISP=OLD
14. //LKED.OLDLIB DD DSN=USR.L679.CB.NEW,DISP=SHR
15. //LKED.SYSIN DD *
16. INCLUDE OLDLIB(LMNSYSTEM)
17. ENTRY MAIN
18. //*END
```

Set 4 - JCL that accesses NEW(LMNSYSTEM) and uses a specific data input stream (Similar to SIMULAT1 or SIMULAT2) for a simulation run.

```
1. //JCLSIM JOB (L679,004D,02,003,CB), 'GRIFFIN',MSGLEVEL=(2,0)
2. //*LEVEL 1
3. //*FORMAT PR,DDNAME=,DEST=XEROX,FORMS=1101
4. // EXEC PGM=LMNSYSTEM,REGION=256K
5. //STEPLIB DD DSNAME=USR.L679.CB.NEW,DISP=SHR
6. //FTO6FOO1 DD SYSOUT=A
7. //FTO8FOO1 DD DSN=USR.L679.CB.SASOUT,DISP=OLD
8. //FTO5FOO1 DD DSN=USR.L679.CB.SIMULAT1,DISP=SHR
9. //*END
```



Set 5 - JCL to modify any subroutine in the load moduls NEW(LMNSYSTEM)

```
1. //JCLCHBIN JOB (L679,004A,002,15,CB), 'GRIFFIN'  
2. // EXEC COMPRESS, LIBRARY='USR.L679.CB.NEW'  
3. // EXEC FORTXCL, FXREGN=320K  
4. //FORT.SYSIN DD DSN=USR.L679.CB.CHBIN, DISP=SHR  
5. //LKED.SYSLMOD DD DSN=USR.L679.CB.NEW(LMNSYSTEM),  
6. // DISP=OLD  
7. //LKED.OLDLOAD DD DSN=USR.L679.CB.NEW, DISP=OLD  
8. //LKED.SYSIN DD *  
9. INCLUDE OLDLOAD(LMNSYSTEM)  
10. ENTRY MAIN  
11. //*END
```

Appendix E - SIMULATE

SIMULAT1

FOR THIS DATA STREAM NSP=2, ND=3, NC=96, NA=1, NVC=2, NSC=8, NM=12,  
NER=48, NPH=12, NBC=1, NSX=2.

```

1.      A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17 A18 A19 A20
2.      F  F  F  T  T  T  F  T  F  T  F  F  F  F  F  F  F  F  F  F
3.      T  T  T  T  T  F  F  F  F  F  F  T
4.      ***** TIME PARAMETERS *****
5.      IC ITER FTIME CCT CN CER CPR CR
6.      4 2 48 1.0 4.01 1.01 4.01 1000
7.      ***** RECRUITMENT AND MOVEMENT COEFFICIENTS *****
8.      ICOF(NSP,NA)
9.      5.500E 8 2.500E 8
10.     E(NSP,NER)
11.     .03 .03 .05 .08 .10 .12 .13 .16 .19 .22 .25 .28
12.     .35 .38 .40 .40 .40 .40 .37 .33 .30 .30 .30 .30
13.     .05 .04 .03 .03 .03 .03 .03 .03 .03 .03 .03 .03
14.     .03 .03 .03 .03 .02 .02 .02 .01 .01 .01 .01 .01
15.     .03 .03 .03 .03 .05 .05 .05 .05 .23 .23 .23 .23
16.     .23 .23 .23 .23 .25 .25 .25 .25 .25 .25 .25 .25
17.     .35 .35 .35 .35 .35 .35 .35 .35 .35 .35 .35 .35
18.     .22 .22 .22 .22 .15 .15 .15 .15 .15 .15 .15 .15
19.     ISZ(NSP,NSX)
20.     25. 25. 25. 25.
21.     ER(NSP,ND,NA,NA)
22.     .44 .28 .08 .08 .05 .00
23.     SZCOF(NSP,NSX)
24.     50. 50. 50. 50.
25.     ***** GROWTH COEFFICIENTS *****
26.     GCOF(NSP,NSX)
27.     .072 .080 .0739 .0739
28.     MSZC(NSP,NSX)
29.     97.56 113.93 115. 115.
30.     GRT(NSP,NPH)
31.     1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
32.     1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
33.     C3(NSP)
34.     .0000234.0000092
35.     C4(NSP)
36.     2.9991 3.196
37.     ***** NATURAL MORTALITY COEFFICIENTS ***** ;NMCOF(NSP,ND,NSC)
38.     .0408 .0408 .0408 .0408 .0408 .0408 .0408 .0408 .0408 .0408 .0408 .0408
39.     .0408 .0408 .0408 .0408 .0408 .0408 .0408 .0408 .0408 .0408 .0408 .0408
40.     .0794 .0794 .0794 .0794 .0794 .0794 .0794 .0794 .0794 .0794 .0794 .0794
41.     .0794 .0794 .0794 .0794 .0794 .0794 .0794 .0794 .0794 .0794 .0794 .0794
42.     ***** HARVEST COEFFICIENTS *****
43.     BMESH RMESH BC RC EMT C5 C6 C7
44.     45. 45. .0020 .1315 453.6 68.4 45. 1.0
45.     CMESH(NSP,ND)
46.     45. 45. 45. 60. 60. 60.
47.     CF(NVC,ND)
48.     .95 .95 .95 .50 .50 .50
49.     SL(NSP,NSC)
50.     108.9 99.0 91.9 86.5 78.6 72.9 65.8 45.0 109.8 100.3 93.6 88.4
51.     80.8 75.3 68.4 45.0

```

52.	FMAX(NSP,ND,NA)											
53.	.000192 .000081 .000031 .000120 .000040 .000071											
54.	VKIL(NSP,ND,NM)											
55.	67.6 67.6 67.6 67.6 45.0 45.0 45.0 67.6 67.6 67.6 67.6 67.6											
56.	67.6 67.6 67.6 67.6 67.6 67.6 67.6 67.6 67.6 67.6 67.6 67.6											
57.	67.6 67.6 67.6 67.6 67.6 67.6 67.6 67.6 67.6 67.6 67.6 67.6											
58.	70.2 70.2 70.2 70.2 45.0 45.0 45.0 70.2 70.2 70.2 70.2 70.2											
59.	70.2 70.2 70.2 70.2 70.2 70.2 70.2 70.2 70.2 70.2 70.2 70.2											
60.	70.2 70.2 70.2 70.2 70.2 70.2 70.2 70.2 70.2 70.2 70.2 70.2											
61.	PERF(NA,ND,NBC,NVC,NM)											
62.	O											
63.	O											
64.	O											
65.	O											
66.	O											
67.	O											
68.	***** RELATIVE FISHING POWER COEFFICIENTS *****											
69.	FRL(NVC)											
70.	25. 40.											
71.	HP(NVC)											
72.	150. 300.											
73.	C1(NSP)											
74.	.1776 .1377											
75.	C2(NSP)											
76.	.3906 .2261											
77.	***** ECONOMICS COEFFICIENTS *****											
78.	CONV(SNSP)											
79.	1.0 1.0											
80.	NV(NA,NVC)											
81.	500 1169											
82.	C(NVC)											
83.	1. 2.											
84.	FC(NVC)											
85.	7602 30983											
86.	COST(NVC,ND)											
87.	419 419 419 915 915 915											
88.	CVC(NVC)											
89.	0 0											
90.	OCO(NVC)											
91.	10541 13442											
92.	OCC(NVC)											
93.	7621 10701											
94.	PC(NVC)											
95.	0.0 0.08											
96.	SHARE(NVC)											
97.	.20 .20											
98.	PMCS( NSP, NSC, NM)											
99.	6.36 6.50 6.50 6.46 6.56 6.60 5.89 5.65 5.88 5.89 5.54 5.33											
100.	4.91 5.17 5.37 5.60 5.79 5.93 5.66 5.43 5.58 5.56 5.28 4.92											
101.	4.33 4.58 4.78 5.04 5.21 5.46 5.30 5.09 5.17 5.04 4.71 4.70											
102.	4.12 4.38 4.60 4.72 4.98 5.26 5.00 4.74 4.70 4.81 4.61 4.68											
103.	3.63 3.88 4.03 4.22 4.58 4.53 4.17 3.79 3.87 4.08 3.96 3.90											
104.	3.11 3.34 3.49 3.48 3.83 3.53 3.21 3.06 3.18 3.34 3.38 3.23											
105.	2.50 2.72 2.75 2.77 3.05 2.75 2.48 2.36 2.55 2.50 2.40 2.58											
106.	0.75 0.90 1.29 1.87 1.13 1.34 1.43 1.76 1.51 1.40 1.32 1.29											
107.	6.36 6.50 6.50 6.46 6.56 6.60 5.89 5.65 5.88 5.89 5.54 5.33											
108.	4.91 5.17 5.37 5.60 5.79 5.93 5.66 5.43 5.58 5.56 5.28 4.92											
109.	4.33 4.58 4.78 5.04 5.21 5.46 5.30 5.09 5.17 5.04 4.71 4.70											
110.	4.12 4.38 4.60 4.72 4.98 5.26 5.00 4.74 4.70 4.81 4.61 4.68											
111.	3.63 3.88 4.03 4.22 4.58 4.53 4.17 3.79 3.87 4.08 3.96 3.90											
112.	3.11 3.34 3.49 3.48 3.83 3.53 3.21 3.06 3.18 3.34 3.38 3.23											

```

113.      2.50  2.72  2.75  2.77  3.05  2.75  2.48  2.36  2.55  2.50  2.40  2.58
114.      0.75  0.90  1.29  1.87  1.13  1.34  1.43  1.76  1.51  1.40  1.32  1.29
115. PCATCH(NBC)
116. 0
117. ***** ADJUSTMENT FACTORS FOR EFFORT *****
118. YR(20)
119. 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
120. TYR(ITER,NA,NSP,NVC)
121. 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
122. PRNT(ITER)
123. 00
124. ***** CLOSURE PARAMETERS *****
125.      KBD  KED  KBP  KEP  KAR  KDT  KSP  ADJ
126.      5    5    1    2    1    1    1    0.0
127.      7    7    3    4    1    1    1    0.0
128.      8    8    1    2    1    1    2    0.0
129.     12   12    3    4    1    1    2    0.0
130. ***** LABELS ***** ;COL(NM)
131.      JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC
132. ***** STOCHASTIC PARAMETERS *****
133.      X1  X2  X3  X4  X5
134.     185 958 439 285 691
135.      SP1  SP2  SP3  SP4  SP5  SP6  SP7  SP8  SP9  SP10
136.      .05  .05  .05  .05  .05  .05  .05  .05  .05  .05
137. THE STUDENT T VALUE IS
138. 1.708
139. ***** NOMINAL DAYS FISHED ***** ;DFN(NSP,NA,ND,NVC,NM)
140.      0    0    7    33   414  402  228   29    4    2    0    0
141.      0    0    0    0    0    0    0    0    0    0    0    0
142.      0    0    0    0    0    0    0    0    0    0    0    0
143.      2    2    2    36  130  147  251  457  140  159  43  17
144.      0    0    0    0    0    0    0    0    0    0    0    0
145.     397  348  227  352  502  942  1758  1862  1401  1147  970  600
146.      2    2    22   33   29   15   18   790  577  367  179  50
147.      0    0    0    0    0    0    0    0    0    0    0    0
148.      0    0    0    0    0    0    0    0    0    0    0    0
149.      81   134  279  381  299  220  133  176  557  868  457  237
150.      0    0    0    0    0    0    0    0    0    0    0    0
151.      41   25   20   31   17   25   29   13   50   79  102  41
152. ***** DEMAND EQUATIONS *****
153. FLEX(10)
154. -0.10 -0.09 -0.05 -0.04 -0.05 -0.04 -0.04 -0.03
155. XBAR(NSC,NM)
156.      44633      35316      48826      31791      27703      74183      94900      127460
157.      80589      91880      111461      41835      102517      115299      88254      91566
158.     234362     210262     238155     221549     116971     312726     276628     204180
159.     168247     164996     127668     101826     293499     146542     161523     241576
160.     260568     1434186     805810     237788     156089     161423     64158     125210
161.     138768     88696     206584     391220     429208     716268     281801     171623
162.     246070     167915     156204     227516     212263     304462     959174     1069008
163.     1243644     697264     518625     292034     34429     39685     63037     91995
164.      96105     268865     804866     959606     354185     315084     162720     114735
165.      22798     22485     71633     101677     145474     709735     1250808     1254823
166.     262151     326454     252177     100353     9865     13346     54667     48910
167.     1081528     2131358     1021145     875505     392346     671639     1089847     376921

```

SIMULAT2

FOR THIS DATA STREAM NSP=1, ND=2, NC=24, NA=1, NVC=2, NSC=4, NM=12,  
NER=12, NPH=12, NBC=1, NSX=2.

```

1.      A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17 A18 A19 A20
2.      F  F  F  F  T  T  F  T  F  T  F  F  F  F  F  F  F  F  F  F
3.      T  T  T  F  F  F  F  T
4.      ***** TIME PARAMETERS *****
5.      IC ITER FTIME CCT CN CER CPH C8
6.      O 2 12 1.0 1.01 1.01 1.01 500
7.      ***** RECRUITMENT AND MOVEMENT COEFFICIENTS *****
8.      ICOF(NSP,NA)
9.      1.3000E 08
10.     E(NSP,NER)
11.     .083 .050 .119 .139 .136 .133 .079 .079 .046 .020 .033 .089
12.     ISZ(NSP,NSX)
13.     7.73 3.94
14.     ER(NSP,ND,NA,NA)
15.     .95
16.     SZCOF(NSP,NSX)
17.     23.78 23.19
18.     ***** GROWTH COEFFICIENTS *****
19.     GCOF(NSP,NSX)
20.     .37 .16
21.     MSZC(NSP,NSX)
22.     33.0 51.9
23.     GRT(NSP,NPH)
24.     1. 1. 1. 1. 1. 1. .5 .5 .5 1. 1. 1.
25.     C3(NSP)
26.     .0000014
27.     C4(NSP)
28.     2.83
29.     ***** NATURAL MORTALITY COEFFICIENTS ***** ;NMCOF(NSP,ND,NSC)
30.     .21 .21 .21 .21 .21 .21 .21 .21
31.     ***** HARVEST COEFFICIENTS *****
32.     BMESH RMESH BC RC ETM C5 C6 C7
33.     O O O O 1.0 40.78 20.03 1.0
34.     CMESH(NSP,ND)
35.     20.03 27.67
36.     CF(NVC,ND)
37.     1.0 1.0 1.0 1.0 1.0 1.0
38.     SL(NSP,NSC)
39.     40.78 31.93 27.67 20.03
40.     FMAX(NSP,ND,NA)
41.     .000730 .000827
42.     VKIL(NSP,ND,NM)
43.     20.03 20.03 20.03 20.03 20.03 20.03 20.03 20.03 20.03 20.03 20.03 20.03
44.     27.67 27.67 27.67 27.67 27.67 27.67 27.67 27.67 27.67 27.67 27.67 27.67
45.     PERF(NBC,NVC,NM)
46.     O.O O.O O.O O.O O.O O.O O.O O.O O.O O.O O.O O.O O.O O.O
47.     118. 118. 118. 118. 118. 118. 118. 118. 118. 118. 118. 118. 118. 118.
48.     O.O O.O O.O O.O O.O O.O O.O O.O O.O O.O O.O O.O O.O O.O
49.     118. 118. 118. 118. 118. 118. 118. 118. 118. 118. 118. 118. 118. 118.
50.     ***** RELATIVE FISHING POWER COEFFICIENTS *****
51.     FRL(NVC)

```

```

52.      4.5  40.
53. HP(NVC)
54.      .25 385.
55. CI(NSP)
56.      .1776
57. C2(NSP)
58.      .3906
59. ***** ECONOMIC COEFFICIENTS *****
60. CONV(NSP)
61.      1.0
62. NV(NVC)
63.      580 6
64. C(NVC)
65.      1 13
66. FC(NVC)
67.      54 84384
68. COST(NVC,ND)
69.      00 00 1516 1516
70. CVC(NVC)
71.      0 0
72. OCO(NVC)
73.      0 0
74. OCC(NVC)
75.      0 0
76. PC(NVC)
77.      0 0
78. SHARE(NVC)
79.      0 .118
80. PMCSP(NSP,NSC,NM)
81.      13.02 13.02 13.02 13.02 13.02 13.02 13.02 13.02 13.02 13.02 13.02 13.02
82.      9.80 9.80 9.80 9.80 9.80 9.80 9.80 9.80 9.80 9.80 9.80 9.80
83.      5.33 5.33 5.33 5.33 5.33 5.33 5.33 5.33 5.33 5.33 5.33 5.33
84.      1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
85. PCATCH(NBC)
86.      .30
87. ***** ADJUSTMENT FACTORS FOR EFFORT *****
88. YR(ITER)
89.      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
90. TYR(ITER,NA,NSP,NVC)
91.      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
92. PRNT(ITER)
93.
94. ***** CLOSURE PARAMETERS *****
95. KBD KED KBP KEP KAR KDT KSP ADJ
96. ***** LABELS ***** ;COL(NM)
97. JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC
98. ***** STOCHASTIC PARAMETERS *****
99.      IX1 IX2 IX3 IX4 IX5
100.      185 958 439
101. SP1 SP2 SP3 SP4 SP5 SP6 SP7 SP8 SP9 SP10
102.      .05 .05 .05 .05 .05 .05 .05 .05 .05 .05
103. ***** NOMINAL DAYS FISHID ***** ;DFN(NSP,NA,ND,NVC,NM)
104.      3900 3200 2700 2300 2500 3500 4200 4200 4200 5100 5000 4900
105.      0 0 0 0 0 0 0 0 0 0 0 0
106.      0 0 0 0 0 0 0 0 0 0 0 0
107.      112 116 123 119 134 118 132 133 132 139 139 134

```

Appendix F - OUTPUT OF GBFSM



NSP ND NC NA NVC NSC NM NER NPH NBC NSX  
 2 3 96 1 2 8 12 48 12 1 2  
 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17 A18 A19 A20  
 F F F T T T F F F T F F F F F F F F F F F F F  
 T T T T T F F F F F F F F F F F F F F F

\*\*\*\*\* TIME PARAMETERS \*\*\*\*\*  
 IC ITER FTIME CCT CN CER CPR CR  
 4 2 48 1.0 4.0 4.0 1.0 4.0 1000.

\*\*\*\*\* RECRUITMENT AND MOVEMENT COEFFICIENTS \*\*\*\*\*  
 ICOF(NSP,NA)  
 0.5500E+09 0.2500E+09  
 E(NSP,NER)  
 0.03 0.03 0.05 0.08 0.10 0.12 0.13 0.16 0.19 0.22 0.25 0.28  
 0.35 0.38 0.40 0.40 0.40 0.37 0.30 0.30 0.30 0.30 0.30 0.30  
 0.05 0.04 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03  
 0.03 0.03 0.03 0.03 0.03 0.02 0.02 0.01 0.01 0.01 0.01 0.01  
 0.23 0.23 0.23 0.23 0.23 0.05 0.05 0.05 0.23 0.23 0.23 0.23  
 0.35 0.35 0.35 0.35 0.35 0.35 0.25 0.25 0.25 0.25 0.25 0.25  
 0.22 0.22 0.22 0.22 0.15 0.15 0.35 0.35 0.35 0.35 0.35 0.35  
 ISZ(NSP,NSX)  
 25.00 25.00 25.00 25.00 25.00

ER(NSP,ND,NA,NA)  
 0.44 0.28 0.08 0.08 0.05 0.0  
 SZCOF(NSP,NSX)  
 50.00 50.00 50.00 50.00

\*\*\*\*\* GROWTH COEFFICIENTS \*\*\*\*\*  
 GCOF(NSP,NSX)  
 0.0720 0.0800 0.0739 0.0739  
 MSZC(NSP,NSX)  
 97.56 113.93 115.00 115.00  
 GRT(NSP,NPH)  
 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00  
 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00  
 C3(NSP)  
 0.00002340 0.00000920  
 C4(NSP)  
 3.00 3.20

\*\*\*\*\* NATURAL MORTALITY COEFFICIENTS \*\*\*\*\* :NMCOF(NSP,ND,NSC)  
 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04  
 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04  
 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08  
 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08

\*\*\*\*\* HARVEST COEFFICIENTS \*\*\*\*\*  
 BMESH RMESH BC EMT RC EMT C5 C6 C7  
 45.00 45.00 45.00 0.00 0.13 453.60 68.40 45.00 1.00  
 CMESH(NSP,ND)  
 45.00 45.00 45.00 60.00 60.00 60.00  
 CF(NVC,ND)  
 0.95 0.95 0.95 0.50 0.50 0.50  
 SL(NSP,NSC)  
 108.90 99.00 91.90 86.50 78.60 72.90 65.80 45.00 109.80 100.30 93.60 88.40  
 80.80 75.30 68.40 45.00

FMAX(NSP,ND,NA)	0.00019200	0.00008100	0.00003100	0.00012000	0.00004000	0.00007100			
VKIL(NSP,ND,NM)	67.60	67.60	67.60	67.60	67.60	67.60	67.60	67.60	67.60
	67.60	67.60	67.60	67.60	67.60	67.60	67.60	67.60	67.60
	67.60	67.60	67.60	67.60	67.60	67.60	67.60	67.60	67.60
	70.20	70.20	70.20	70.20	70.20	70.20	70.20	70.20	70.20
	70.20	70.20	70.20	70.20	70.20	70.20	70.20	70.20	70.20
PERF(NA,ND,NBC,NVC,NM)	0.	0.	0.	0.	0.	0.	0.	0.	0.
	0.	0.	0.	0.	0.	0.	0.	0.	0.
	0.	0.	0.	0.	0.	0.	0.	0.	0.
	0.	0.	0.	0.	0.	0.	0.	0.	0.
	0.	0.	0.	0.	0.	0.	0.	0.	0.
	0.	0.	0.	0.	0.	0.	0.	0.	0.

\*\*\*\*\* RELATIVE FISHING POWER COEFFICIENTS \*\*\*\*\*

FRL(NVC)	25.00	40.00							
HP(NVC)	150.00	300.00							
C1(NSP)	0.1776	0.1377							
C2(NSP)	0.3906	0.2261							

\*\*\*\*\* ECONOMICS COEFFICIENTS \*\*\*\*\*

CONV(SNSP)	1.00	1.00							
NV(NA,NVC)	500.00	1169.00							
C(NVC)	1.00	2.00							
FC(NVC)	7602.0030983	0.00							
COST(NVC,ND)	419.00	419.00	419.00	915.00	915.00	915.00	915.00	915.00	915.00
CVC(NVC)	0.0	0.0							
DCO(NVC)	10541.0013442	0.00							
OCC(NVC)	7621.0010701	0.00							
PC(NVC)	0.0	0.08							
SHARE(NVC)	0.20	0.20							
PMCS( NSP, NSC, NM)	6.36	6.50	6.50	6.46	6.56	6.60	6.60	6.60	6.60
	4.91	5.17	5.37	5.60	5.79	5.93	5.93	5.93	5.93
	4.33	4.58	4.78	5.04	5.21	5.46	5.46	5.46	5.46
	4.12	4.38	4.60	4.72	4.98	5.26	5.26	5.26	5.26
	3.63	3.88	4.03	4.22	4.58	4.93	4.93	4.93	4.93
	3.11	3.34	3.49	3.48	3.83	3.53	3.53	3.53	3.53
	2.50	2.72	2.75	2.77	3.05	2.75	2.75	2.75	2.75
	0.75	0.90	1.29	1.87	1.13	1.34	1.34	1.34	1.34
	6.36	6.50	6.50	6.46	6.56	6.60	6.60	6.60	6.60
	4.91	5.17	5.37	5.60	5.79	5.93	5.93	5.93	5.93
	4.33	4.58	4.78	5.04	5.21	5.46	5.46	5.46	5.46
	4.12	4.38	4.60	4.72	4.98	5.26	5.26	5.26	5.26
	3.63	3.88	4.03	4.22	4.58	4.93	4.93	4.93	4.93
	3.11	3.34	3.49	3.48	3.83	3.53	3.53	3.53	3.53
	2.50	2.72	2.75	2.77	3.05	2.75	2.75	2.75	2.75
	0.75	0.90	1.29	1.87	1.13	1.34	1.34	1.34	1.34
	5.89	5.66	5.89	5.66	5.43	5.89	5.89	5.89	5.89
	5.58	5.35	5.58	5.35	5.12	5.58	5.58	5.58	5.58
	4.74	4.51	4.74	4.51	4.28	4.74	4.74	4.74	4.74
	3.79	3.56	3.79	3.56	3.33	3.79	3.79	3.79	3.79
	3.06	2.83	3.06	2.83	2.60	3.06	3.06	3.06	3.06
	2.40	2.17	2.40	2.17	1.94	2.40	2.40	2.40	2.40
	1.40	1.17	1.40	1.17	0.94	1.40	1.40	1.40	1.40
	5.89	5.66	5.89	5.66	5.43	5.89	5.89	5.89	5.89
	5.58	5.35	5.58	5.35	5.12	5.58	5.58	5.58	5.58
	4.74	4.51	4.74	4.51	4.28	4.74	4.74	4.74	4.74
	3.79	3.56	3.79	3.56	3.33	3.79	3.79	3.79	3.79
	3.06	2.83	3.06	2.83	2.60	3.06	3.06	3.06	3.06
	2.40	2.17	2.40	2.17	1.94	2.40	2.40	2.40	2.40
	1.40	1.17	1.40	1.17	0.94	1.40	1.40	1.40	1.40
	5.89	5.66	5.89	5.66	5.43	5.89	5.89	5.89	5.89
	5.58	5.35	5.58	5.35	5.12	5.58	5.58	5.58	5.58
	4.74	4.51	4.74	4.51	4.28	4.74	4.74	4.74	4.74
	3.79	3.56	3.79	3.56	3.33	3.79	3.79	3.79	3.79
	3.06	2.83	3.06	2.83	2.60	3.06	3.06	3.06	3.06
	2.40	2.17	2.40	2.17	1.94	2.40	2.40	2.40	2.40
	1.40	1.17	1.40	1.17	0.94	1.40	1.40	1.40	1.40
	5.89	5.66	5.89	5.66	5.43	5.89	5.89	5.89	5.89
	5.58	5.35	5.58	5.35	5.12	5.58	5.58	5.58	5.58
	4.74	4.51	4.74	4.51	4.28	4.74	4.74	4.74	4.74
	3.79	3.56	3.79	3.56	3.33	3.79	3.79	3.79	3.79
	3.06	2.83	3.06	2.83	2.60	3.06	3.06	3.06	3.06
	2.40	2.17	2.40	2.17	1.94	2.40	2.40	2.40	2.40
	1.40	1.17	1.40	1.17	0.94	1.40	1.40	1.40	1.40
	5.89	5.66	5.89	5.66	5.43	5.89	5.89	5.89	5.89
	5.58	5.35	5.58	5.35	5.12	5.58	5.58	5.58	5.58
	4.74	4.51	4.74	4.51	4.28	4.74	4.74	4.74	4.74
	3.79	3.56	3.79	3.56	3.33	3.79	3.79	3.79	3.79
	3.06	2.83	3.06	2.83	2.60	3.06	3.06	3.06	3.06
	2.40	2.17	2.40	2.17	1.94	2.40	2.40	2.40	2.40
	1.40	1.17	1.40	1.17	0.94	1.40	1.40	1.40	1.40
	5.89	5.66	5.89	5.66	5.43	5.89	5.89	5.89	5.89
	5.58	5.35	5.58	5.35	5.12	5.58	5.58	5.58	5.58
	4.74	4.51	4.74	4.51	4.28	4.74	4.74	4.74	4.74
	3.79	3.56	3.79	3.56	3.33	3.79	3.79	3.79	3.79
	3.06	2.83	3.06	2.83	2.60	3.06	3.06	3.06	3.06
	2.40	2.17	2.40	2.17	1.94	2.40	2.40	2.40	2.40
	1.40	1.17	1.40	1.17	0.94	1.40	1.40	1.40	1.40
	5.89	5.66	5.89	5.66	5.43	5.89	5.89	5.89	5.89
	5.58	5.35	5.58	5.35	5.12	5.58	5.58	5.58	5.58
	4.74	4.51	4.74	4.51	4.28	4.74	4.74	4.74	4.74
	3.79	3.56	3.79	3.56	3.33	3.79	3.79	3.79	3.79
	3.06	2.83	3.06	2.83	2.60	3.06	3.06	3.06	3.06
	2.40	2.17	2.40	2.17	1.94	2.40	2.40	2.40	2.40
	1.40	1.17	1.40	1.17	0.94	1.40	1.40	1.40	1.40
	5.89	5.66	5.89	5.66	5.43	5.89	5.89	5.89	5.89
	5.58	5.35	5.58	5.35	5.12	5.58	5.58	5.58	5.58
	4.74	4.51	4.74	4.51	4.28	4.74	4.74	4.74	4.74
	3.79	3.56	3.79	3.56	3.33	3.79	3.79	3.79	3.79
	3.06	2.83	3.06	2.83	2.60	3.06	3.06	3.06	3.06
	2.40	2.17	2.40	2.17	1.94	2.40	2.40	2.40	2.40
	1.40	1.17	1.40	1.17	0.94	1.40	1.40	1.40	1.40
	5.89	5.66	5.89	5.66	5.43	5.89	5.89	5.89	5.89
	5.58	5.35	5.58	5.35	5.12	5.58	5.58	5.58	5.58
	4.74	4.51	4.74	4.51	4.28	4.74	4.74	4.74	4.74
	3.79	3.56	3.79	3.56	3.33	3.79	3.79	3.79	3.79
	3.06	2.83	3.06	2.83	2.60	3.06	3.06	3.06	3.06
	2.40	2.17	2.40	2.17	1.94	2.40	2.40	2.40	2.40
	1.40	1.17	1.40	1.17	0.94	1.40	1.40	1.40	1.40
	5.89	5.66	5.89	5.66	5.43	5.89	5.89	5.89	5.89
	5.58	5.35	5.58	5.35	5.12	5.58	5.58	5.58	5.58
	4.74	4.51	4.74	4.51	4.28	4.74	4.74	4.74	4.74
	3.79	3.56	3.79	3.56	3.33	3.79	3.79	3.79	3.79
	3.06	2.83	3.06	2.83	2.60	3.06	3.06	3.06	3.06
	2.40	2.17	2.40	2.17	1.94	2.40	2.40	2.40	2.40
	1.40	1.17	1.40	1.17	0.94	1.40	1.40	1.40	1.40
	5.89	5.66	5.89	5.66	5.43	5.89	5.89	5.89	5.89
	5.58	5.35	5.58	5.35	5.12	5.58	5.58	5.58	5.58
	4.74	4.51	4.74	4.51	4.28	4.74	4.74	4.74	4.74
	3.79	3.56	3.79	3.56	3.33	3.79	3.79	3.79	3.79
	3.06	2.83	3.06	2.83	2.60	3.06	3.06	3.06	3.06
	2.40	2.17	2.40	2.17	1.94	2.40	2.40	2.40	2.40
	1.40	1.17	1.40	1.17	0.94	1.40	1.40	1.40	1.40
	5.89	5.66	5.89	5.66	5.43	5.89	5.89	5.89	5.89
	5.58	5.35	5.58	5.35	5.12	5.58	5.58	5.58	5.58
	4.74	4.51	4.74	4.51	4.28	4.74	4.74	4.74	4.74
	3.79	3.56	3.79	3.56	3.33	3.79	3.79	3.79	3.79
	3								



96105.	268865.	804866.	959606.	315084.	162720.	114735.
22798.	22485.	71633.	101677.	709735.	1250808.	1254823.
262151.	326454.	252177.	100353.	13346.	54667.	48910.
1081528.	2131358.	1021145.	875505.	671639.	1089847.	376921.
MONTH= 1	SIZE= 1	PRICE=	6.28			
MONTH= 1	SIZE= 2	PRICE=	4.58			
MONTH= 1	SIZE= 3	PRICE=	4.41			
MONTH= 1	SIZE= 4	PRICE=	4.15			
MONTH= 1	SIZE= 5	PRICE=	3.76			
MONTH= 1	SIZE= 6	PRICE=	3.11			
MONTH= 1	SIZE= 7	PRICE=	2.50			
MONTH= 1	SIZE= 8	PRICE=	0.76			
MONTH= 2	SIZE= 1	PRICE=	6.13			
MONTH= 2	SIZE= 2	PRICE=	5.07			
MONTH= 2	SIZE= 3	PRICE=	4.64			
MONTH= 2	SIZE= 4	PRICE=	4.47			
MONTH= 2	SIZE= 5	PRICE=	4.00			
MONTH= 2	SIZE= 6	PRICE=	3.38			
MONTH= 2	SIZE= 7	PRICE=	2.75			
MONTH= 2	SIZE= 8	PRICE=	0.91			
MONTH= 3	SIZE= 1	PRICE=	6.47			
MONTH= 3	SIZE= 2	PRICE=	4.76			
MONTH= 3	SIZE= 3	PRICE=	4.76			
MONTH= 3	SIZE= 4	PRICE=	4.61			
MONTH= 3	SIZE= 5	PRICE=	4.14			
MONTH= 3	SIZE= 6	PRICE=	3.56			
MONTH= 3	SIZE= 7	PRICE=	2.83			
MONTH= 3	SIZE= 8	PRICE=	1.32			
MONTH= 4	SIZE= 1	PRICE=	5.84			
MONTH= 4	SIZE= 2	PRICE=	4.60			
MONTH= 4	SIZE= 3	PRICE=	4.96			
MONTH= 4	SIZE= 4	PRICE=	4.79			
MONTH= 4	SIZE= 5	PRICE=	4.36			
MONTH= 4	SIZE= 6	PRICE=	3.52			
MONTH= 4	SIZE= 7	PRICE=	2.80			
MONTH= 4	SIZE= 8	PRICE=	1.79			
MONTH= 5	SIZE= 1	PRICE=	5.97			
MONTH= 5	SIZE= 2	PRICE=	5.89			
MONTH= 5	SIZE= 3	PRICE=	5.37			
MONTH= 5	SIZE= 4	PRICE=	5.05			
MONTH= 5	SIZE= 5	PRICE=	4.60			
MONTH= 5	SIZE= 6	PRICE=	3.62			
MONTH= 5	SIZE= 7	PRICE=	2.91			
MONTH= 5	SIZE= 8	PRICE=	1.13			
MONTH= 6	SIZE= 1	PRICE=	6.75			
MONTH= 6	SIZE= 2	PRICE=	6.07			
MONTH= 6	SIZE= 3	PRICE=	5.23			
MONTH= 6	SIZE= 4	PRICE=	4.88			
MONTH= 6	SIZE= 5	PRICE=	4.27			
MONTH= 6	SIZE= 6	PRICE=	3.33			
MONTH= 6	SIZE= 7	PRICE=	2.74			
MONTH= 6	SIZE= 8	PRICE=	1.35			
MONTH= 7	SIZE= 1	PRICE=	6.11			
MONTH= 7	SIZE= 2	PRICE=	5.57			
MONTH= 7	SIZE= 3	PRICE=	4.20			
MONTH= 7	SIZE= 4	PRICE=	4.52			
MONTH= 7	SIZE= 5	PRICE=	3.99			
MONTH= 7	SIZE= 6	PRICE=	3.12			
MONTH= 7	SIZE= 7	PRICE=	2.48			
MONTH= 7	SIZE= 8	PRICE=	1.44			
MONTH= 8	SIZE= 1	PRICE=	5.96			

MONTH= 8	SIZE= 2	PRICE= 4.70
MONTH= 8	SIZE= 3	PRICE= 3.81
MONTH= 8	SIZE= 4	PRICE= 4.42
MONTH= 8	SIZE= 5	PRICE= 3.57
MONTH= 8	SIZE= 6	PRICE= 2.98
MONTH= 8	SIZE= 7	PRICE= 2.35
MONTH= 8	SIZE= 8	PRICE= 1.76
MONTH= 9	SIZE= 1	PRICE= 5.87
MONTH= 9	SIZE= 2	PRICE= 2.54
MONTH= 9	SIZE= 3	PRICE= 4.11
MONTH= 9	SIZE= 4	PRICE= 4.49
MONTH= 9	SIZE= 5	PRICE= 3.79
MONTH= 9	SIZE= 6	PRICE= 2.96
MONTH= 9	SIZE= 7	PRICE= 2.40
MONTH= 9	SIZE= 8	PRICE= 1.48
MONTH= 10	SIZE= 1	PRICE= 5.96
MONTH= 10	SIZE= 2	PRICE= 4.47
MONTH= 10	SIZE= 3	PRICE= 5.11
MONTH= 10	SIZE= 4	PRICE= 4.81
MONTH= 10	SIZE= 5	PRICE= 3.95
MONTH= 10	SIZE= 6	PRICE= 3.20
MONTH= 10	SIZE= 7	PRICE= 2.51
MONTH= 10	SIZE= 8	PRICE= 1.41
MONTH= 11	SIZE= 1	PRICE= 5.81
MONTH= 11	SIZE= 2	PRICE= 4.34
MONTH= 11	SIZE= 3	PRICE= 4.79
MONTH= 11	SIZE= 4	PRICE= 4.52
MONTH= 11	SIZE= 5	PRICE= 3.89
MONTH= 11	SIZE= 6	PRICE= 3.31
MONTH= 11	SIZE= 7	PRICE= 2.44
MONTH= 11	SIZE= 8	PRICE= 1.35
MONTH= 12	SIZE= 1	PRICE= 5.26
MONTH= 12	SIZE= 2	PRICE= 4.46
MONTH= 12	SIZE= 3	PRICE= 4.77
MONTH= 12	SIZE= 4	PRICE= 4.61
MONTH= 12	SIZE= 5	PRICE= 3.94
MONTH= 12	SIZE= 6	PRICE= 3.25
MONTH= 12	SIZE= 7	PRICE= 2.63
MONTH= 12	SIZE= 8	PRICE= 1.33

RUN 2 AREA 1 SPECIES 1  
 P R E D I C T E D V A L U E S

\*\*\*\*\*TOTAL CATCH BY SCP\*\*\*\*\*

	COMMERCIAL	RECREATION	BAIT
JAN	401187.	0.	2110.
FEB	268459.	0.	5373.
MAR	155453.	1031.	15885.
APR	395030.	9467.	30945.
MAY	1773328.	99758.	49688.
JUN	4223531.	188262.	50513.
JUL	6927102.	47660.	43860.
AUG	780840.	5289.	19674.
SEP	4710028.	252.	6790.
OCT	2985449.	92.	4964.
NOV	1773796.	0.	4694.
DEC	799564.	0.	3486.
TOT	32221168.	351812.	237981.

\*\*\*\*\*TOTAL COMMERCIAL CATCH BY DEPTH AND SCP\*\*\*\*\*

DEPTH-	1	2	3
JAN	0.	478.	400710.
FEB	0.	387.	268072.
MAR	7485.	820.	147148.
APR	69049.	35711.	290269.
MAY	807365.	248684.	717284.
JUN	1520922.	406797.	2295817.
JUL	375252.	738790.	5813061.
AUG	38879.	1134788.	6635173.
SEP	1851.	194807.	4513370.
OCT	671.	107718.	2877059.
NOV	0.	17475.	1756321.
DEC	0.	5455.	794109.
TOT	2821463.	2891888.	26508144.

\*\*\*\*\*TOTAL COMMERCIAL CATCH BY DEPTH, SIZE CLASS, AND SCP\*\*\*\*\*

DEPTH 1	1	2	3	4	5	6	7	8
JAN	0.	0.	0.	0.	0.	0.	0.	0.
FEB	0.	0.	0.	0.	0.	0.	0.	0.
MAR	378.	0.	1.	1.	10.	44.	251.	7179.
APR	3429.	0.	1.	9.	168.	735.	2977.	65159.
MAY	0.	0.	61.	309.	2966.	10286.	39293.	754450.
JUN	0.	0.	221.	789.	7613.	25050.	86406.	1400833.
JUL	1592.	0.	88.	290.	2721.	7376.	24507.	340269.
AUG	69.	0.	43.	115.	796.	1999.	5670.	30256.
SEP	30.	0.	7.	17.	103.	151.	262.	1310.
OCT	0.	0.	4.	6.	13.	28.	51.	570.
NOV	0.	0.	0.	0.	0.	0.	0.	0.
DEC	0.	0.	0.	0.	0.	0.	0.	0.
TOT	5499.	0.	426.	1537.	14390.	45679.	159419.	2600024.

DEPTH 2	1	2	3	4	5	6	7	8
JAN	88.	0.	22.	32.	91.	100.	133.	88.
FEB	115.	0.	23.	24.	64.	63.	92.	115.
MAR	407.	0.	18.	13.	48.	88.	241.	407.
APR	16030.	0.	163.	288.	2387.	5246.	11514.	16030.
MAY	102087.	0.	1456.	4085.	21029.	38800.	81050.	102087.
JUN	139375.	0.	5156.	9106.	43230.	74516.	135175.	139375.
JUL	213232.	0.	15955.	25642.	105081.	144013.	233530.	213232.
AUG	249481.	0.	42737.	52874.	183757.	236453.	364789.	249481.
SEP	21198.	0.	13322.	13932.	49717.	43823.	50525.	21198.
OCT	13357.	1.	14059.	11460.	22764.	24419.	18453.	13357.
NOV	3635.	2.	1738.	729.	4094.	2502.	3921.	3635.
DEC	1254.	2.	187.	298.	941.	993.	1537.	1254.
TOT	760258.	5.	94836.	118482.	433205.	571014.	900959.	760258.

DEPTH 3	1	2	3	4	5	6	7	8
JAN	3823.	114766.	56291.	93122.	51526.	26046.	18785.	3823.
FEB	2536.	56509.	57367.	45960.	39139.	17143.	11004.	2536.
MAR	5527.	21777.	35490.	13214.	21866.	11124.	12678.	5527.
APR	22419.	31811.	32940.	21709.	52783.	47110.	56461.	22419.

MAY	58599.	20679.	34815.	53961.	70941.	155633.	152739.	169917.	58599.
JUN	154502.	22379.	58394.	249262.	223372.	565113.	517404.	505393.	154502.
JUL	272776.	29654.	231130.	794895.	635770.	1632010.	1181909.	1034918.	272776.
AUG	217927.	21402.	464803.	1193635.	787715.	1834149.	1173336.	942209.	217927.
SEP	58419.	10929.	573601.	924146.	558347.	1349599.	618217.	420114.	58419.
OCT	16693.	16398.	619813.	667874.	417790.	728909.	312764.	96822.	16693.
NOV	14280.	30140.	558712.	323851.	234967.	472672.	68052.	53649.	14280.
DEC	8476.	35131.	298801.	69119.	158351.	148576.	41837.	33819.	8476.
TOT	835977.	311986.	3064929.	4458829.	3261251.	7051974.	4167678.	3355765.	835977.

COSTS & RETURNS

VESSEL CLASS 1									
	REV	VC	PACK	SHARES	RENT				
JAN	0.	0.	0.	0.	0.				
FEB	0.	0.	0.	0.	0.				
MAR	10369.	11732.	0.	2074.	-3437.				
APR	128348.	55308.	0.	25670.	47370.				
MAY	1022970.	520398.	0.	204594.	297978.				
JUN	2246317.	673752.	0.	449263.	1123302.				
JUL	585387.	286596.	0.	117077.	181713.				
AUG	76082.	48604.	0.	15216.	12262.				
SEP	3509.	6704.	0.	702.	-3897.				
OCT	1123.	3352.	0.	225.	-2454.				
NOV	0.	0.	0.	0.	0.				
DEC	0.	0.	0.	0.	0.				
TOT	4074100.	1606446.	0.	814820.	1652836.				

VESSEL CLASS 2									
	REV	VC	PACK	SHARES	RENT				
JAN	1714516.	1460340.	25676.	342903.	-114402.				
FEB	1242068.	1281000.	17181.	248414.	-304526.				
MAR	673732.	838140.	9470.	134746.	-308625.				
APR	1246294.	1420080.	20863.	249259.	-443906.				
MAY	3425153.	2313120.	61822.	685031.	365182.				
JUN	9691859.	3985740.	172967.	1938371.	3594781.				
JUL	22787744.	7352940.	419318.	4557548.	10457938.				
AUG	26059568.	8487540.	497277.	5211913.	11862838.				
SEP	16453751.	5640060.	301323.	3290750.	7221618.				
OCT	12811391.	4779960.	191026.	2562278.	5278128.				
NOV	7479850.	3707580.	113523.	1495969.	2162779.				
DEC	3413592.	2258220.	51172.	682718.	421482.				
TOT	106999472.	43524672.	1881616.	21399872.	40193264.				

RUN 2 AREA 1 SPECIES 2

PREDICTED VALUES

\*\*\*\*\*TOTAL CATCH BY SCP\*\*\*\*\*

COMMERCIAL				RECREATION				BAIT			
JAN	195961.		875.				67965.				
FEB	239255.		711.				55246.				
MAR	426636.		6035.				42625.				
APR	493043.		9209.				43364.				
MAY	347523.		9675.				51838.				
JUN	274297.		5982.				61971.				
JUL	229128.		8218.				70939.				
AUG	1474930.		199967.				79225.				

SEP 2508908. 285770. 76955.  
 OCT 2368336. 188460. 79791.  
 NOV 1364175. 91456. 79389.  
 DEC 486402. 12181. 74325.  
 TOT 10408524. 818539. 783631.

\*\*\*\*\*TOTAL COMMERCIAL CATCH BY DEPTH AND SCP\*\*\*\*\*

DEPTH-	1	2	3
JAN	5892.	115785.	74284.
FEB	5162.	186131.	47962.
MAR	42536.	345882.	38218.
APR	51804.	385603.	55637.
MAY	56647.	263127.	27749.
JUN	36078.	200532.	37688.
JUL	50828.	136607.	41693.
AUG	1250083.	206256.	18600.
SEP	1734352.	701071.	73493.
OCT	1139453.	1106916.	121973.
NOV	594739.	602135.	167309.
DEC	81921.	332486.	71995.
TOT	5049479.	4582498.	776600.

\*\*\*\*\*TOTAL COMMERCIAL CATCH BY DEPTH, SIZE CLASS, AND SCP\*\*\*\*\*

DEPTH 1	1	2	3	4	5	6	7	8
JAN	47.	32.	1042.	1117.	1174.	741.	344.	891.
FEB	38.	42.	1154.	813.	819.	582.	309.	715.
MAR	120.	505.	9924.	6396.	8122.	4455.	1359.	2282.
APR	487.	859.	10349.	8516.	4705.	2376.	1518.	9251.
MAY	0.	1011.	10209.	7289.	3171.	6543.	12438.	13813.
JUN	0.	680.	4267.	1401.	6808.	6611.	7141.	7384.
JUL	0.	882.	3081.	1711.	10082.	8229.	9239.	9332.
AUG	17931.	14434.	170850.	163995.	231489.	182584.	97335.	340698.
SEP	28794.	14057.	202342.	202906.	269268.	244620.	154923.	547084.
OCT	18752.	5094.	109597.	107762.	191470.	187955.	110223.	356293.
NOV	8535.	1228.	51239.	69452.	113208.	102109.	58044.	162156.
DEC	842.	163.	9016.	11463.	17693.	15357.	6890.	15990.
TOT	75545.	38986.	576584.	583551.	858011.	762161.	459762.	1465888.

DEPTH 2	1	2	3	4	5	6	7	8
JAN	2097.	3819.	30251.	22819.	17196.	7507.	2620.	2097.
FEB	2946.	8228.	54937.	27453.	19850.	9768.	3910.	2946.
MAR	1639.	17981.	85366.	40514.	36715.	14251.	3246.	1639.
APR	5565.	25299.	82720.	48532.	20001.	6753.	3284.	5565.
MAY	11882.	23863.	51852.	11090.	11379.	16444.	10598.	11882.
JUN	8912.	21753.	74427.	12494.	34642.	24126.	9753.	8912.
JUL	5776.	14265.	13618.	26083.	26607.	15220.	6326.	5776.
AUG	8182.	16244.	49297.	37693.	27376.	21229.	8520.	8182.
SEP	32072.	38806.	169558.	120051.	112214.	68906.	31785.	32072.
OCT	49896.	39479.	251037.	161657.	190081.	122059.	52075.	49896.
NOV	24935.	10883.	113548.	100255.	109665.	67154.	28574.	24935.
DEC	8413.	6272.	72658.	59893.	63735.	32457.	11228.	8413.
TOT	162316.	226891.	979825.	668531.	679840.	405872.	171918.	162316.



SIZE CLASS-	CULLS	1	2	3	4	5	6	7	8
JAN	117.	10214.	34621.	16083.	7857.	4006.	1116.	270.	117.
FEB	67.	8513.	23132.	10443.	3437.	1687.	533.	149.	67.
MAR	23.	7254.	21109.	6309.	1960.	1199.	310.	53.	23.
APR	86.	11325.	33417.	7014.	2718.	813.	192.	72.	86.
MAY	133.	7011.	16190.	3145.	462.	325.	322.	161.	133.
JUN	196.	12226.	19261.	1822.	997.	1941.	952.	294.	196.
JUL	236.	15242.	14824.	2960.	4037.	2896.	1137.	359.	236.
AUG	108.	5982.	4677.	3769.	1984.	1385.	534.	162.	108.
SEP	543.	17744.	22013.	16812.	8091.	5299.	2223.	768.	543.
OCT	908.	19835.	47406.	26753.	12057.	9612.	4097.	1304.	908.
NOV	1166.	15327.	79776.	32351.	18259.	13227.	5439.	1765.	1166.
DEC	308.	5996.	34618.	14961.	7849.	5709.	2017.	536.	308.
TOT	3891.	136669.	351045.	142423.	69709.	48100.	18874.	5893.	3891.

COSTS & RETURNS

VESSEL CLASS	1	2	3	4	5	6	7	8
REV	VC	PACK	SHARES	RENT	REV	VC	PACK	SHARES
JAN	20208.	3352.	0.	4042.	12815.			
FEB	19689.	3352.	0.	3938.	12399.			
MAR	181597.	36872.	0.	36319.	108406.			
APR	212277.	55308.	0.	42455.	114514.			
MAY	206453.	48604.	0.	41291.	116558.			
JUN	127129.	25140.	0.	25426.	76563.			
JUL	169029.	30168.	0.	33806.	105055.			
AUG	3880223.	993030.	0.	778045.	2119149.			
SEP	5001004.	967052.	0.	1000201.	3033752.			
OCT	3564723.	615092.	0.	712945.	2236687.			
NOV	1867271.	300004.	0.	373454.	1193812.			
DEC	279477.	62850.	0.	55895.	160732.			
TOT	15539079.	3140824.	0.	3107814.	9290440.			

VESSEL CLASS	2	3	4	5	6	7	8
REV	VC	PACK	SHARES	RENT	REV	VC	PACK
JAN	828777.	446520.	12164.	165755.	204337.		
FEB	1095744.	581940.	14982.	219149.	279673.		
MAR	1812882.	1094340.	24582.	362576.	331384.		
APR	2082821.	1507920.	28239.	416564.	130098.		
MAY	1535577.	1156560.	18616.	307115.	53286.		
JUN	1227959.	896700.	15246.	245592.	70422.		
JUL	822726.	592920.	11411.	164545.	53850.		
AUG	900180.	691740.	14391.	180036.	14014.		
SEP	2836037.	2221620.	49572.	567207.	-2362.		
OCT	5288497.	3466020.	78649.	1057699.	686130.		
NOV	3198911.	2045940.	49244.	639782.	463945.		
DEC	1724121.	1017480.	25887.	348824.	335931.		
TOT	23354192.	15719700.	342984.	4670843.	2620706.		

ECONOMIC DATA FOR AREA 1

TOTAL DAYS FISHED FOR VESSEL CLASS 1 IS 9848.  
 NUMBER OF VESSELS IN VESSEL CLASS 1 IS 500.

ECONOMIC DATA FOR AREA 1

TOTAL DAYS FISHED FOR VESSEL CLASS 2 IS 64748.  
 NUMBER OF VESSELS IN VESSEL CLASS 2 IS 1169.

TOTAL OWNERS COSTS \$1000!

VC SP	DAYF	\$/LB	CATCH	REV-F	REV-B	REV-T	TVC	FC	OC	TCOST	RENT
1	1	3192.	1.44	2821.	4074.	4074.	2421.	1232.	1708.	5362.	-1287.
2	1	47568.	3.64	29400.	106999.	106999.	66806.	26609.	11544.	104959.	2040.
1	2	6656.	3.08	5049.	15539.	15539.	6249.	2569.	3562.	12380.	3159.
2	2	17180.	4.36	5359.	23354.	23354.	20734.	9610.	4169.	34513.	-11159.
TOT VS. CL.											
1		9848.	2.49	7871.	19613.	19613.	8670.	3801.	5271.	17741.	1872.
2		64748.	3.75	34757.	130354.	130354.	87540.	36219.	15714.	139472.	-9119.
TOT SP											
1		50760.	3.45	32221.	111074.	111074.	69227.	27841.	13253.	110321.	753.
2		23836.	3.74	10409.	38893.	38893.	26982.	12179.	7732.	46893.	-8000.
TOT		74596.	3.52	42630.	149967.	149967.	96210.	40020.	20984.	157214.	-7247.

TOTAL CREWS COSTS \$1000!

VC SP	REV-F	REV-B	REV-T	TVC	OC	TCOST	RENT
1	1	815.	815.	0.	1235.	1235.	-420.
2	1	21400.	21400.	470.	18380.	18851.	2549.
1	2	3108.	3108.	0.	2575.	2575.	532.
2	2	4671.	4671.	86.	6638.	6724.	-2053.
TOT VS. CL.							
1		3923.	3923.	0.	3811.	3811.	112.
2		26071.	26071.	556.	25019.	25575.	496.
TOT SP							
1		22215.	22215.	470.	3810.	4281.	17934.
2		7779.	7779.	86.	25019.	25105.	-17326.
TOT		29993.	29993.	556.	28829.	29386.	608.

COMBINED COSTS

REV-F	REV-B	REV-T	TVC	FC	OC	TCOST	RENT
TOT	179960.	0.	179960.	96766.	40020.	49814.	146579.
							-6639.

