INFORMATIONAL LEAFLET NO. 178

SYSTEM LBOOK

A Fisheries Logbook Information Management System

By

SuzAnne M. Miller

and

Fred G. Gaffney

STATE OF ALASKA

Jay S. Hammond, Governor

DEPARTMENT OF FISH AND GAME

Ronald O. Skoog, Commissioner

Subport Building, Juneau 99801

February 1979

TABLE OF CONTENTS

SYSTEM LBOOK

A Fisheries Logbook Information Management System[1]

By:

SuzAnne M. Miller[2]

and

Fred G. Gaffney

Alaska Department of Fish and Game
Division of Commercial Fisheries
Alaska Pandalid Shrimp Research Project

I. INTRODUCTION

The purpose of this report is to document the design and use of the fisheries logbook information management system,--SYSTEM LBOOK. This computer system evolved out of the management need for timely analysis of standardized catch per unit effort (CPUE) date collected from the Gulf of Alaska commercial shrimp fishery (Gaffney 1977). SYSTEM LBOOK is a user oriented data base management system composed of subroutines which edits raw logbook information, updates the time series data base, retrieves user specified records, standardizes CPUE between fishing areas for a single fishing season or between fishing seasons, and reports this information in a variety of formats. The system was designed to be highly flexible but simple enough to be used by fisheries managers with little knowledge of computer programming.

---

[1]This project was partially funded by the Commercial Fisheries Research and Development Act of 1964 (P.L. 88-309 as amended) project number 5-42-R.

[2]Present address: U.S. Fish and Wildlife Service, 1011 East Tudor Road, Anchorage, Alaska 99503

Although SYSTEM LBOOK was written specifically for the Alaska domestic commercial shrimp fishery, it can be readily adapted for use in other fisheries. The majority of the Alaska shrimp harvest is taken from many individual bays and fjords along the coast. This affords the opportunity to manage these individual shrimp concentrations as discrete stock segments. These stock segments are identified as Geographical Catch Areas which are in turn composed of one or more Statistical Reporting Areas. The shrimp fishing year in Alaska begins the spring of 1 year and runs to the spring of the next year. Compilation of data on this basis is more meaningful biologically as stock trends are compared from the termination of one egg hatch period through the duration of the next.

Effort data, along with catch reports, are generally the first information collected by a fisheries management agency. Implicit in the interpretation of effort data is the assumption that the catchability coefficient must remain constant among time area strata and the measure of effort must be proportional to the instantaneous fishing mortality coefficient (Beaverton and Holt 1957). The assumption of a constant catchability is likely invalid for a complex and varied fishery such as the Gulf of Alaska domestic shrimp fishery (Gaffney 1977). Hence, the need to standardize fishing effort in order to make critical management decisions of the commercial exploitation of fish stocks.

Standardization of fisheries effort has been discussed by other investigators (Abramson and Tomlinson 1972; Fox and McCrary 1976; Gulland 1956; Robson 1966; Rothschild 1977). Previous work has hinged on the selection of a standard gear or vessel operating in a standard area. This assumes that stock distribution and abundance does not change in the time and area stratum. Also, the standard vessel's fishing performance does not change. Computerized standardization routines are available (Berude and Abramson 1972; Stark 1971), but they do not offer the necessary flexibility in analysis or reporting, nor do they provide for continuous data base management. To fill these needs, SYSTEM LBOOK was written.

SYSTEM LBOOK is a comprehensive data base management system which offers a unique approach to effort standardization. Relatively large data sets over numerous time-area stratas can easily be standardized. The uniqueness of this system is the flexibility it affords the user as there is no prescribed standardization method. This technique does not select a standard for comparison but rather assumes that the deviation in the mean reflects the deviation in the stock. Because of this greater flexibility more responsibility of the user is assumed.

Copies of the program on punched cards or magnetic tape can be obtained from the Alaska Pandalid Shrimp Research Project, Commercial Fisheries Division, Alaska Department of Fish and Game, Box 686, Kodiak, Alaska 99615.

II. SYSTEM DESIGN

Fisheries logbook information has many uses. Resource managers may be primarily interested in using the catch and effort statistics as an index to stock status. This requires timely reporting of up to date information. They also may be investigating the relationship between CPUE and vessel characteristics which require accessibility to logbook information over many years and areas. However, an individual skipper may be interested in comparing his vessel's performance in a number of different areas. His needs require that the information specific to his vessel be separated and analyzed independently.

In view of the varying demands for fisheries logbook information, the goal of a logbook information management system is flexibility and timeliness. This is reflected in the criteria used for the design of SYSTEM LBOOK:

1) Information Completeness
   All information on the original logbooks as filled out by the vessel skippers must be readily accessible.

2) Information Correctness
   Original logbook data must be edited for errors in coding.

3) Flexible Reporting of Information
   Each user must have the flexibility to tailor logbook information reports to his specific needs.

4) Timely Reporting of Information
   There must be no delay between the time new information is added to the system and the time new reports are generated.

5) Meaningful Reporting of Information
   The information must be reported in such a manner as to facilitate interpretation.

6) System Simplicity
   Minimal training should be required for system use.

II.A. INFORMATION FLOW

The task of processing fisheries logbook information was broken into five basic steps: 1) editing of raw logbook records, 2) condensing logbook information into meaningful categories, 3) retrieving both current and historical logbook information 4) report generating, and 5) information analysis. Since logbook information is received on an almost continual basis throughout the fishing season, simultaneous processing of the individual steps is necessary.

II.A.1. EDITING NEW LOGBOOK INFORMATION

The editing step consists of two functions:

1) Error checking

New logbook information is checked for validity of data. This requires the development of edit specifications. New data can be compared to the specifications, and errors noted. The edit specifications must be able to handle extra-normal data, but which is valid.

2) Addition of Information

Information which is static or calculated on the basis of other information need not be entered on the new logbooks. Vessel characteristics, for example, rarely change throughout the duration of a fishing season. CPUE is desired for each logbook entry, but this is a calculated value. This type of information is automatically added to the new logbook records.

11.A.2. CONDENSING LOGBOOK INFORMATION

While it is important that the original logbook information be retained, the most common use of the information is on a summary basis. To prevent continued recalculation of summary statistics and to condense large data sets into smaller, more manageable sets, meaningful categories of data must be formed. Rarely is information needed on an individual vessel basis. Information on different gear types, however, is often required. Similarly, stocks identified to specific geographic catch areas are more meaningful than the smaller statistical reporting areas. This led to the formation of summary categories based on gear type, geographic catch area, and date.

11.A.3. RETRIEVING LOGBOOK INFORMATION

As new information is added to existing data sets, the ability to extract portions of the data for reporting or analysis becomes more important. To insure complete flexibility, the criteria for data selection should include all possible combinations of data groups. Geographic catch area, statistical area, season, month, day, gear type,and vessel are designated as retrieval criteria.

11.A.4. REPORTING LOGBOOK INFORMATION

The most commonly used logbook information includes effort, catch, and CPUE. This information may be required on the basis of a number of different factors: by vessel, by area, by season, by date, or by a combination of breakdown factors. To accomodate these different needs, the reporting system will permit the user to specify up to four of the seven possible breakdown factors: 1) geographic catch area, 2) statistical area, 3) season, 4) month, 5) day, 6) gear type, or 7) vessel. The number of tows, the total catch, the mean CPUE, and the standard deviation of the CPUE is presented for each breakdown category (unique combination of breakdown factors). Breakdown factors are properly labeled to identify each category.

11.A.5. INFORMATION ANALYSIS

While logbook information may be used to investigate many different aspects of a fishery, one of the most important uses involves comparisons of the CPUE between different seasons, areas, vessels, gear types, or other factors. These types of comparisons can be misleading if the unit of effort is subject to change. Two analysis of variance models are included in the logbook information system to adjust for variation introduced in CPUE as a result of changes in the unit of effort. These two models are the results of an in-depth analysis of the sources of variation in CPUE (Miller 1977). Model I, the Within Season Model, is based on an individual vessel performance at a given time in a given geographical catch area. A substantial amount of data from a number of different vessels of each gear type is necessary for the model. This enables the model to correct for bias in the data. Model II, the Between Seasons Model, is based on the performance of the different gear types through time for a given geographical catch area. Model II can be employed with a small amount of data, and from Model II the standardized CPUE on an in-season basis can be predicted.

11.A.5.a. ANOVA MODEL I

The first model is a two factor nested or hierachic analysis of variance. It assumes that the second factor is nested or grouped under the different levels of the first factor. For example, the individual vessels of a particular gear type can be considered as being nested within that gear type. Development of Model I is based on individual vessels within a gear type within a time-area strata. The CPUE of a vessel operating in a specified time-area strata is assumed to consist of four quantities: 1) the mean CPUE for the strata; 2) the effect of the type of gear; 3) the effect of the individual vessel; and 4) a random error. Catch-effort frequency distributions for each time-area strata were calculated and found to be log normally distributed.

Model I assumes the following relationship:

$$Y_{ijk} = \mu + \alpha_i + \beta_{j(i)} + E_{ijk}$$

where

Y_{ijk} = $\log_e$ CPUE of the kth tow of the jth level of the second factor within the ith level of the first factor.

α_i = effect of the ith level of the first (major) factor.

$\beta_{j(i)}$ = effect of the jth level of the second (nested) factor within the ith level of the first factor.

E_{ijk} = error

The coefficients of α and β are estimated under the following restrictions:

$$\Sigma\alpha=0 \quad \text{and} \quad \Sigma\beta_{(i)}=0 \text{ for each } i.$$

Using these coefficients, the standardized CPUE is defined as

$$\text{Stand CPUE}_{ij} = e^{\hat{\theta}(1 + .5\sigma^2_{\hat{\theta}})}$$

where

$$\hat{\theta} = \hat{\mu} + \hat{\alpha}_i + \beta_{j(i)}$$

The estimates of the coefficients and the standardized CPUE are presented for each level of the two factors. An analysis of variance (ANOVA) table is an option which can be requested by the user.

II.A.5.b. ANOVA MODEL II

The second model is a two-way factorial analysis of variance without interaction. It assumes that the variation in CPUE is due to two factors of equal rank. For example, the performance of gear types across various seasons. Development of Model II was similar to that of Model I. Model II breaks down the CPUE for a given catch area into four quantities: 1) the average CPUE for the area over the seasons considered without regard to gear type; 2) the effect or deviation from the average of each gear type; 3) the effect of deviation from the average of each season; and 4) a random error term with an expected value of zero.

Model II assumes the following relationship:

$Y_{ijk} = \mu + \alpha_i + \beta_j + E_{ijk}$ where

Y_{ijk} = $\log_e$ CPUE of the kth tow of the ij combination of factors one and two.

α_i = effect of the ith level of the first factor.

β_j = effect of the jth level of the second factor.

E_{ijk} = error

The coefficients of α and β are estimated under the following restrictions:

$$\Sigma\alpha = 0 \quad \text{and} \quad \Sigma\beta = 0$$

Using these coefficients, the standardized CPUE is defined as

$$\text{Stand CPUE}_{ij} = e^{\hat{\theta}(1 + .5\sigma^2_{\hat{\theta}})}$$

where

$$\hat{\theta} = \hat{\mu} + \hat{\alpha}_i + \beta_j$$

The estimates of the coefficients and the standardized CPUE are presented for each level of the two factors. An optional ANOVA table can be requested by the user.

II.A.5.c. MODEL I vs MODEL II

Model I is intended to be used for specific detail during a given time period, i.e. a single fishing season. Model II is used to give a more generalized comparison between fishing seasons. Comparison of the results of Model I and Model II using the same data indicates a high degree of correlation (Miller 1977). The use of Model I for vessels within gear type across time did however, indicate interaction between area (i.e. stock abundance and distribution) and a gear type's relative ability to exploit the stock.

II.B. INFORMATION FILES

Associated with each step of the information process are data sets or information files. These files represent the input to each step and the output from each step.

FILE A--Unedited logbook data. This file is created directly from the original logbook.

FILE B--Edited logbook data. This file is the result of the editing step. It contains error-free, augmented logbook information.

FILE C--Summary logbook data. This file is the result of the summarizing of logbook information into categories.

FILE D--Retrieved data. This file is the result of an information retrieval from either FILE B or FILE C.

FILE E--Report statistics. This file contains the statistics generated by the report step. Further analysis can be made using this file of information.

III. COMPUTER IMPLEMENTATION

System LBOOK consists of two versions of a FORTRAN IV program written to be executed on the University of Alaska Computer Network's Honeywell 66/20. Although there are several unique features of the Honeywell FORTRAN that have been employed, ANSI Standard FORTRAN has been used whenever possible (see Appendix).

Two versions of the program were necessary to accomodate execution via the Time Sharing System (TSS) and the General Comprehensive Operating System (GCOS-BATCH). The versions differ only in the amount of data that each can handle. The TSS version provides for online communication with the program, but is limited to small data sets (5000 records or less) which are stored as

permanent disk files. The BATCH version requires submission of a batch job with the necessary Job Control Language (JCL), but can handle unlimited data sets stored on any media (disk, tape, cards). In order for both versions to utilize the same data files, all files must be in the ASCII character set. This is the character set normally used under TSS.

III.A. PROGRAM STRUCTURE

The computer program has a tree-like structure. The trunk is formed by the main program (PROGRAM LOGBOOK) which branches to the appropriate subroutine (SUBROUTINE EDIT, SUBROUTINE UPDATE, SUBROUTINE RETRV, SUBROUTINE REPORT, or SUBROUTINE STAND) for each step or task in the information process. Upon completion of a task, control is returned to the main program for continuation to the next task or termination of the program (see Figure III.I). The program documents the path of execution by printing the name of each program unit upon entry and edit. As many tasks as necessary can be performed in a single execution of the program. The order in which the tasks are performed is defined by the user. Communication between the program and the user is established via input directives.

III.B. INPUT DIRECTIVES

Input directives permit the user to define each task to be executed by the program. Associated with each program unit is a list of input variables. By setting the values of the input variables the user directs the flow of information through the system.

While the number and type of variables are unique to each program unit, the basic form of the form of the input directives is consistent throughout. Input directs consist of a series of assignment statements which set the appropriate variables to the desired values.

III.B.I. INPUT VARIABLES

Input variables are of three types. The value assigned to an input variable must be of the same type.

1) Type Integer--Integer variables may be assigned an integer constant. An integer constant is a continuous string of digits which does not contain a decimal point.
   EXAMPLES: 2 34567
   107 7879

2) Type Character--Character variables may be assigned any character value. A character value consists of a continuous string of alpha-numeric characters which is enclosed in single quotes (').
   EXAMPLES: 'KODIAK' 'WESTWARD REGION' 'FILE ONE'
   'INCOMPLETE DATA'

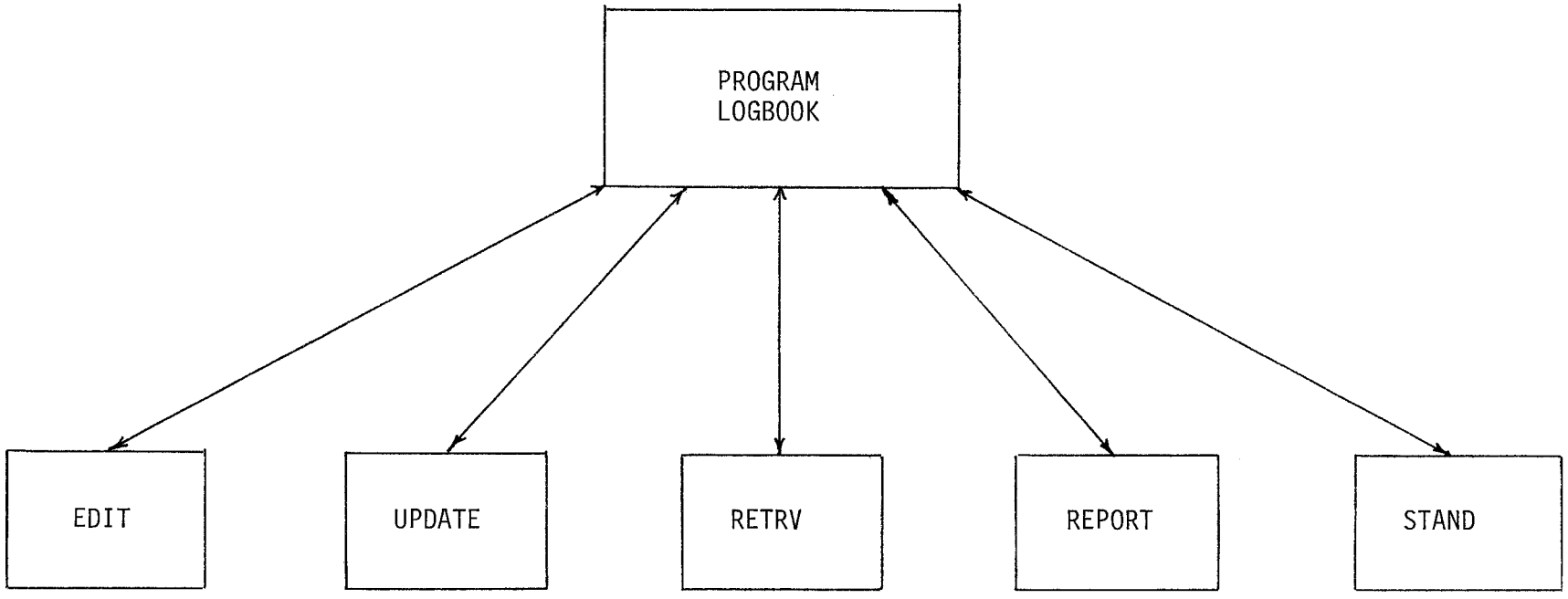

Figure III.1 STRUCTURE OF COMPUTER PROGRAM

3) Type Logical--Logical variables may be assigned a logical value. The logical value words TRUE or T and FALSE or F are enclosed in periods (.).
EXAMPLE: .TRUE. .T.
.FALSE. .F.

III.B.2 ASSIGNMENT STATEMENTS

All input information is in the form of assignment statements:

VARIABLE1=VALUE1

This defines the value of VARIABLE1 to be equal to VALUE1.

Some variables may take on more than one value. These variables are called arrays. Associated with those variables are the number of permissible values (called its DIMENSION). These variables are esentially lists. Associated with each assigned value is its position in the list. Values can be assigned to arrays in two manners:

1) VARIABLE=VALUE1, VALUE2....., VALUEN where VARIABLE has dimension N, and VALUE1 is the 1st value in the list, VALUE 2 the second, and VALUE N the Nth. For example, if FACTORS is a variable of type character and has a dimension of 3 we could assign the values of 'AREA', 'SEASON', and 'GEAR' to the 1st, 2nd, and 3rd places in the list respectively by the following:

FACTORS='AREA','SEASON','GEAR'

2) VARIABLE (subscript)=VALUE where subscript is an integer value which refers to the position of the value in the list. The value of the subscript must be equal to or greater than 1 and equal to or less than the dimension of the variable. Returning to the previous example of assigning values to the variable FACTORS, we could accomplish the same thing by the following: FACTORS(3)='GEAR', FACTORS(1)='AREA',FACTORS(2)='SEASON'

Notice that the order is unimportant, whereas under the first method of assignment, order is all important.

Default values are automatically assigned to all input variables. The user need only to include assignment statements for those variables which are to be changed from the default values.

III.B.3 SYNTAX

The basic form of an input directive is as follows: $IN Assignment Statement1, Assignment Statement2....., Assignment StatementN$. Input directives begin with an $IN and terminate with a $. Each program unit requests information only once. All input information must, therefore, be entered at one time in one input directive. If the information is too extensive to be contained on

one line or record, it may be continued on the next line. For example: $IN Assignment Statement1, Assignment Statement2, Assignment Statement3....., Assignment StatementN$.

The $IN must be followed by at least one blank character. The $IN need not begin in the first column, but it must be the first character combination encountered.

Each program unit allows for a varying number of assignment statements. All assignment statements must be separated by commas in an input entry. The order in which the assignment statement occurs is arbitrary.

III.B.4. SUMMARY OF RULES

The following rules apply to all input directives:

1) Input directives must begin with a $IN.

2) At least one blank must follow the $IN.

3) Assignment statements must be separated by commas.

III.C. COMPUTER FILES

In addition to the five logbook information files (FILEA, FILEB, FILEC, FILED, FILEE, two files containing the edit specifications for SUBROUTINE EDIT, files BOATS and DISTRICT are required. File BOAT contains information on all the vessels within the fleet and their corresponding characteristics. File DISTRICT contains a listing of all legal statistical areas and their corresponding geographic catch areas.

Each of the files used by system LBOOK is identified by a file number. File numbers are important to the user when executing a BATCH run of SYSTEM LBOOK. Otherwise, these numbers are transparent to the user. The format specifications, file numbers, and file descriptions are listed in Section III.J. Figure III.C.1 illustrates the use of files by SYSTEM LBOOK. Simultaneous processing of the different information tasks is possible. One user can be editing and correcting new log information while another is generating reports or analyzing historical information. Other computer software packages can be used to graphically display or analyze file contents.

File usage does require good organization on the part of the user. To avoid confusion about which file contains which information, a standard system of nomenclature is recommended. Use of the suffixes A, B, C, D, E, corresponding the FILEA, FILEB, FILEC, FILED, and FILEE, is helpful in identifying the contents of individual files.

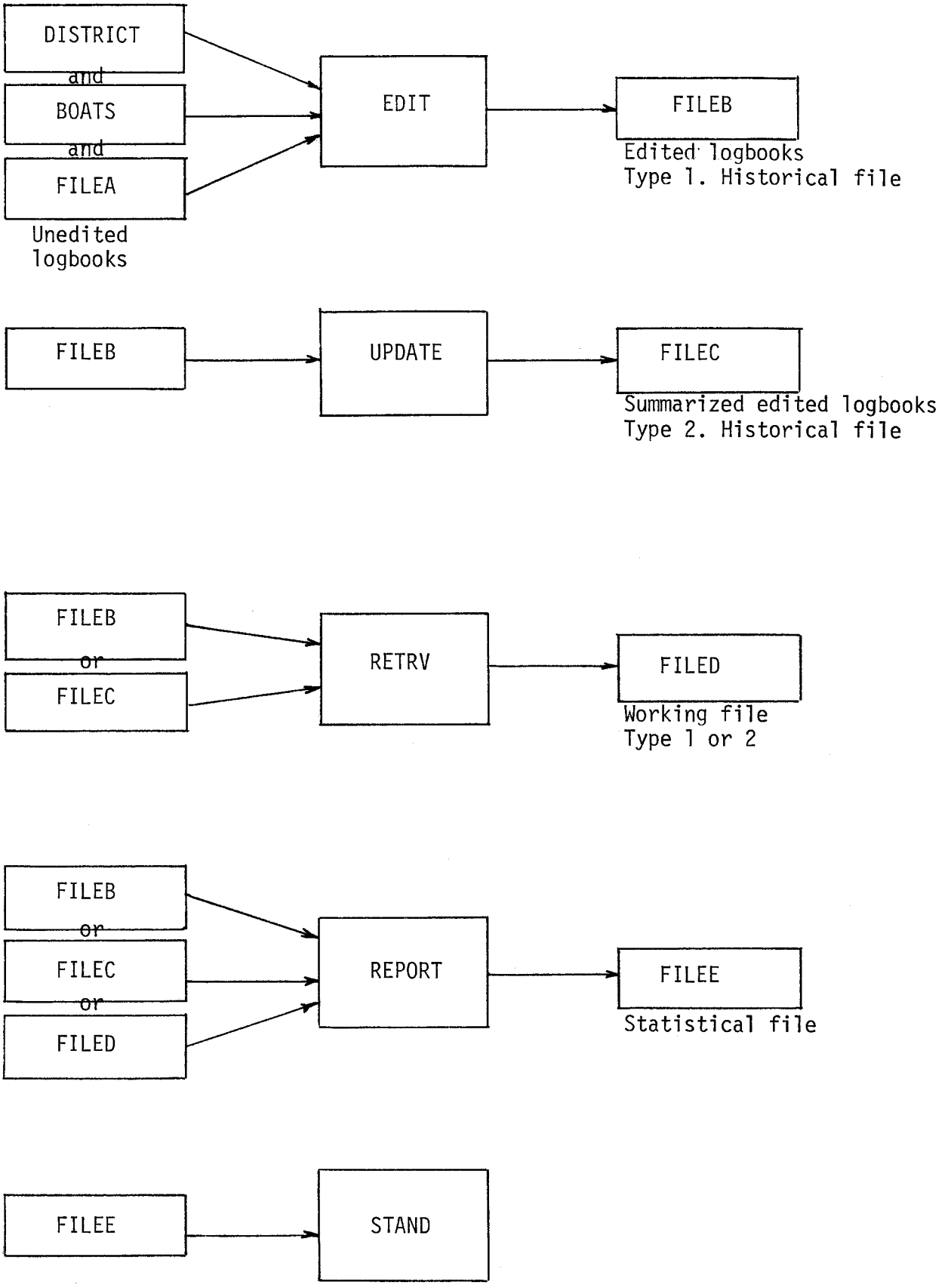

Figure III C.1. INFORMATION FLOW

III.D. PROGRAM LOGBOOK

INPUT DIRECTIVES

Variable name	Type	Default value	Purpose
EDIT	Logical	.F.	.TRUE. to call Subroutine EDIT
UPDATE	Logical	.F.	.TRUE. to call Subroutine UPDATE
RETRV	Logical	.F.	.TRUE. to call Subroutine RETRV
REPORT	Logical	.F.	.TRUE. to call Subroutine REPORT
STAND	Logical	.F.	.TRUE. to call Subroutine STAND
FINISH	Logical	.F.	.TRUE. to terminate program
TITLE	Character	Blank	Heading Title
WIDTH	Integer	136	Number of characters per output (maximum 136, minimum 72)

FILES

File number	Variable name	Purpose
05	-	Input directives
06	-	Printed output

ERROR MESSAGES

None

EXAMPLES

1. Call Subroutine UPDATE and REPORT
   $IN UPDATE=.T.,REPORT=.T.$

2. Call Subroutine EDIT then terminate program
   $IN EDIT=.T.,FINISH=.T.$

3. Terminate program
   $IN FINISH=.T.$

III.E. Subroutine EDIT

INPUT DIRECTIVES

Variable name	Type	Default value
TITLE	Character	Blank
FILEIN	Character	Blank
FILEOUT	Character	Blank
VESSELS	Character	'DGGAFFNEY/LBOOK/BOATS$BOATS,R'
GCAREAS	Character	'DGGAFFNEY/LBOOK/DISTRICT$DISTRICT,R'
LIST	Logical	.F.
IGNORE	Logical	.F.

	Purpose
TITLE --	Heading Title
FILEIN --	Name of file containing unedited log records. FILEIN must be specified.
FILEOUT --	Name of file to which edited records will be written. If FILEOUT already contains information, output records will be added to the end. If FILEOUT=BLANK (default value), no input file is created, and the records from FILEIN are simply checked for errors.
VESSELS --	Name of file containing information about vessels and corresponding characteristics.
GCAREAS --	Name of file containing information about geographic catch areas.
LIST --	.TRUE. to list records on FILEOUT
IGNORE --	.TRUE. to output records which contain errors to FILEOUT.

FILES

File number	Variable name	Purpose
05	-	Input directives
06	-	Printed output
11	FILEIN	Unedited logbooks
12	FILEOUT	Edited logbooks
25	VESSELS	Vessels information
26	GCAREAS	Catch area information

ERROR MESSAGES

Any record which is in error will be printed along with the appropriate message:

1.*****Invalid ADF&G Vessel number

2.*****Invalid Gear type

3.*****Invalid Statistical area

EXAMPLES

1. Read unedited log records from FILEA and check for errors only (i.e. no output file).
   $IN FILEIN='FILEA'$

2. Read unedited log records from FILEA, check for errors, and write correct edited records to FILEB. List records of FILEB.
   $IN FILEIN='FILEA',FILEOUT='FILEB',LIST=.T.$

3. Same as Example 2, except all records, regardless of errors, written to output file.
   $IN FILEIN='FILEA',FILEOUT='FILEB',LIST=.T.,IGNORE=.T.$

III.F. Subroutine UPDATE

INPUT DIRECTIVES

Variable Name	Type	Default value
TITLE	Character	Blank
FILEIN	Character	Blank
FILEOUT	Character	Blank
LIST	Logical	.F.

	Purpose
TITLE --	Heading Title
FILEIN --	Name of file containing edited log records. FILEIN must be specified.
FILEOUT --	Name of file to which summarized records will be written. Current contents of FILEOUT will be overwritten by UPDATE. FILEOUT must be specified.
LIST --	.TRUE. to list records on FILEOUT.

FILES

File number	Variable name	Purpose
05	-	Input directives
06	-	Printed output
12	FILEIN	Edited logbooks
13	FILEOUT	Summary records

ERROR MESSAGES

If the number of summary categories exceeds the limits the following warning message is printed. The program continued but no new categories are created.

*****WARNING - - - - NUMBER OF CELLS EXCEEDS Limit

LIMITATIONS

1. TSS version -- the number of categories (cells) is limited to 1000.

2. BATCH version -- the number of categories (cells) is limited to 2500.

EXAMPLES

1. Read edited log records from FILEB and write summary records to FILEC
   $IN FILEIN='FILEB',FILEOUT='FILEC'$

2. Same as Example 1, except a listing of FILEOUT is desired and a heading title.
   $IN FILEIN=FILEB,FILEOUT=FILEC,LIST=.T.,
   TITLE='1977 LOGS ADDED TO HISTORICAL FILE'$

III.G. Subroutine RETRV

INPUT DIRECTIVES

Variable name	Type	Default value
TITLE	Character	Blank
FILEIN	Character	Blank
FILEOUT	Character	Blank
AREA(1)	Integer	0
AREA(2)	Integer	99999
SEASON(1)	Integer	0
SEASON(2)	Integer	99999
GEAR(1)	Integer	0
GEAR(2)	Integer	99999
MONTH(1)	Integer	0
MONTH(2)	Integer	99999
DAY(1)	Integer	0
DAY(2)	Integer	99999
SAREA(1)	Integer	0
SAREA(2)	Integer	99999
VESSEL(1)	Integer	0
VESSEL(2)	Integer	99999
LIST	Logical	.F.
TYPE	Logical	2

		Purpose
TITLE	--	Heading Title
FILEIN	--	Name of file from which records are to be retrieved. FILEIN must be specified.

FILEOUT	--	Name of file to which retrieved records will be written. If FILEOUT=BLANK (default value) a temporary file is created. This temporary file can be used as input to subroutine REPORT under the same execution of the program. If FILEOUT already contains information, new records will be added to the end.
AREA(1),AREA(2)	--	Minimum and maximum values of geographic catch areas to be retrieved. The default values will cause all areas to be selected. Setting AREA(1) and AREA(2) equal to the same value, will cause only one area to be retrieved.
SEASON(1),SEASON(2)	--	Same as AREA(1),AREA(2) for retrieval based on seasons.
GEAR(1),GEAR(2)	--	Same as AREA(1),AREA(2) for retrieval based on gear types.
MONTH(1),MONTH(2)	--	Same as AREA(1),AREA(2) for retrieval based on months.
DAY(1),DAY(2)	--	Same as AREA(1),AREA(2) for retrieval based on days.
SAREA(1),SAREA(2)	--	Same as AREA(1),AREA(2) for retrieval based on statistical areas.
VESSEL(1),VESSEL(2)	--	Same as AREA(1),AREA(2) for retrieval based on vessels.
LIST	--	.TRUE. to list records on FILEOUT
TYPE	--	Type of input file: 1=edited log records, 2=summarized log records
		See CODES sections for legal values of AREA, SEASON, SAREA, VESSEL, GEAR, MONTH, DAY.

FILES

File number	Variable name	Purpose
05	-	Input directives
06	-	Printed output
13	FILEIN	Records to be retrieved
14	FILEOUT	Retrieved records.

ERROR MESSAGES

None

LIMITATIONS

None

EXAMPLES

1. Retrieve all information for areas 101 to 125 for single draggers for season 1973-1974 from FILEB.
   $IN AREA=101,125,SEASON=7374,7374,GEAR=07,08,FILEIN='FILEB'$

2. Retrieve all information for all areas, for the month of April of season 1975-1976 from FILEB and output records to FILED.
   $IN MONTH=4,4,SEASON=7576,7576,FILEIN='FILEB',FILEOUT='FILED'$

3. Retrieve all information on vessel 21669 for area 102 during the month of September from FILEB.
   $IN FILEIN='FILEB',TYPE=1,MONTH=9,9,VESSEL=21669,21669$

III.H. Subroutine REPORT

INPUT DIRECTIVES

Variable name	Type	Default value
TITLE	Character	Blank
FILEIN	Character	Blank
FILEOUT	Character	Blank
LIST	Logical	.F.
TYPE	Integer	2
FACTORS(1)	Character	'AREA'
FACTORS(2)	Character	'MONTH'
FACTORS(3)	Character	'GEAR'
FACTORS(4)	Character	Blank
TABLES	Logical	.T.

		Purpose
TITLE	--	Heading Title
FILEIN	--	Name of file containing information upon which report is to be based. If FILEIN=BLANK (default value) the program assumes that a temporary file was created under subroutine RETRV.
FILEOUT	--	Name of file to which summarization statistics will be written. One record for each breakdown of the innermost factor specified is written. If FILEOUT= BLANK (default value) a temporary file is created which can be used as input to subroutine STAND.

Purpose (continued)

FILEOUT (continued)		If FILEOUT already contains information, that information is overwritten.
LIST	--	.TRUE. to list records on FILEOUT
TYPE	--	Type of input file: 1=Edited log records 2=summarized log records
FACTORS(1)	--	1st breakdown variable.
FACTORS(2)	--	2nd breakdown variable.
FACTORS(3)	--	3rd breakdown variable.
FACTORS(4)	--	4th breakdown variable.
		Possible values for FACTORS are:
		'AREA' --breakdown by geographic catch
		'SEASON' --breakdown by season
		'GEAR' --breakdown by gear type
		'MONTH' --breakdown by months
		'DAY' --breakdown by day
		'SAREA' --breakdown by statistical area
		'VESSEL' --breakdown by vessels
TABLES	--	.TRUE. to produce printed report, .FALSE. to create an output file only.

FILES

File number	Variable name	Purpose
05	-	Input directives
06	-	Output directives
14	FILEIN	Information to be reported
15	FILEOUT	Summary statistics

ERROR MESSAGES

If the number of breakdown categories exceeds the limit, the following warning message will be printed. The program will continue, but no new categories (CELLS) will be created:

*****WARNING---NUMBER OF CELLS EXCEEDS Limit
If an illegal FACTORS is specified, the following is printed
*****ILLEGAL FACTOR=

LIMITATIONS

1. TSS version---the number of categories (CELLS) is limited to 1000.

2. BATCH version---the number of categories (CELLES) is limited to 3000.

EXAMPLES

1. Report the information contained on a temporary file created by RETRV broken down by AREA,MONTH,GEAR:$IN $

2. Report the information contained in file FILEC by AREA,SEASON,MONTH:
   $IN FACTORS(1)='AREA',FACTORS(2)='SEASON',FACTORS(3)='MONTH',
   FILEIN='FILEC'$

3. Report the information contained in file FILEB by AREA,GEAR,VESSEL, and write statistical records to FILEE:
   $IN FACTORS='AREA','GEAR','VESSEL',FILEIN='FILEB',FILEOUT='FILEE'$

III.I. Subroutine STAND

INPUT DIRECTIVES

Variable name	Type	Default value Model I	Default value Model II
TITLE	Character	Blank	Blank
FILEIN	Character	Blank	Blank
FACTORS(1)	Character	'AREA'	'AREA'
FACTORS(2)	Character	'GEAR'	'SEASON'
FACTORS(3)	Character	'VESSEL'	'GEAR'
MODEL	Integer	1	2
ANOVA	Logical	.T.	.T.

		Purpose
TITLE	--	Heading Title
FILEIN	--	Name of file containing statistical information to be standardized. If FILEIN=BLANK (default value) the program assumes that a temporary file has been created by subroutine REPORT.

FACTORS(1)	--	1st breakdown factor used when file was created by REPORT
FACTORS(2)	--	2nd breakdown factor
FACTORS(3)	--	3rd breakdown factor
		See Subroutine REPORT for discussion on FACTORS
MODEL	--	Standardization model desired 1=MODEL I 2=MODEL II
ANOVA	--	.TRUE. to print Analysis of Variance Table.

FILES

File number	Variable name	Purpose
05	-	Input directive
06	-	Printed output
15	FILEIN	Statistical information

ERROR MESSAGES

If MODEL does not equal 1 or 2, the following is printed:

*****ILLEGAL MODEL NUMBER=

If an illegal value of FACTORS is specified, the following message appears:

*****ILLEGAL FACTOR=

If the level limit is exceeded, the program terminates and the following is printed:

*****NUMBER OF LEVELS EXCEEDS Limits

LIMITATIONS

1. TSS version--the number of levels of any factors cannot exceed 25.

2. BATCH version--the number of levels of any factor cannot exceed 50.

EXAMPLES

1. Standardize the statistics generated by REPORT which were broken down by 'AREA','GEAR','VESSEL' according to MODEL I.
   $IN FACTORS='AREA','GEAR,'VESSEL', MODEL=1$

2. Standardize the statistics generated by REPORT which were broken down by 'AREA','SEASON','GEAR', according to MODEL II.
   $IN $

3. Standardize the statistics generated by a previous execution of REPORT and stored on file FILEE. The report was broken down by 'AREA','GEAR','SEASON', and MODEL II is desired.
   $IN FACTORS='AREA','GEAR','SEASON',FILEIN='FILEE'$

4. Standardize the statistics generated by a previous execution of REPORT and stored on file FILEE. The report was broken down by 'AREA','GEAR', 'SEASON', MODEL II is desired and an ANOVA table is requested.
   $IN FACTORS='AREA','GEAR','SEASON',FILEIN='FILEE',ANOVA=.T.$

III.J. FILE FORMATS

III.J.1. FILEA---UNEDITED LOGBOOK RECORDS

Character Set: ASCII
Characters/Record: 52
Logical Unit Device Number: 11
Format:

Columns	Read format	Information
1-4	I4	Season
5-9	I5	ADF&G Vessel number
10-11	I2	Gear code
12-14	I3	Footrope length
15-16	I2	Month
17-18	I2	Day
19-22	I4	Time start
23-26	I4	Time end
27-27	I1	Hours
28-29	I2	Minutes
30-30	A1	Light or Dark
31-33	I3	% Cloud cover
34-36	F3.1	Temp (°C)
37-39	I3	Average depth
40-44	F5.0	Pounds caught
45-49	I5	Statistical area
50-52	I3	% fish

III.J.2. FILEB --EDITED LOGBOOK RECORDS

Character Set: ASCII
Characters/Record: 72
Logical Unit Device Number: 12
Format:

Columns	Read format	Information
1-3	I3	Geographic catch area
4-7	I4	Season
8-9	I2	Gear code
10-11	I2	Month
12-13	I2	Day
14-18	I5	Statistical area
19-23	I5	ADF&G number
24-26	I3	Horsepower
27-29	I3	Vessel length
30-32	I3	Footrope length
33-36	I4	Time start
37-40	I4	Time end
41-43	F3.1	Effort (hours)
44-44	A1	Light or Dark
45-47	I3	% Cloud cover
48-50	F3.1	Temp (°C)
51-53	I3	Average depth
54-58	F5.0	Pounds caught
59-64	F6.1	CPUE
65-69	F5.3	ln (CPUE)
70-72	F3.2	% fish

III.J.3. FILEC---SUMMARIZED LOGBOOK RECORDS

Character Set: ASCII
Characters/Record: 76
Logical Unit Device Number: 13
Format:

Columns	Read format	Information
1-3	I3	Geographic catch area
4-7	I4	Season
8-9	I2	Gear code
10-11	I2	Month
12-13	I2	Day
14-16	I3	Number of tows
17-26	E10.4	Total pounds
27-36	E10.4	Pounds **2
37-46	E10.4	Total (CPUE)
47-56	E10.4	CPUE **2
57-66	E10.4	Total ln(CPUE)
47-76	E10.4	ln(CPUE)**2

III.J.4. FILED---RETRIEVED INFORMATION

Character Set: ASCII
Characters/Record: TYPE 1-72; TYPE 2=76
Logical Unit Device Number: 14
Format: The format of FILE D follows that of the file which was used as as input for the retrieval.

III.J.5. FILEE---CPUE STATISTICS

Character Set: ASCII
Characters/Record: 75
Logical Unit Device Number: 15
Format:

Columns	Read format	Information
1-5	I5	Value of FACTORS(1)
6-10	I5	Value of FACTORS(2)
11-15	I5	Value of FACTORS(3)
16-20	I5	Value of FACTORS(4)
21-25	F5.0	Number of tows
26-35	E10.4	Total pounds
36-45	E10.4	Mean CPUE
46-55	E10.4	Standard deviation CPUE
56-65	E10.4	Total ln(CPUE)
66-75	E10.4	Sum ln CPUE**2

III.J.6. File BOATS--VESSEL CHARACTERISTICS

Character Set: ASCII
Characters/Record: 46
Logical Unit Device Number: 25
Format:

Columns	Read format	Information
1-5	I5	ADF&G vessel number
6-8	I3	Gear code
9-12	I4	Horsepower
13-15	I3	Vessel length
16-46	A30	Vessel name

III.J.7. File DISTRICTS--GEOGRAPHIC CATCH AREAS

Character Set: ASCII
Characters/Record: 9
Logical Unit Device Number: 26
Format:

Columns	Read format	Information
1-5	I5	Statistical area
6-9	I4	Geographic catch area

III.K. CODES

AREA -- The codes for geographic catch areas are formed by adding the ADF&G catch area number to the appropriate region number:

Region	Number
Kodiak	100
Chignik	200
South Peninsula	300

EXAMPLE: Kiliuda Bay is catch area number 07 in the Kodiak region; therefore, the geographic catch area number is 107.

SEASON-- The seasons codes are formed by combining the last two digits of the two calendar years in which the season occurs.

EXAMPLE: A fishing season which begins in April 1972 and runs to February 1973 would be coded 7273.

GEAR -- The following gear codes apply:

07 -- Single dragger
17 -- Beam trawl
27 -- Double dragger

MONTH-- Numeric value of the month (e.g. January=1)

DAY -- Numeric value of the day

VESSEL--5 digit ADF&G number

SAREA --5 digit ADF&G statistical area number

III.L. HONEYWELL 66/20 SYSTEM CONSIDERATIONS

Several features of the Honeywell System must be kept in mind when using SYSTEM LBOOK.

1. Creation of Files:

Before a file can be used as an output file from Program LOGBOOK that file must already exist. New files can be created via the Timesharing System ACCESS. The file must be large enough to contain all the information which

will be output. In addition the file must have an EOF (end of file) at the beginning. This can be accomplished by saving an empty current file in it via TSS:

*CARD NEW

*RESAVE FILENAME

where filename is the name of the file previously created under ACCESS.

2. Character Sets:

The General Comprehensive Operating System (the BATCH environment) normally operates in a BCD environment. The only time that this will be of concern to the user of LBOOK is in the creation of the unedited logbook file. If new logs are entered into a file via the Timesharing System, there are no further steps necessary. If new logs are entered via the TSS command BCDASC, see the Honeywell TSS/BATCH Interface Manual for instructions.

III.M. EXECUTION OF PROGRAM LOGBOOK

III.M.1. TSS version

A response of DGGAFFNEY/LBOOK/TSS/,E to the Timesharing System inquiry of * will load and execute Program LOGBOOK:

*DGGAFFNEY/LBOOK/TSS,E

The program will tell you which section is executing (PROGRAM LOGBOOK, SUBROUTINE EDIT, SUBROUTINE UPDATE, SUBROUTINE RETRV, SUBROUTINE REPORT, or SUBROUTINE STAND) and will prompt you for the appropriate input directives with:

READY
=

at which time you should enter the input entry for the section executing.

Subroutines REPORT and STAND will ask you to position the paper for the printing of a report. You should position the paper manually without using the automatic line feed and depress the RETURN key when ready.

III.M.2. BATCH version

To submit a BATCH job, a file of the correct Job Control Language (JCL) must be created under TSS and submitted via the JRUN command. In addition, a file containing the input directives necessary, in the proper order, must be created. This input directives file cannot contain line numbers (use system EDITOR to create the file).

Job Control Language (JCL)

The following JCL is necessary to load and execute the BATCH version of Program LOGBOOK:

```
Column 1          8                16
       $          IDENT            Userid,Banner
       $          OPTION           FORTRAN
       $          USE              .GTLIT
       $          SELECT           DGGAFFNEY/LBOOK/BATCH,R
       $          EXECUTE
       $          LIMITS           50,75K

            Appropriate JCL for files
                                    letter-number
                   .
                   .
                   .
       $          SYSOUT           06,ORG
       $          ENDJOB
```

Each file that is to be used by Program LOGBOOK must be identified by its Logical Unit Device Number and the media upon which it resides (Magnetic Tape, Permanent Disk File, Temporary Mass Storage). The Logical Unit Device number (LUD) can be found under the descriptions of each section of the program and under the descriptions of the individual files. Each type of media requires different JCL.

1. Magnetic tapes

   A magnetic tape file which is to be used as input only to program LOGBOOK requires the following:

```
Column 1          8          16
       $          TAPE9      LUD,T#D,Tape number, Tape label
```

   where

   LUD=Logical Unit Device Number

   # =consecutive number of the tape used during this job
   (#=1 if 1st tape, #=2 if 2nd, etc.)

   Tape number=5 digit tape number assigned by Tape Librarian.

   Tape label=outside label of tape.

   EXAMPLE:

   If the unedited log records are stored on a magnetic tape numbered X0155 with label FILEA, and they are to be used as input to subroutine EDIT, the following JCL is necessary.

```
Column 1          8          16
       $          TAPE9      11,TID,X0155,FILEA
```

In addition to the above JCL, magnetic tape files which are to used as output or as input/output must have the following:

```
Column 1          8        16
       $          MSG2     2,RING IN Tape number,Userid,Label
       $          FFILE    LUD,NOSLEW
```

where

Tape number=5 digit tape number assigned by Tape Librarian
Userid=userid under which tape is stored
Label=outside label of tape.

EXAMPLE:

Continuing with Example 1, if in addition to reading unedited log records from tape X0155, we wished to write the edited log records on tape X0156 with label FILEB which was under DGGAFFNEY, we would need the following:

```
Column 1          8        16
       $          TAPE9    11,T1D,X0155,FILEA
       $          TAPE9    12,T2D,,X0156,,FILEB
       $          MSG2     2,RINGIN,X0156,DGGAFFNEY,FILEB
       $          FFILE    12,NOSLEW
```

2. Permanent Disk Files

   A permanent disk file which is to be used as input only to program LOGBOOK requires the following:

```
Column 1          8        16
       $          PRMFL    LUD,R,S,Userid/Filename
```

   where

   LUD=Logical Unit Device Number

   Userid=Userid under which the file is stored

   Filename=name of the file

   EXAMPLE:

   If summarized log records are stored on file FILEC under DGGAFFNEY, and they are to be used as input to subroutine RETRV, the following JCL is necessary:

```
Column 1        8          16
      $         PRMFL      13,R,S,DGGAFFNEY/FILEC
```

The JCL for permanent disk files which are to be used as output only has a similar form:

```
Column 1        8          16
      $         PRMFL      LUD,W,S,Userid/Filename
      $         FFILE      LUD,NOSLEW
```

The R (for READ) is changed to a W (for WRITE). Permanent files which are used as both input and output require a R/W or a W/R.

```
Column 1        8          16
      $         PRMFL      LUD,R/W,S,Userid/Filename
      $         FFILE      NOSLEW
```

EXAMPLE:

Continuing with Example 1, if the retrieved records from file FILEC are to be written to FILED which is then to be used as input to subroutine REPORT, the following JCL would be necessary:

```
Column 1        8          16
      $         PRMFL      13,R,S,DGGAFFNEY/FILEC
      $         PRMFL      14,W/R,S,DGGAFFNEY/FILED
      $         FFILE      14,NOSLEW
```

3. Temporary Mass Storage

   Temporary files used as output, input, or both are identified by the following:

```
Column 1        8          16
      $         FILE       LUD,D#
      $         FFILE      LUD,NOSLEW
```

   where

   LUD=Logical Unit Device Number

   # =Consecutive number of temporary file used during this job.

EXAMPLE:

Retrieve records stored on permanent file FILEC under DGGAFFNEY, write retrieval records to a temporary file and use the temporary file as input to a subroutine REPORT:

```
Column 1        8       16
       $        PRMFL   13,R,S,DGGAFFNEY/FILEC
       $        FILE    14,DI
       $        FFILE   14,NOSLEW
```

As mentioned earlier, the BATCH version of program LOGBOOK requires the input directives to be entered via a file. This file has a Logical Unit Drive number of 05. In order to create this file, the order in which the directives are read must be known.

Program LOGBOOK always begins execution and expects the first input directive. If an input direct to LOGBOOK calls more than one subroutine, the subroutines will be executed in the following order: 1) EDIT, 2) UPDATE, 3) RETRV, 4) REPORT, 5) STAND. Control will not return to LOGBOOK, but will continue to the next called routine. Upon completion of all called subroutines, control will return to LOGBOOK. The easiest method to keep track of program execution is, therefore, to call only one subroutine with each input directive to LOGBOOK. This assures the following order of input directives.

```
INPUT DIRECTIVE  --  LOGBOOK
INPUT DIRECTIVE  --  Called Subroutine 1
INPUT DIRECTIVE  --  LOGBOOK
INPUT DIRECTIVE  --  Called Subroutine 2
```

To create the input directive file, use the Timesharing System of EDITOR:

```
*EDITOR NEW
ENTER
* Input Directive
* Input Directive
* Input Directive
* (Carriage return)
- SAVE filename
```

EXAMPLE:

Create an input directive file called DIRECT (under DGGAFFNEY) to do the following:

1. Read unedited log records from permanent file FILEA (under DGGAFFNEY)

2. Write edited log records to permanent file FILEB (under DGGAFFNEY)

3. Update file FILEC (under DGGAFFNEY) by the new edited logs on FILEB.

```
*EDITOR NEW

ENTER

* $IN EDIT=.T.$

* $IN FILEIN='DGGAFFNEY/FILEA'

* $FILEOUT='DGGAFFNEY/FILEB',LIST=.T.$

* $IN FILEIN='DGGAFFNEY/FILEB',

* FILEOUT='DGGAFFNEY/FILEC',LIST=.T.$

* $IN FINISH=.T.$

* (Carriage return)
- SAVE DIRECT
```

The JCL for the job would be:

```
Column 1        8               16
       $        IDENT           DGGAFFNEY,BATCHJOB
       $        OPTION          FORTRAN
       $        USE             .GTLIT
       $        SELECT          DGGAFFNEY/LBOOK/BATCH,R
       $        EXECUTE
       $        LIMITS          50,75K
       $        PRMFL           05,R,S,DGGAFFNEY/DIRECT
       $        PRMFL           11,R,S,DGGAFFNEY/FILEA
       $        PRMFL           12,W/R,S,DGGAFFNEY/FILEB
       $        FFILE           12,NOSLEW
       $        PRMFL           13,R/W,S,DGGAFFNEY/FILEC
       $        FFILE           13,NOSLEW
       $        PRMFL           25,R,S,DGGAFFNEY/LBOOK/BOATS$BOATS
       $        PRMFL           25,R,S,DGAFFNEY/LBOOK/DISTRICT$DISTRICT
       $        SYSOUT          06,ORG
       $        ENDJOB
```

A BATCH job submitted via JRUN will be returned to the JOUT system. The output can then be directed to a printer and mailed, or printed on the terminal by the use of EPRINT on file code 06.

IV. LITERATURE CITED

Abramson, N.J. and P.K. Tomlinson, 1972. An application of Yield Models to a California Ocean Shrimp Population. Fish. Bull., U.S. 70:1021-1041.

Berude, C.L. and N.J. Abramson, 1972. Relative Fishing Power, CDC 6600 FORTRAN IV. Trans. Am. Fish. Soc. 101:133.

Fox, W.W. and J.A. McCrary, 1976. The Relative Fishing Power of Otter Trawls in the Kodiak Shrimp Fishery. Unpublished ms., NMFS So. West Fish. Centr., La Jolla, Calif. 13 pp.

Gaffney, F.G., 1977. Kodiak Pandalid Shrimp Research Comp. rept. 5-36-K, Comm. Fish. Res. and Devel. Act. July 1, 1973 to June 30, 1976. 72 pp.

Gulland, J.A., 1956. "On the Fishing Effort in English Demersal Fisheries". Fishery Invest., Lond., 2, 20:(5) 40 pp.

Miller, S.M., 1977. Progress report on Shrimp Logbook Analysis. Unpub. ms. Alaska Dept. Fish and Game, Kodiak, Alaska. 35 pp.

Robson, D.S., 1966. Estimation of the Relative Fishing Power of Individual Shrimps. Int. Comm. North West Atl. Fish., Res. Bull. 3:5-14.

Rothschild, B.J., 1977. 'Fishing Effort' in Fish Population Dynamics, 96-115, edited by J.A. Gulland, John Wiley and Sons, N.Y. 372 pp.

Stark, A.E., 1971. A computer programme to Estimate Fishing Power by the Method of Fitting Constants. J. Cons. Int. Explor. Mer. 33:478-482.

V. APPENDIX

V.A. Case Study

```
*DGGAFFNEY/LBOOK/TSS



*****PROGRAM LOGBOOK EXECUTING*****
READY
=$IN RETRV=.T.,TITLE='EXAMPLE OF SYSTEM LBOOK'$



*****SUBROUTINE RETRV EXECUTING****EXAMPLE OF SYSTEM LBOOK
READY
=$IN TYPE=1,FILEIN='DGGAFFNEY/FILEB',FILEOUT='DGGAFFNEY/FILED',
=AREA=107,119$

  549 RECORDS WRITTEN ON FILE DGGAFFNEY/FILED

*****SUBROUTINE RETRV TERMINATING****EXAMPLE OF SYSTEM LBOOK



*****PROGRAM LOGBOOK EXECUTING*****
READY
=$IN REPORT=.T.$



****SUBROUTINE REPORT EXECUTING****
READY
=$IN TYPE=1,FILEIN='DGGAFFNEY/FILED',FILEOUT='DGGAFFNEY/FILEE',
=FACTORS='AREA','GEAR','VESSEL'$
*****REPORT READY---POSITION PAPER
=
```

	TOWS	TOTAL CATCH	UNSTANDARDIZED CPUE MEAN	S.D.
AREA = 119 (WIDE BAY)	309.	2300550.	2976.21	2236.34
GEAR = 27 (DOUBLE DRAGGER)	214.	1855000.	3312.91	2424.53
VESSEL=35266	2.	2500.	625.00	530.33
VESSEL=35167	18.	203000.	4552.67	2870.75
VESSEL=34815	8.	71000.	3854.16	1901.04
VESSEL=33744	13.	126000.	3563.27	1340.91
VESSEL=22294	37.	352000.	2731.02	912.61
VESSEL=21668	12.	157000.	7844.77	3353.92
VESSEL=21652	29.	187000.	1552.61	724.01
VESSEL=21631	11.	80000.	2921.57	1330.58
VESSEL=21591	25.	178000.	3420.68	1863.18

VESSEL= 51		30.	343000.	4738.36	2854.64
VESSEL= 18		19.	96500.	1353.59	564.31
VESSEL= 17		10.	59000.	2287.54	1195.60
GEAR = 17	(BEAM TRAWL)	6.	20500.	1101.18	343.95
VESSEL= 6218		6.	20500.	1101.18	343.95
GEAR = 7	(SINGLE DRAGGER)	89.	425050.	2293.02	1510.57
VESSEL=17391		37.	173000.	2711.99	1386.47
VESSEL=16893		25.	48050.	908.40	613.39
VESSEL= 6334		27.	204000.	3000.92	1449.31

	TOWS	TOTAL CATCH	UNSTANDARDIZED CPUE MEAN	S.D.
AREA = 117 (S. SHELIKOF)	131.	1312900.	5324.36	3761.26
GEAR = 27 (DOUBLE DRAGGER)	111.	1202400.	5717.68	3913.95
VESSEL=35266	15.	151000.	9947.67	3438.48
VESSEL=35167	13.	99000.	2589.23	1776.71
VESSEL=21668	18.	221000.	5425.65	2854.16
VESSEL=21652	11.	100000.	2782.71	1162.10
VESSEL=21631	7.	91000.	8806.97	4018.92
VESSEL= 1877	18.	196400.	4352.57	1846.33
VESSEL= 51	29.	344000.	6328.32	4518.84
GEAR = 7 (SINGLE DRAGGER)	20.	110500.	3141.44	1478.88
VESSEL=17391	13.	72000.	2923.61	1210.78
VESSEL= 6334	7.	38500.	3545.97	1923.56

	TOWS	TOTAL CATCH	UNSTANDARDIZED CPUE MEAN	S.D.
AREA = 112 (UGANIK BAY)	4.	150.	16.65	23.57
GEAR = 27 (DOUBLE DRAGGER)	2.	0.	0.	0.
VESSEL=21668	2.	0.	0.	0.
GEAR = 17 (BEAM TRAWL)	2.	150.	33.30	23.62
VESSEL= 6218	2.	150.	33.30	23.62

	TOWS	TOTAL CATCH	UNSTANDARDIZED CPUE MEAN	S.D.
AREA = 110 (UYAK BAY)	55.	129600.	1188.51	734.50
GEAR = 7 (SINGLE DRAGGER)	55.	129600.	1188.51	734.50
VESSEL=16893	15.	23600.	740.74	394.97
VESSEL=16651	40.	106000.	1356.42	764.83

	TOWS	TOTAL CATCH	UNSTANDARDIZED CPUE MEAN	S.D.
AREA = 109 (ALITAK BAY)	50.	369000.	3280.19	2273.85
GEAR = 27 (DOUBLE DRAGGER)	33.	288000.	3787.28	2259.23
VESSEL=35167	16.	109000.	3832.20	2765.69
VESSEL=22294	17.	179000.	3745.01	1741.64
GEAR = 7 (SINGLE DRAGGER)	17.	81000.	2295.82	2016.47
VESSEL= 6361	17.	81000.	2295.82	2016.47

549 RECORDS WERE READ FROM FILE DGGAFFNEY/FILED
0 RECORDS WERE UNCLASSIFIABLE

****SUBROUTINE REPORT TERMINATING****

*****PROGRAM LOGBOOK EXECUTING*****
READY
=$IN STAND=.T.$

```
***SUBROUTINE STANDARD EXECUTING***
READY
=$IN FILEIN='DGGAFFNEY/FILEE',MODEL=1$
*****REPORT READY---POSITION PAPER
=
```

ANALYSIS OF VARIANCE FOR FITTING MODEL

SOURCE	D.F.	SUMS OF SQUARES	MEAN SQUARE	F-STATISTIC
MODEL	20	18470.43	923.52	1909.6187
ERROR	289	139.76	0.48	
TOTAL	309	18610.19		

MULTIPLE CORRELATION COEFFICIENT 0.992

	MODEL 1 STANDARDIZATION COEFFICIENT	CPUE
AREA = 119 (WIDE BAY)	7.3597	1609.3
GEAR = 27 (DOUBLE DRAGGER)	0.4219	2389.0
VESSEL=35266	-1.5666	560.7
VESSEL=35167	0.3850	3545.3
VESSEL=34815	0.3846	3603.9
VESSEL=33744	0.3261	3359.6

VESSEL=22294	0.0643	2554.9
VESSEL=21668	1.0850	7187.5
VESSEL=21652	-0.5506	1383.9
VESSEL=21631	0.0229	2489.4
VESSEL=21591	0.1864	2895.6
VESSEL= 51	0.5017	3962.7
VESSEL= 18	-0.6553	1251.8
VESSEL= 17	-0.1836	2029.3
GEAR = 17 (BEAM TRAWL)	-0.4030	1284.6
VESSEL= 6218	-0.0000	1008.6
GEAR = 7 (SINGLE DRAGGER)	-0.0189	1546.2
VESSEL=17391	0.4106	2277.9
VESSEL=16893	-0.9328	596.3
VESSEL= 6334	0.5222	2553.1

ANALYSIS OF VARIANCE FOR FITTING MODEL

SOURCE	D.F.	SUMS OF SQUARES	MEAN SQUARE	F-STATISTIC
MODEL	12	9107.90	758.99	1817.3866
ERROR	119	49.70	0.42	
TOTAL	131	9157.60		

MULTIPLE CORRELATION COEFFICIENT 0.995

	MODEL 1 STANDARDIZATION COEFFICIENT	CPUE
AREA = 117 (S. SHELIKOF)	8.1575	3618.0
GEAR = 27 (DOUBLE DRAGGER)	0.2315	4361.3
VESSEL=35266	0.7310	9127.1
VESSEL=35167	-0.8105	1957.9
VESSEL=21668	0.0610	4659.6
VESSEL=21652	-0.5717	2493.2
VESSEL=21631	0.5753	7936.3
VESSEL= 1877	-0.1445	3793.8
VESSEL= 51	0.1593	5118.2
GEAR = 7 (SINGLE DRAGGER)	-0.2315	2800.4
VESSEL=17391	-0.0568	2522.7
VESSEL= 6334	0.0568	2865.5

ANALYSIS OF VARIANCE FOR FITTING MODEL

SOURCE	D.F.	SUMS OF SQUARES	MEAN SQUARE	F-STATISTIC
MODEL	5	22.34	4.47	-6.8514
ERROR	-1	0.65	-0.65	
TOTAL	4	22.99		

MULTIPLE CORRELATION COEFFICIENT 0.972

	MODEL 1 STANDARDIZATION COEFFICIENT	CPUE
AREA = 112 (UGANIK BAY)	1.6710	4.5
GEAR = 27 (DOUBLE DRAGGER)	-1.6710	0.7
VESSEL=21668	0.	1.0
GEAR = 17 (BEAM TRAWL)	1.6710	20.4
VESSEL= 6218	0.	28.3

ANALYSIS OF VARIANCE FOR FITTING MODEL

SOURCE	D.F.	SUMS OF SQUARES	MEAN SQUARE	F-STATISTIC
MODEL	4	2477.06	619.27	290.1861
ERROR	51	108.84	2.13	
TOTAL	55	2585.90		

MULTIPLE CORRELATION COEFFICIENT 0.958

	MODEL 1 STANDARDIZATION COEFFICIENT	CPUE
AREA = 110 (UYAK BAY)	6.6406	1746.4
GEAR = 7 (SINGLE DRAGGER)	0.0000	600.8
VESSEL=16893	-0.1519	414.2
VESSEL=16651	0.1519	536.8

ANALYSIS OF VARIANCE FOR FITTING MODEL

SOURCE	D.F.	SUMS OF SQUARES	MEAN SQUARE	F-STATISTIC
MODEL	6	3111.94	518.66	1216.3614
ERROR	44	18.76	0.43	
TOTAL	50	3130.70		

MULTIPLE CORRELATION COEFFICIENT 0.994

	MODEL 1 STANDARDIZATION COEFFICIENT	CPUE
AREA = 109 (ALITAK BAY)	7.7876	2520.8
GEAR = 27 (DOUBLE DRAGGER)	0.2994	3272.8
VESSEL=35167	-0.0307	3029.9
VESSEL=22294	0.0307	3219.2
GEAR = 7 (SINGLE DRAGGER)	-0.2994	2013.0
VESSEL= 6361	0.0000	1626.5

SUBROUTINE STANDARD TERMINATING

*****PROGRAM LOGBOOK EXECUTING*****
READY
=$IN FINISH=.T.$

****PROGRAM LOGBOOK TERMINATING****
*

01/11/79 14.23

```
00010   C      PROGRAM LOGBOOK
00020   C
00030   C-PURPOSE-----------LOGBOOK IS A SYSTEM DESIGNED TO EDIT,
00040   C                    UPDATE, RETRIEVE, SUMMERIZE, AND
00050   C                    STANDARDIZE SHRIMP LOGBOOK DATA.
00060   C
00070   C-PROGRAMMER--------SUZANNE M MILLER    NOVEMBER 1977
00080   C                    ALASKA DEPT FISH & GAME    KODIAK
00090   C
00100   C-FILES-------------
00110   C      11        INPUT--UNEDTIED DATA RECORDS
00120   C      12        IN/OUTPUT--EDITED DATA RECORDS
00130   C      13        INPUT/OUTPUT--HISTORICAL RECORDS
00140   C      14        INPUT/OUTPUT--RETRIEVAL FILE
00150   C      15        INPUT/OUTPUT--STATISTICAL FILE
00160   C      05        INPUT--PROGRAM DIRECTIVES
00170   C      06        OUTPUT--PROGRAM MESSAGES, REPORTS, ERROR MESSAGES
00180   C      25        IN/OUTPUT--LEGAL VESSELS AND CHARACTERISTICS
00190   C      26        IN/OUTPUT--LEGAL STAT AREA AND CATCH AREAS
00200   C
00210   C-GLOBAL VARIABLES--
00220         COMMON   /IO/ TSS,BATCH,TITLE,WIDTH,SPACE(20000)
00230         CHARACTER  TITLE*80
00240         LOGICAL     TSS,BATCH
00250         INTEGER  SPACE,WIDTH
00260   C     DEFINITIONS
00270   C          TITLE     JOB TITLE
00280   C          TSS       TRUE IF OPERATING IN TIMESHARING MODE
00290   C          BATCH     TRUE IF OPERATING IN BATCH MODE
00300   C          WIDTH     MAXIMIN NUMBER OF CHARACTERS PER OUTPUT LINE
00310   C          SPACE     WORK SPACE
00320   C
00330   C-LOCAL VARIABLES---
00340         LOGICAL    EDIT,UPDATE,REPORT,RETRV,STAND,FINISH
00350         CHARACTER  BLANK*80
00360   C     DEFINITIONS
00370   C          EDIT      TRUE TO CALL EDIT SUBROUTINE
00380   C          UPDATE    TRUE TO CALL UPDATE SUBROUTINE
00390   C          REPORT    TRUE TO CALL REPORT SUBROUTINE
00400   C          RETRV     TRUE TO CALL RETRV SUBROUTINE
00410   C          STAND     TRUE TO CALL STANDARDIZATION SUBROUTINE
00420   C          FINISH    TRUE TO TERMINATE PROGRAM
00430   C          BLANK     CHARACTER CONSTANT
00440   C
00450   C-INPUT-------------
00460         NAME LIST /IN/ EDIT,UPDATE,REPORT,RETRV,STAND,FINISH,
00470        &                        TITLE,WIDTH
00480   C
00490   C-CONSTANTS---------
00500         DATA  BLANK/1H /
00510   C
```

```
00520   C-START-------------
00530         TSS=.T.
00540         BATCH=.F.
00550         FINISH=.F.
00560         WIDTH=136
00570      10 TITLE=BLANK
00580         EDIT=.F.
00590         REPORT=.F.
00600         RETRV=.F.
00610         STAND=.F.
00620         UPDATE=.F.
00630         IF(BATCH) GO TO 15
00640         CALL FPARAM(1,120)
00650      15 WRITE(6,600)
00660         IF(TSS) WRITE(6,603)
00670         READ(5,IN)
00680         IF(EDIT) CALL EDITOR
00690         IF(UPDATE) CALL ADDITION
00700         IF(RETRV) CALL RETRIEVE
00710         IF(REPORT) CALL SUMMARY
00720         IF(STAND) CALL STANDARD
00730         IF(.NOT.FINISH) GO TO 10
00740         WRITE(6,605)
00750         STOP
00760   C
00770   C-FORMATS-----------
00780     600 FORMAT(1H1,/////' *****PROGRAM LOGBOOK EXECUTING*****'/)
00790     603 FORMAT(1H ,'READY')
00800     605 FORMAT(//1H ,'****PROGRAM LOGBOOK TERMINATING****')
00810         END
00820         SUBROUTINE GETFILE(FNAME,FNUM)
00830   C
00840   C-PURPOSE-----------TO ATTACH A FILE
00850   C
00860   C-ARGUMENTS----------
00870         CHARACTER  FNAME*80
00880         INTEGER    FNUM
00890   C     DEFINITIONS
00900   C         FNAME  NAME OF FILE TO BE ATTACHED
00910   C         FNUM   LOGICAL UNIT NUMER OF FILE
00920   C
00930   C-LOCAL VARIABLES---
00940         CHARACTER  BLANK*1,SEMIC*1
00950         INTEGER    POS,OK1,OK2
00960   C     DEFINITIONS
00970   C         BLANK  CHARACTER CONSTANT
00980   C         SEMIC  CHARACTER CONSTANT
00990   C         POS    POSITION OF FIRST BLANK IN FNAME
01000   C         OK1    CHECK VALUE
01010   C         OK2    CHECK VALUE
01020   C
01030   C-CONSTANTS---------
01040         DATA BLANK/1H /,SEMIC/1H;/,OK1/0400000000000/,OK2/0403700000000/
01050   C
01060   C-START-------------
01070         POS=0
01080      20 POS=POS+1
01090         II=KOMPCH(FNAME,POS,BLANK,1,1)
```

```
01100          IF(II.EQ.0) GO TO 30
01110          GO TO 20
01120       30 IF(POS.EQ.1) GO TO 40
01130          CALL CONCAT(FNAME,POS,SEMIC,1,1)
01140          CALL ATTACH(FNUM,FNAME,3,0,ISTAT, )
01150          CALL CONCAT(FNAME,POS,BLANK,1,1)
01160          IF(ISTAT.EQ.0.OR.ISTAT.EQ.OK1.OR.ISTAT.EQ.OK2) RETURN
01170       40 WRITE(6,600) FNAME
01180          STOP
01190    C
01200    C-FORMATS-----------
01210      600 FORMAT(1H ,'UNABLE TO ATTACH FILE---',A80/
01220         &       1H ,'PROGRAM TERMINATES')
01230          END
01240          SUBROUTINE LABEL(FNUM,FVALUE,VLABEL)
01250    C
01260    C-PURPOSE-----------TO ASSIGN A LABEL TO FACTOR VALUES
01270    C
01280    C-ARGUMENTS----------
01290          CHARACTER   VLABEL*17
01300          INTEGER FNUM,FVALUE
01310    C     DEFINITIONS
01320    C         FNUM       FACTOR CODE
01330    C         FVALUE     FACTOR VALUE
01340    C         VLABEL     VALUE LABEL
01350    C
01360    C-LOCAL VARIABLES---
01370          CHARACTER *17 BLANK,AREANAME(45),GEARNAME(3),MNAME(12)
01380          INTEGER     AREA(45),GEAR(3),MONTH(12),YEAR1,YEAR2
01390    C     DEFINITIONS
01400    C         BLANK      CHARACTER CONSTANT
01410    C         AREANAME   CATCH AREA NAMES
01420    C         GEARNAME   GEAR TYPES
01430    C         MNAME      NAMES OF THE MONTHS
01440    C         AREA       AREA CODES
01450    C         GEAR       GEAR CODES
01460    C         MONTH      MONTH CODES
01470    C         YEAR1      BEGINNING OF SEASON
01480    C         YEAR2      ENDING OF SEASON
01490    C
01500    C-CONSTANTS----------
01510          DATA (AREA(I),I=1,45)/101,102,103,104,105,106,107,
01520         &     108,109,110,111,113,114,115,116,117,118,119,
01530         &     120,201,202,203,204,205,206,207,208,209,210,
01540         &     211,212,213,214,215,216,217,218,219,220,221,
01550         &     301,302,303,112,121/
01560          DATA (AREANAME(I),I=1,45)
01570         &/17H (NORTH AFOGNAK ),17H (INNER MARMOT  ),17H (OUTER MARMOT  ),
01580         & 17H (CHINIAK BAY   ),17H (CHINIAK       ),17H (UGAK BAY      ),
01590         & 17H (KILIUDA BAY   ),17H (TWOHEADED IS. ),17H (ALITAK BAY    ),
01600         & 17H (UYAK BAY      ),17H (VIEKODA BAY   ),17H (WEST AFOGNAK  ),
01610         & 17H (KUKAK BAY     ),17H (NO. SHELIKOF  ),17H (C. SHELIKOF   ),
01620         & 17H (S. SHELIKOF   ),17H (IKOLIK        ),17H (WIDE BAY      ),
01630         & 17H (ALBATROSS PK. ),17H (MITROFANIA IS.),17H (STEPOVAK BAY  ),
01640         & 17H (UNGA STRAIT   ),17H (BALBOA BAY    ),17H (WEST NAGAI ST.),
01650         & 17H (PAVLOF BAY    ),17H (CHIGNIK BAY   ),17H (SANAK ISLAND  ),
01660         & 17H (KUJULIK BAY   ),17H (EAST NAGAI ST.),17H (BEAVER BAY    ),
01670         & 17H (BELKOFSKI BAY ),17H (COLD BAY      ),17H (PORT WRANGELL ),
```

```
01680        & 17H (SEMIDI ISLANDS),17H (SUTWIK ISLAND ),17H (IVANOF BAY     )
01690        & 17H (DOLGOI ISLAND ),17H (SEAL CAPE      ),17H (KUIUKTA BAY    )
01700        & 17H (MORZHOVOI BAY ),17H (UNALASKA BAY   ),17H (BEAVER INLET   )
01710        & 17H (CAPE PROMINENC),17H (UGANIK BAY     ),17H (SITKINAK ST.   )
01720         DATA (GEAR(I),I=1,3)/7,17,27/
01730         DATA (GEARNAME(I),I=1,3)/17H (SINGLE DRAGGER),
01740        & 17H (BEAM TRAWL     ),17H (DOUBLE DRAGGER)/
01750         DATA BLANK /1H /
01760         DATA (MONTH(I),I=1,12) /1,2,3,4,5,6,7,8,9,10,11,12/
01770         DATA (MNAME(I),I=1,12)/17H (JANUARY        ),
01780        & 17H (FEBRUARY       ),17H (MARCH          ),17H (APRIL          )
01790        & 17H (MAY            ),17H (JUNE           ),17H (JULY           )
01800        & 17H (AUGUST         ),17H (SEPTEMBER      ),17H (OCTOBER        )
01810        & 17H (NOVEMBER       ),17H (DECEMBER       )/
01820   C
01830   C-INITIALIZATION----
01840         VLABEL=BLANK
01850   C
01860   C-START-------------
01870         GO TO (10,20,30,40,50,50,50), FNUM
01880      10 DO 15 I=1,45
01890      15     IF(FVALUE.EQ.AREA(I)) VLABEL=AREANAME(I)
01900         RETURN
01910      20 YEAR1=FVALUE/100
01920         YEAR2=FVALUE-YEAR1*100
01930         ENCODE(VLABEL,1000) YEAR1,YEAR2
01940      30 DO 35 I=1,3
01950      35     IF(FVALUE.EQ.GEAR(I)) VLABEL=GEARNAME(I)
01960         RETURN
01970      40 DO 45 I=1,12
01980      45     IF(FVALUE.EQ.MONTH(I)) VLABEL=MNAME(I)
01990      50 RETURN
02000   C
02010   C-FORMATS-----------
02020    1000 FORMAT(' (19',I2,'-19',I2,')')
02030         END
02040         SUBROUTINE EDITOR
02050   C
02060   C-PURPOSE-----------READ, EDIT AND AUGMENT LOGBOOK
02070   C                     RECORDS TO BE WRITTEN ON OUTPUT FILE
02080   C
02090   C-GLOBAL VARIABLES--
02100         COMMON  /IO/ TSS,BATCH,TITLE,WIDTH,SPACE(20000)
02110         LOGICAL TSS,BATCH
02120         CHARACTER TITLE*80
02130         INTEGER SPACE,WIDTH
02140   C     DEFINITIONS
02150   C         SEE MAIN PROGRAM
02160   C
02170   C-LOCAL VARIABLES---
02180         LOGICAL      ERRORS,LIST,IGNORE
02190         CHARACTER *80 FILEIN,FILEOUT,VESSELS,GCAREAS,BLANK,
02200        &             VCHAR*10(200),RIN*52,ROUT*72,INFO*10
02210         INTEGER      SAREA(200),CAREA(200),V(200),NTOTAL,NBAD,VES,
02220        &             GEAR,GEARCK,VL,HP,HOUR,MIN,POUNDS,NRITE,NOUT,
02230        &             AREA,NUMV,NUMA,TIME,EFFORT,CPE,LCPE,GCA,FISH
02240   C     DEFINITIONS
02250   C         ERRORS     TRUE IF ERRORS ENCOUNTERED
```

```
02260   C          LIST       TRUE IF LISTING OF OUTPUT RECORDS DESIRED
02270   C          IGNORE     TRUE IF RECORDS WITH ERRORS TO BE OUTPUT
02280   C          FILEIN     INPUT FILE
02290   C          FILEOUT    OUTPUT FILE
02300   C          VESSELS    FILE CONTAINING VESSEL INFORMATION
02310   C          GCAREA     FILE CONTAINING AREA INFORMATION
02320   C          RIN        INPUT RECORD
02330   C          ROUT       OUTPUT RECORD
02340   C          BLANK      CHARACTER CONSTANA
02350   C          VCHAR      VESSEL CHARACTERISTICS FROM VESSELS FILE
02360   C           FISH       % FISH IN TOW
02370   C          SAREA      STATISTICAL AREAS
02380   C          CAREA      CATCH AREAS CORRESPONDING TO STAT AREAS
02390   C          V          VESSEL NUMBER FORM VESSELS FILE
02400   C          NTOTAL     TOTAL NUMBER OF RECORDS READ
02410   C          NRITE      NUMBER OF NEW RECORDS ON OUTPUT FILE
02420   C          NOUT       TOTAL NUMBER OF RECORDS ON OUTPUT TILE
02430   C          NBAD       TOTAL NUMBER OF RECORDS WITH ERRORS
02440   C          VES        VESSEL NUMBER FORM INPUT RECORD
02450   C          GEAR       GEAR TYPE FORM INPUT RECORD
02460   C          GEARCK     GEAR TYPE FORM VESSELS FILE
02470   C          VL         VESSEL LENGTH FROM VESSELS FILE
02480   C          HP         HORSE POWER FORM VESSELS FILE
02490   C          HOUR       DURATION OF TOW
02500   C          MIN        CURATION OF TOW
02510   C          POUNDS     POUNDS OF SHRIMP HAILED
02520   C          AREA       STAT AREA FROM INPUT RECORD
02530   C          NUMV       NUMBER OF VESSELS IN VESSELS FILE
02540   C          NUMA       NUMBER OF STAT AREAS IN AREAS FILE
02550   C          EFFORT     TOTAL DURATION OF TOW IN HOURS
02560   C          CPE        CATCH PER UNIT EFFORT
02570   C          YEAR       YEAR OF DRAG FROM INPUT RECORD
02580   C          INFO       STRING INFORMATION FROM INPUT RECORD
02590   C          ROPE       FOOTROPE LENGTH FROM INPUT RECORD
02600   C          TIME       STARTING AND ENDING TIME OF TOW
02610   C          LCPE       NATURAL LOG OF CPE
02620   C          GCA        GEOGRAPHIC CATCH AREA
02630   C
02640   C-INPUT-------------
02650         NAMELIST /IN/ TITLE,FILEIN,FILEOUT,VESSELS,GCAREAS,LIST,IGNORE
02660   C
02670   C-EQUIVALENCES------
02680         EQUIVALENCE (SPACE(1),CAREA(1)),(SPACE(201),SAREA(1)),
02690        &                (SPACE(401),V(1))
02700   C
02710   C-CONSTANTS----------
02720         DATA BLANK /1H /
02730   C
02740   C-INIALIZATION-------
02750         FILEIN=BLANK
02760         FILEOUT=BLANK
02770         VESSELS='DGGAFFNEY/LBOOK/BOATS$BOATS,R'
02780         GCAREAS='DGGAFFNEY/LBOOK/DISTRICT$DISTRICT,R'
02790         NTOTAL=0
02800         NBAD=0
02810         NRITE=0
02820         NOUT=0
02830         LIST=.F.
```

```
02840          IGNORE=.F.
02850    C
02860    C-START-------------
02870          WRITE(6,600) TITLE
02880          IF(TSS) WRITE(6,605)
02890          READ(5,IN)
02900          IF(BATCH) GO TO 10
02910          CALL GETFILE(FILEIN,11)
02920          CALL GETFILE(VESSELS,25)
02930          CALL GETFILE(GCAREAS,26)
02940          IF(FILEOUT.EQ.BLANK) GO TO 20
02950          CALL GETFILE(FILEOUT,12)
02960       10 READ(12,END=15)
02970          NOUT=NOUT+1
02980          GO TO 10
02990       15 BACKSPACE 12
03000       20 I=0
03010       25 I=I+1
03020             READ(25,1500,END=30) V(I),VCHAR(I)
03030             GO TO 25
03040       30 NUMV=I
03050          IF(TSS) CALL DETACH(25,ISTAT,)
03060          I=0
03070       35 I=I+1
03080             READ(26,1600,END=40) SAREA(I),CAREA(I)
03090             GO TO 35
03100       40 NUMA=I
03110          IF(TSS) CALL DETACH(26,ISTAT,)
03120       45 ERRORS=.F.
03130          READ(11,1100,END=100) RIN
03140          NTOTAL=NTOTAL+1
03150          DECODE(RIN,1000) YEAR,VES,GEAR,ROPE,MO,DAY,TIME,
03160         & HOUR,MIN,INFO,POUNDS,AREA,FISH
03170          DO 50 I=1,NUMV
03180       50    IF(VES.EQ.V(I)) GO TO 55
03190          CALL ERROR(RIN,1,ERRORS,NTOTAL)
03200          GO TO 60
03210       55 DECODE(VCHAR(I),1010) GEARCK,HP,VL
03220          IF(GEAR.NE.GEARCK) CALL ERROR(RIN,2,ERRORS,NTOTAL)
03230       60 DO 65 I=1,NUMA
03240       65    IF(AREA.EQ.SAREA(I)) GO TO 70
03250          GCA=0
03260          CALL ERROR(RIN,3,ERRORS,NTOTAL)
03270          GO TO 75
03280       70 GCA=CAREA(I)
03290       75 IF(ERRORS) NBAD=NBAD+1
03300          IF(FILEOUT.EQ.BLANK) GO TO 45
03310          IF(ERRORS.AND.(.NOT.IGNORE)) GO TO 45
03320          EFFORT=IFIX((FLOAT(HOUR)*60.+MIN)/6.)
03330          CPE=0
03340          IF(EFFORT.GT.0.0)
03350         &  CPE=IFIX(FLOAT(POUNDS*100)/FLOAT(EFFORT))
03360          LCPE=0
03370          IF(CPE.GT.0.0)
03380         & LCPE=IFIX(ALOG(FLOAT(CPE/10))*1000.)
03390          ENCODE(ROUT,1020) GCA,YEAR,GEAR,MO,DAY,
03400         &AREA,VES,HP,VL,ROPE,TIME,EFFORT,INFO,POUNDS,CPE,LCPE,FISH
03410          WRITE(12,1200) ROUT
```

```
03420          NRITE=NRITE+1
03430          NOUT=NOUT+1
03440          GO TO 45
03450        100 NOUT=NOUT-1
03460          IF(FILEOUT.EQ.BLANK) GO TO 120
03470          IF(.NOT.LIST) GO TO 120
03480          REWIND 12
03490          WRITE(6,610) FILEOUT
03500        110 READ(12,1200,END=120) ROUT
03510          WRITE(6,615) ROUT
03520          GO TO 110
03530        120 IF(BATCH) GO TO 140
03540          CALL DETACH(11,ISTAT,)
03550          IF(FILEOUT.EQ.BLANK) GO TO 140
03560          CALL DETACH(12,ISTAT,)
03570        140 WRITE(6,620) NTOTAL,FILEIN,NBAD,NRITE,FILEOUT,NOUT,TITLE
03580          RETURN
03590     C
03600     C-FORMATS-----------
03610        600 FORMAT(1H1,/////' *****SUBROUTINE EDIT EXECUTING*****',A80)
03620        605 FORMAT(1H ,'READY')
03630        610 FORMAT(1H1,//' LISTING OF FILE ',A80)
03640        615 FORMAT(1H ,A70)
03650        620 FORMAT(//1H ,I5,' RECORDS READ FROM FILE ',A80/
03660           &          1H ,I5,' RECORDS IN ERROR'//
03670           &          1H ,I5,' NEW RECORDS WRITTEN ON FILE ',A80/
03680           &          1H ,I5,' RECORDS IN TOTAL'//
03690           &          1H ,'****SUBROUTINE EDIT TERMINATING****',A80)
03700       1000 FORMAT(I4,I5,I2,I3,2I2,I8,I1,I2,A10,2I5,I3)
03710       1010 FORMAT(I3,I4,I3)
03720       1020 FORMAT(I3,I4,3I2,2I5,3I3,I8,I3,A10,I5,I6,I5,I3)
03730       1100 FORMAT(A52)
03740       1200 FORMAT(A72)
03750       1500 FORMAT(I5,A10)
03760       1600 FORMAT(I5,I4)
03770          END
03780          SUBROUTINE ERROR(RECORD,ENUM,ERRORS,NR)
03790     C
03800     C-PURPOSE-----------PRINTS A RECORD IN ERROR AND
03810     C                   MESSAGE CORRESPONDING TO ERROR NUMBER
03820     C
03830     C-ARGUMENTS---------
03840          CHARACTER  RECORD*52
03850          INTEGER    ENUM,NR
03860          LOGICAL    ERRORS
03870     C    DEFINITIONS
03880     C        RECORD    RECORD IN ERROR
03890     C        ENUM      ERROR NUMBER
03900     C        ERRORS    TRUE TO INDICATE ERROR DETECTED
03910     C         NR        RECORD NUMBER
03920     C
03930     C-LOCAL VARIABLES---
03940          CHARACTER  MESSAGE*30(3),LRECORD*52
03950     C    DEFINITIONS
03960     C        MESSAGE   ERROR MESSAGE
03970     C        LRECORD   LAST RECORD WITH ERROR
03980     C
03990     C-CONSTANTS---------
```

```
04000          DATA MESSAGE(1)/30H INVALID ADF&G VESSEL NUMBER  /,
04010         &     MESSAGE(2)/30H INVALID GEAR TYPE            /,
04020         &     MESSAGE(3)/30H INVALID STATISTICAL AREA     /
04030    C
04040    C-START-------------
04050          ERRORS=.T.
04060          IF(RECORD.NE.LRECORD) WRITE(6,605) RECORD
04070          WRITE(6,600) NR, MESSAGE(ENUM)
04080          LRECORD=RECORD
04090          RETURN
04100    C
04110    C-FORMATS-----------
04120      600 FORMAT(1H '*****RECORD ',I4,2X,A30)
04130      605 FORMAT(/1H ,A49)
04140          END
04150          SUBROUTINE ADDITION
04160    C
04170    C-PURPOSE-----------ACCUMULATE EDITED LOGBOOK RECORDS
04180    C                    AND CREATE FILE OF DAILY TOTALS
04190    C
04200    C-GLOBAL VARIABLES--
04210          COMMON /IO/ TSS,BATCH,TITLE,WIDTH,SPACE(20000)
04220          LOGICAL TSS,BATCH
04230          CHARACTER  TITLE*80
04240          INTEGER SPACE,WIDTH
04250    C     DEFINITIONS    SEE MAIN PROGRAM
04260    C
04270    C-LOCAL VARIABLES---
04280          CHARACTER  *80 FILEIN,FILEOUT,BLANK,ACCUM(1000),LOG*72,
04290         &               KEY*13
04300          INTEGER     NR,IR,NTOTAL,NTOWS
04310          REAL        POUNDS,LBS,CPUE,CPE,LCPUE,LCPE,SSP,SSC,SSL
04320          LOGICAL     LIST
04330    C     DEFINITIIONS
04340    C         FILEIN    INPUT FILE---EDITED LOGBOOK RECORDS
04350    C         FILEOUT   OUTPUT TILE--ACCUMULATED DAILY TOTALS
04360    C         BLANK     CHARACTER CONSTANT
04370    C         ACCUM     ACCUMULATED DAILY TOTALS
04380    C         LOG       EDITED LOGBOOK RECORD
04390    C         KEY       ACCUMULATION KEY--AREA,SEASON,MO,DAY,GEAR
04400    C         NR        NUMBER OF CELLS IN ACCUM
04410    C         IR        CALL IN ACCUM MATCHING KEY
04420    C         NTOTAL    NUMBER OF INPUT RECORDS
04430    C         NTOWS     NUMBER OF TOWS--DAILY TOTAL
04440    C         POUNDS    POUNDS CAUGHT--DAILY TOTAL
04450    C         LBS       POUNDS CAUGHT--INDIVIDUAL DRAG
04460    C         CPUE      TOTAL CPUE--DAILY TOTAL
04470    C         CPE       CPUE--INDIVIDUAL DRAG
04480    C         LCPUE     TOTAL LOG(CPUE)--DAILY TOTAL
04490    C         LCPE      LOG(CPUE)--INDIVIDUAL DRAG
04500    C         SSP       SUMS OF POUNDS**2--DAILY TOTAL
04510    C         SSC       SUMS OF CPUE**2--DAILY TOTAL
04520    C         SSL       SUMS OF LOG(CPUE)**2--DAILY TOTAL
04530    C         LIST      TRUE IF LISTING OF OUTPUT FILE DESIRED
04540    C
04550    C-EQUIVALENCES------
04560          EQUIVALENCE (SPACE(1),ACCUM(1))
04570    C
```

```
04580   C-INPUT-------------
04590         NAMELIST /IN/ TITLE,FILEIN,FILEOUT,LIST
04600   C
04610   C-CONSTANTS---------
04620          DATA BLANK/1H /
04630   C
04640   C-INITALIZATION------
04650         NTOTAL=0
04660         NR=0
04670         FILEIN=BLANK
04680         FILEOUT=BLANK
04690         LIST=.F.
04700   C
04710   C-START-------------
04720         WRITE(6,600) TITLE
04730         IF(TSS) WRITE(6,605)
04740         READ(5,IN)
04750         IF(BATCH) GO TO 10
04760         CALL GETFILE(FILEIN,12)
04770         CALL GETFILE(FILEOUT,13)
04780      10 NR=NR+1
04790      20 ACCUM(NR)=BLANK
04800      30 READ(12,1200,END=80) LOG
04810         NTOTAL=NTOTAL+1
04820         DECODE(LOG,1000) KEY,LBS,CPE,LCPE
04830         DO 40 I=1,NR
04840         IR=NR-I+1
04850         II=KOMPCH(ACCUM(IR),1,KEY,1,13)
04860      40 IF(II.EQ.0) GO TO 50
04870         IF(NR.LT.1000) GO TO 45
04880         WRITE(6,625)
04890         GO TO 30
04900      45 IR=NR
04910         ACCUM(IR)=BLANK
04920         NR=NR+1
04930      50 DECODE(ACCUM(IR),1010)NTOWS,POUNDS,SSP,CPUE,SSC,LCPUE,SSL
04940         NTOWS=NTOWS+1
04950         POUNDS=POUNDS+LBS
04960         SSP=SSP+LBS**2
04970         CPUE=CPUE+CPE
04980         SSC=SSC+CPUE**2
04990         LCPUE=LCPUE+LCPE
05000         SSL=SSL+LCPE**2
05010         ENCODE(ACCUM(IR),1020)KEY,NTOWS,POUNDS,SSP,CPUE,SSC,LCPUE,SSL
05020         GO TO 30
05030      80 NR=NR-1
05040         REWIND 13
05050         DO 90 I=1,NR
05060      90 WRITE(13,1300) ACCUM(I)
05070         REWIND 13
05080         IF(BATCH) GO TO 95
05090         CALL DETACH(12,ISTAT,)
05100         CALL DETACH(13,ISTAT,)
05110      95 IF(.NOT.LIST) GO TO 110
05120         WRITE(6,610) FILEOUT
05130         DO 100 I=1,NR
05140     100 WRITE(6,615) ACCUM(I)
05150     110 WRITE(6,620) NTOTAL,FILEIN,NR,FILEOUT,TITLE
```

```
05160          RETURN
05170    C
05180    C-FORMAT----------
05190      600 FORMAT(1H1/////' ****SUBROUTINE UPDATE EXECUTING****',A80)
05200      605 FORMAT(1H ,'READY')
05210      610 FORMAT(1H1//' LISTING OF FILE ',A80)
05220      615 FORMAT(1H ,A80)
05230      620 FORMAT(//1H ,I5,' RECORDS READ FROM FILE ',A80/
05240         &          1H ,I5,' RECORDS WRITTEN ON FILE ',A80//
05250         &          1H ,'****SUBROUTINE UPDATE TERMINATING****',A80)
05260      625 FORMAT(//1H ,'*****WARNING----NUMBER OF CELLS EXCEEDS 1000')
05270     1000 FORMAT(A13,40X,F5.0,F6.1,F5.3)
05280     1010 FORMAT(13X,I3,6E10.4)
05290     1020 FORMAT(A13,I3,6E10.4)
05300     1200 FORMAT(A72)
05310     1300 FORMAT(A80)
05320          END
05330          SUBROUTINE RETRIEVE
05340    C
05350    C-PURPOSE-----------RETRIEVE RECORDS FROM HISTORICAL FILE IN
05360    C                    A MANNER SPECIFIED BY THE USER
05370    C
05380    C-GLOBAL VARIABLES--
05390          COMMON /IO/ TSS,BATCH,TITLE,WIDTH,SPACE(20000)
05400          LOGICAL TSS,BATCH
05410          CHARACTER TITLE*80
05420          INTEGER SPACE,WIDTH
05430    C     DEFINITIONS   SEE MAIN PROGRAM
05440    C
05450    C-LOCAL VARIABLES---
05460          CHARACTER *80 RECORD,FILEIN,FILEOUT,BLANK
05470          INTEGER   VAL(2,7),KEY(7),AREA(2),SEASON(2),
05480         &           GEAR(2),MONTH(2),DAY(2),SAREA(2),VESSEL(2),
05490         &           NUM(2),TYPE,NR
05500          LOGICAL   LIST
05510    C     DEFINITIONS
05520    C         RECORD    RETRIEVED INFORMATION
05530    C         FILEIN    INPUT FILE
05540    C         FILEOUT   OUTPUT FILE
05550    C         BLANK     CHARACTER CONSTANT
05560    C         VAL       RETRIEVAL VALUES FOR EACH VARIABLE
05570    C         KEY       VALUE OF VARIABLES FROM INPUT RECORD
05580    C         AREA     MIN,MAX VALUES FOR CATCH AREA
05590    C         SEASON    MIN,MAX VALUES FOR SEASON
05600    C         GEAR      MIN,MAX VALUES FOR GEAR
05610    C         MONTH     MIN,MAX VALUES FOR MONTH
05620    C         DAY       MIN,MAX VALUES FOR DAY
05630    C         SAREA     MIN,MAX VALUES FOR STATISTICAL AREA
05640    C         VESSEL    MIN,MAX VALUES FOR VESSEL
05650    C         NUM       NUMBER OF VARIABLES FOR EACH TYPE OF INPUT FILE
05660    C         TYPE      TYPE OF INPUT FILE
05670    C         NR        NUMBER OF RECORDS RETRIEVED
05680    C         LIST      TRUE TO LIST RETRIEVED RECORDS
05690    C
05700    C-EQUIVALENCES------
05710          EQUIVALENCE (SPACE(1),VAL(1,1))
05720          EQUIVALENCE (AREA,VAL(1,1)),
05730         &            (SEASON,VAL(1,2)),(GEAR,VAL(1,3)),
```

```
05740       &              (MONTH,VAL(1,4)),(DAY,VAL(1,5)),
05750       &             (SAREA,VAL(1,6)),(VESSEL,VAL(1,7))
05760   C
05770   C-INPUT-------------
05780         NAMELIST /IN/ FILEIN,FILEOUT,AREA,GEAR,SEASON,
05790       &               MONTH,DAY,SAREA,VESSEL,LIST,TYPE,TITLE
05800   C
05810   C-CONSTANTS---------
05820         DATA  NUM/7,5/,BLANK/1H /
05830   C
05840   C-INITIALIZATION------
05850         DO 10 I=1,8
05860             VAL(1,I)=0
05870      10     VAL(2,I)=99999
05880         FILEIN=BLANK
05890         FILEOUT=BLANK
05900         LIST=.F.
05910         TYPE=2
05920   C
05930   C-START-------------
05940         WRITE(6,600) TITLE
05950         IF(TSS) WRITE(6,605)
05960         READ(5,IN)
05970         IF(BATCH) GO TO 13
05980         CALL GETFILE(FILEIN,13)
05990         IF(FILEOUT.EQ.BLANK) GO TO 13
06000         CALL GETFILE(FILEOUT,14)
06010      13 REWIND 14
06020      15 READ(14,END=17)
06030         GO TO 15
06040      17 BACKSPACE 14
06050      20 NR=0
06060         IF(LIST) WRITE(6,610) FILEOUT
06070      25 READ(13,3300,END=100) RECORD
06080         DECODE(RECORD,1000) (KEY(I),I=1,NUM(TYPE))
06090         DO 30 I=1,NUM(TYPE)
06100      30     IF(KEY(I).LT.VAL(1,I).OR.KEY(I).GT.VAL(2,I)) GO TO 25
06110         WRITE(14,4400) RECORD
06120         IF(LIST) WRITE(6,615) RECORD
06130         NR=NR+1
06140         GO TO 25
06150     100 WRITE(6,620) NR,FILEOUT
06160         WRITE(6,625) TITLE
06170         IF(BATCH) GO TO 110
06180         CALL DETACH(13,ISTAT,)
06190         IF(FILEOUT.EQ.BLANK) GO TO 110
06200         CALL DETACH(14,ISTAT,)
06210     110 RETURN
06220   C
06230   C-FORMATS-----------
06240       600 FORMAT(1H1,/////' *****SUBROUTINE RETRV EXECUTING****',A80)
06250       605 FORMAT(1H ,'READY')
06260       610 FORMAT(1H1,//' LISTING OF RECORDS WRITTEN ON FILE ',A80)
06270       615 FORMAT(1H ,A80)
06280       620 FORMAT(//1H ,I5,' RECORDS WRITTEN ON FILE ',A80)
06290       625 FORMAT(//1H ,'*****SUBROUTINE RETRV TERMINATING****',A80)
06300      1000 FORMAT(I3,I4,3I2,2I5)
06310      3300 FORMAT(A80)
```

```
 4400 FORMAT(A80)
      END
      SUBROUTINE SUMMARY
C-PURPOSE-----------SUMMERIZE AND REPORT CATCH AND EFFORT STATISTICS
C
C-GLOBAL VARIABLES--
      COMMON /IO/ TSS,BATCH,TITLE,WIDTH,SPACE(20000)
      LOGICAL  TSS,BATCH
      CHARACTER TITLE*80
      INTEGER SPACE,WIDTH
C     DEFINITIONS   SEE MAIN PROGRAM
C
C-LOCAL VARIABLES---
      CHARACTER *80 FILEIN,FILEOUT,BLANK,RECORD,RFMT*51,VLABEL*17,
     &           KEYFMT*5(4),FNAME*6(7),FMT*8(7),FACTORS*6(4),
     &           CELLKEY*36(1000),RKEY*32,ZERO*32,FMTOUT*51(4,2)
      INTEGER   NKEY,TAG,NCELLS,KEYVAL(4),DUMMY,KEY(4),TYPE,NMISS,NR
      INTEGER   ISTAT,SIZE
      REAL      CVALUE(1000,6),RVALUE(6)
      LOGICAL   LIST,TABLES
C     DEFINITIONS
C         FILEIN     INPUT FILE--HISTORICAL FILE OR EDITED LOGS
C         FILEOUT    OUTPUT FILE--CATCH AND EFFORT STATISTICS
C         BLANK      CHARACTER CONSTANT
C         RECORD     INPUT/OUTPUT RECORD
C         RFMT       FORMAT
C         KEYFMT     FORMATS OF KEY FACTORS IN CELLKEY
C         FNAME      POSSIBLE KEY FACTORS FROM INPUT RECORDS
C         FMT        FORMAT OF KEY FACTORS IN INPUT RECORD
C         FACTORS    USER SPECIFIED KEY FACTORS
C         CELLKEY    CELL IDENTIFICATION
C         RKEY       RECORD IDENTIFICATION
C         ZERO       CHARACTER CONSTANT
C         NKEY       NUMBER OF KEY FACTORS
C         TAG        CORRECT CELL NUMBER
C         NCELLS     NUMBER OF CELLS (FACTOR COMBINATIONS)
C         KEYVAL     KEY VALUE OF A FACTOR
C         DUMMY      DUMMY VARIABLE
C         KEY        CODE FOR KEY FACTORS
C         CVALUE     CELL STATISTICS
C         RVALUE     RECORD STATISTICS
C         FMTOUT     OUTPUT FORMATS
C         LIST       TRUE TO LIST OUTPUT FILE
C         TYPE       2=HISTORICAL INPUT FILE,1= EDITED LOGBOOK INPUT
C         TABLES      TRUE FOR PRINTED OUTPUT
C         VLABEL      VARIABLE LABEL
C         NMISS      NUMBER OF MISSING VALUES
C         NR         NUMBER OF RECORDS READ
C         ISTAT      STATUS OF TEMPORARY FILE
C         SIZE       OUTPUT LINE LIMIT INDEX
C
C-EQUIVALENCES------
      EQUIVALENCE (SPACE(1),CVALUE(1,1)),(SPACE(6001),CELLKEY(1))
C
C-INPUT-------------
      NAMELIST /IN/ FILEIN,FILEOUT,LIST,FACTORS,TABLES,TITLE,TYPE
C
C-CONSTANTS---------
```

```
06900          DATA BLANK/1H /,ZERO/32H00000000000000000000000000000000/
06910         &     KEYFMT(1)/5H(I8) /,KEYFMT(2)/5H(2I8)/,
06920         &     KEYFMT(3)/5H(3I8)/,KEYFMT(4)/5H(4I8)/,
06930         &     FNAME(1)/6HAREA  /,FMT(1)/8H(I3)    /,
06940         &     FNAME(2)/6HSEASON/,FMT(2)/8H(3X,I4) /,
06950         &     FNAME(3)/6HGEAR  /,FMT(3)/8H(7X,I2) /,
06960         &     FNAME(4)/6HMONTH /,FMT(4)/8H(9X,I2) /,
06970         &     FNAME(5)/6HDAY   /,FMT(5)/8H(11X,I2)/,
06980         &     FNAME(6)/6HSAREA /,FMT(6)/8H(13X,I5)/,
06990         &     FNAME(7)/6HVESSEL/,FMT(7)/8H(18X,I5)/
07000          DATA
07010         & FMTOUT(1,1)
07020         & /51H(1X,A6,'=',I5/1X,A17,4X,F5.0,2X,F10.0,2(2X,F10.2)) /,
07030         & FMTOUT(2,1)
07040         & /51H(2X,A6,'=',I5/2X,A17,3X,F5.0,2X,F10.0,2(2X,F10.2)) /,
07050         & FMTOUT(3,1)
07060         & /51H(3X,A6,'=',I5/3X,A17,2X,F5.0,2X,F10.0,2(2X,F10.2)) /,
07070         & FMTOUT(4,1)
07080         & /51H(4X,A6,'=',I5/4X,A17,1X,F5.0,2X,F10.0,2(2X,F10.2)) /,
07090         & FMTOUT(1,2)
07100         & /51H(1X,A6,'=',I5,A17,16X,F5.0,2X,F10.0,2(2X,F10.2))   /,
07110         & FMTOUT(2,2)
07120         & /51H(///6X,A6,'=',I5,A17,11X,F5.0,2X,F10.0,2(2X,F10.2))/,
07130         & FMTOUT(3,2)
07140         & /51H(//11X,A6,'=',I5,A17,6X,F5.0,2X,F10.0,2(2X,F10.2)) /,
07150         & FMTOUT(4,2)
07160         & /51H(16X,A6,'=',I5,A17,1X,F5.0,2X,F10.0,2(2X,F10.2))   /
07170    C
07180    C-INIALIZATION------
07190          FILEIN=BLANK
07200          FILEOUT=BLANK
07210        5 FACTORS(1)=FNAME(1)
07220          FACTORS(2)=FNAME(4)
07230          FACTORS(3)=FNAME(3)
07240          FACTORS(4)=' '
07250          NKEY=0
07260          LIST=.F.
07270          TYPE=2
07280          TABLES=.T.
07290          NMISS=0
07300          NR=0
07310          SIZE=2
07320    C
07330    C-START-------------
07340          IF(WIDTH.LT.100) SIZE=1
07350          WRITE(6,600) TITLE
07360          IF(TSS) WRITE(6,605)
07370        6 READ(5,IN)
07380          IF(BATCH) GO TO 10
07390          IF(FILEIN.NE.BLANK) GO TO 7
07400          WRITE(6,603)
07410          GO TO 6
07420        7 CALL GETFILE(FILEIN,14)
07430          IF(FILEOUT.NE.BLANK) GO TO 8
07440          CALL CREATE(15,50000,0,ISTAT)
07450          IF(ISTAT.EQ.0.OR.ISTAT.EQ.2) GO TO 10
07460          WRITE(6,604)
07470          GO TO 6
```

```
07480       8 CALL GETFILE(FILEOUT,15)
07490      10 DO 20 I=1,4
07500             KEY(I)=0
07510             DO 15 J=1,7
07520      15          IF(FACTORS(I).EQ.FNAME(J)) GO TO 18
07530             IF(FACTORS(I).EQ.BLANK) GO TO 25
07540             WRITE(6,610) FACTORS(I)
07550             IF(BATCH) GO TO 100
07560             GO TO 5
07570      18     NKEY=NKEY+1
07580      20     KEY(I)=J
07590      25 NCELLS=1
07600         CELLKEY(1)=ZERO
07610      30 READ(14,1400,END=70) RECORD
07620         NR=NR+1
07630         RKEY=ZERO
07640         DO 35 I=1,NKEY
07650             RFMT=FMT(KEY(I))
07660             DECODE(RECORD,RFMT) KEYVAL(I)
07670             IF(KEYVAL(I).GT.0) GO TO 35
07680             NMISS=NMISS+1
07690             GO TO 30
07700      35     CONTINUE
07710         GO TO (37,36), TYPE
07720      36 DECODE(RECORD,1100) (RVALUE(K),K=1,6)
07730         GO TO 38
07740      37 RVALUE(1)=1
07750         DECODE(RECORD,1105) RVALUE(2),RVALUE(3),RVALUE(5)
07760         RVALUE(4)=RVALUE(3)**2
07770         RVALUE(6)=RVALUE(5)**2
07780      38 DO 60 I=1,NKEY
07790             RFMT=KEYFMT(I)
07800             ENCODE(RKEY,RFMT) (KEYVAL(K),K=1,I)
07810             DO 40 J=1,NCELLS
07820                 II=KOMPCH(RKEY,1,CELLKEY(J),1,32)
07830      40         IF(II.EQ.0) GO TO 55
07840             NCELLS=NCELLS+1
07850             IF(NCELLS.LE.1000) GO TO 45
07860             WRITE(6,615)
07870             NCELLS=500
07880             GO TO 30
07890      45     CELLKEY(NCELLS)=ZERO
07900             ENCODE(CELLKEY(J),1110) RKEY,J
07910             DO 50 K=1,6
07920      50         CVALUE(J,K)=0
07930      55     DO 60 K=1,6
07940      60         CVALUE(J,K)=CVALUE(J,K)+RVALUE(K)
07950         GO TO 30
07960      70 NCELLS=NCELLS-1
07970         CALL SORTD(CELLKEY,NCELLS,9,0,1,2,3,4,5,6,7)
07980         IF(BATCH) GO TO 75
07990         IF(.NOT.TABLES) GO TO 75
08000         WRITE(6,620)
08010         READ(5,500) DUMMY
08020      75 DO 90 I=1,NCELLS
08030             DECODE(CELLKEY(I),1120) (KEYVAL(K),K=1,4),TAG
08040             IF(CVALUE(TAG,1).GT.1) GO TO 77
08050             CVALUE(TAG,4)=0
```

```
08060              GO TO 78
08070       77     CVALUE(TAG,4)=SQRT((CVALUE(TAG,4)-CVALUE(TAG,3)**2/
08080         &                    CVALUE(TAG,1))/(CVALUE(TAG,1)-1))
08090              CVALUE(TAG,3)=CVALUE(TAG,3)/CVALUE(TAG,1)
08100       78     DO 80 J=1,NKEY
08110       80          IF(KEYVAL(J).EQ.0) GO TO 85
08120              J=J+1
08130       85     IF(J.EQ.1) J=2
08140              NUM=J-1
08150              IF(.NOT.TABLES) GO TO 87
08160              IF(NUM.NE.1) GO TO 86
08170              IF(WIDTH.LT.100) WRITE(6,623) TITLE
08180              IF(WIDTH.GE.100) WRITE(6,625) TITLE
08190       86     RFMT=FMTOUT(NUM,SIZE)
08200              CALL LABEL(KEY(NUM),KEYVAL(NUM),VLABEL)
08210              WRITE(6,RFMT) FNAME(KEY(NUM)),KEYVAL(NUM),VLABEL,
08220         &                          (CVALUE(TAG,K),K=1,4)
08230       87     IF(NUM.NE.NKEY) GO TO 90
08240              RECORD=BLANK
08250              ENCODE(RECORD,1130) (KEYVAL(K),K=1,4),(CVALUE(TAG,N),N=1,6)
08260              WRITE(15,1500) RECORD
08270       90     CONTINUE
08280          IF(.NOT.LIST) GO TO 100
08290          REWIND 15
08300          IF(FILEOUT.EQ.BLANK) GO TO 100
08310          WRITE(6,630) FILEOUT
08320          DO 95 I=1,NCELLS
08330              READ(15,1500) RECORD
08340       95     WRITE(6,635) RECORD
08350      100 WRITE(6,640) NR,FILEIN,NMISS,TITLE
08360          REWIND 15
08370          IF(BATCH) GO TO 110
08380          CALL DETACH(14,ISTAT,)
08390          IF(FILEOUT.EQ.BLANK) GO TO 110
08400          CALL DETACH(15,ISTAT,)
08410      110 RETURN
08420    C
08430    C-FORMATS-----------
08440      500 FORMAT(I1)
08450      600 FORMAT(1H1////' ***SUBROUTINE REPORT EXECUTING***',A80)
08460      603 FORMAT(1H ,'INPUT FILE MUST BE SPECIFIED')
08470      604 FORMAT(1H ,'UNABLE TO CREATE TEMPORARY OUTPUT FILE'/
08480         &           ' SPECIFY OUTPUT FILE NAME')
08490      605 FORMAT(1H ,'READY')
08500      610 FORMAT(1H ,'*****ILLEGAL FACTOR=',A6)
08510      615 FORMAT(1H ,'*****WARNING---NUMBER OF CELLS EXCEEDS 1000')
08520      620 FORMAT(1H ,'*****REPORT READY---POSITION PAPER')
08530      623 FORMAT(1H1//////1H ,70(1H-)/1X,A70/1H ,70(1H-)/
08540         & 32X,'TOTAL',6X,'UNSTANDARDIZED CPUE'/
08550         & 23X,'TOWS',5X,'CATCH',8X,'MEAN',8X,'S.D.'/1H ,70(1H-)//)
08560      625 FORMAT(1H1//////1H ,99(1H-)/10X,A80/1H ,99(1H-)/
08570         &       56X,'TOTAL',6X,'UNSTANDARDIZED CPUE'/47X,'TOWS',5X,
08580         &      'CATCH',8X,'MEAN',8X,'S.D.'/1H ,99(1H-)//)
08590      630 FORMAT(1H1//1H 'LISTING OF RECORDS WRITTEN ON FILE ',A80)
08600      635 FORMAT(1H ,A80)
08610      640 FORMAT(//////1H ,I5,' RECORDS WERE READ FROM FILE ',A80/
08620         &      1H ,I5,' RECORDS WERE UNCLASSIFIABLE'///
08630         &      1H ,'****SUBROUTINE REPORT TERMINATING****',A80)
```

```
08640     1100 FORMAT(13X,F3.0,E10.4,10X,4E10.4)
08650     1105 FORMAT(53X,F5.0,F6.1,F5.3)
08660     1110 FORMAT(A32,I4)
08670     1120 FORMAT(4I8,I4)
08680     1130 FORMAT(4I5,F5.0,5E10.4)
08690     1400 FORMAT(A80)
08700     1500 FORMAT(A80)
08710          END
08720          SUBROUTINE STANDARD
08730    C
08740    C-PURPOSE-----------STANDARDIZE AND REPORT CATCH AND EFFORT STATISTIC
08750    C
08760    C-GLOBAL VARIABLES--
08770          COMMON /IO/ TSS,BATCH,TITLE,WIDTH,SPACE(20000)
08780          LOGICAL TSS,BATCH
08790          CHARACTER TITLE*80
08800          INTEGER SPACE,WIDTH
08810    C     DEFINITIONS   SEE MAIN PROGRAM
08820    C
08830    C-LOCAL VARIABLES---
08840          CHARACTER *80 FILEIN,BLANK,VLABEL*17,FACTORS*6(3),
08850         &           FNAME*6(7)
08860          REAL  CELLS(25,25,2), SE(51,51), SUM,SUMX,SUMN,NUM,
08870         & ISE(51,51),EST(50),SUM2,YPY,VAR,BIAS
08880          INTEGER MODEL,F1,VF1,NF1,F2,VF2(25),NF2,F3,VF3(25,25),
08890         &          NF3(25),NR,NC,DUMMY,KEY(3),MNUM
08900          LOGICAL DONE,ANOVA
08910    C     DEFINITIONS
08920    C          FILEIN    INPUT FILE--OUTPUT FILE FROM SUBROUTINE REPORT
08930    C          VLABEL    VALUE LABEL
08940    C          ANOVA     .T. TO PRINT ANOVA TABLE
08950    C          FACTORS   USER SPECTIFIED KEY FACTORS
08960    C          FNAME     POSSIBLE KEY FACTORS
08970    C          CELLS     STANDARDIZATION MATRIX
08980    C          SE        SIMULTANEOUS EQUATIONS
08990    C          ISE       INVERSE OF SE
09000    C          SUM       LOG(CPUE) FROM INPUT FILE
09010    C          SUMX      SUM OF LOG(CPUE)
09020    C          SUMN      SUM OF TOWS
09030    C          NUM       NUMBER OF TOWS FROM INPUT FILE
09040    C          MODEL     MODEL NUMBER
09050    C          F1        FACTOR 1 FROM INPUT RECORD
09060    C          NF1       NUMBER OF LEVELS OF FACTOR 1
09070    C          VF1       VALUE OF FACTOR 1
09080    C          F2        FACTOR 2 FROM INPUT RECORD
09090    C          VF2       VALUE OF FACTOR 2
09100    C          NF2        NUMBER OF LEVELS OF FACTOR 2
09110    C          F3        FACTOR 3 FROM INPUT RECORD
09120    C          VF3       VALUE OF FACTOR 3
09130    C          NF3       NUMBER OF LEVELS OF FCATOR 3
09140    C          NR        NUMBER OF ROWS IN SE
09150    C          NC        NUMBER OF COLUMNS IN SE
09160    C          DUMMY     DUMMY VARIABLE
09170    C          DONE      TRUE TO INDICATE END OF INPUT FILE
09180    C          KEY       FACTOR CODE
09190    C          BLANK     CHARACTER CONSTANT
09200    C          MNUM      TOTAL NUMBER OF OBSERVATIONS IN MODEL
09210    C          SUM2      SUMS OF SQUARES OF LOG(CPUE)
```

```
09220   C          YPY        TOTAL SUMS OF SQUARES (Y PRINT Y MATRIX)
09230   C          VAR        VARIANCE
09240   C          BIAS       BIAS
09250   C          EST        ESTIMATES OF COEFFICIENTS
09260   C          VAR        VARIANCE OF ESTIMATES
09270   C
09280   C-EQUIVALENCES------
09290         EQUIVALENCE (SPACE(1),CELLS(1,1,1)),(SPACE(1251),SE(1,1)),
09300        &            (SPACE(3852),VF2(1)),(SPACE(3877),NF3(1)),
09310        &            (SPACE(3902),VF3(1,1)),(SPACE(4528),ISE(1,1))
09320   C
09330   C-INPUT-------------
09340         NAMELIST /IN/ FILEIN,FACTORS,MODEL,TITLE,ANOVA
09350   C
09360   C-CONSTANTS---------
09370         DATA FNAME(1)/6HAREA  /,FNAME(2)/6HSEASON/,FNAME(3)/6HGEAR  /,
09380        &     FNAME(4)/6HMONTH /,FNAME(5)/6HDAY   /,FNAME(6)/6HSAREA /,
09390        &     FNAME(7)/6HVESSEL/
09400         DATA BLANK /1H /
09410   C
09420   C-INITIALIZATION----
09430         FILEIN=' '
09440         MODEL=2
09450         DONE=.F.
09460         DO 5 I=1,25
09470             DO 5 J=1,25
09480                 DO 5 K=1,2
09490       5             CELLS(I,J,K)=0.0
09500         ANOVA=.T.
09510   C
09520   C-START-------------
09530         WRITE(6,600) TITLE
09540         IF(TSS) WRITE(6,605)
09550         READ(5,IN)
09560         GO TO (10,12), MODEL
09570      10 FACTORS(1)=FNAME(1)
09580         FACTORS(2)=FNAME(3)
09590         FACTORS(3)=FNAME(7)
09600         GO TO 13
09610      12 FACTORS(1)=FNAME(1)
09620         FACTORS(2)=FNAME(2)
09630         FACTORS(3)=FNAME(3)
09640      13 IF(BATCH) GO TO 15
09650         IF(FILEIN.EQ.BLANK) GO TO 15
09660         CALL GETFILE(FILEIN,15)
09670      15 DO 25 I=1,3
09680             DO 20 J=1,7
09690      20         IF(FACTORS(I).EQ.FNAME(J)) GO TO 25
09700             WRITE(6,610) FACTORS(I)
09710             IF(BATCH) GO TO 400
09720             GO TO 10
09730      25     KEY(I)=J
09740         IF(MODEL.EQ.1.OR.MODEL.EQ.2) GO TO 30
09750         WRITE(6,615) MODEL
09760         IF(BATCH) GO TO 400
09770         GO TO 10
09780      30 IF(BATCH) GO TO 35
09790         WRITE(6,616)
```

```
09800          READ(5,500) DUMMY
09810       35 GO TO (100,200), MODEL
09820    C
09830      100 CONTINUE
09840    C
09850    C-DESCRIPTION
09860    C                  THIS SECTION COMPUTES MODEL 1 STANDARDIZATION
09870    C
09880          VF1=0
09890          NF1=0
09900      110 READ(15,1500,END=115) F1,F2,F3,NUM,SUM,SUM2
09910          IF(F1.EQ.VF1) GO TO 173
09920          IF(NF1.EQ.0) GO TO 170
09930          GO TO 120
09940      115 DONE=.T.
09950      120 NR=NF2+1
09960          DO 125 I=1,NF2
09970      125     NR=NR+NF3(I)
09980          NC=NR+1
09990          DO 130 I=1,NR
10000              DO 130 J=1,NC
10010      130         SE(I,J)=0.0
10020          JJ=NF2+1
10030          DO 140 I=1,NF2
10040              SUMX=0.0
10050              SUMN=0.0
10060              DO 135 J=1,NF3(I)
10070                  SUMX=SUMX+CELLS(I,J,1)
10080      135         SUMN=SUMN+CELLS(I,J,2)
10090              SE(1,1)=SE(1,1)+SUMN
10100              SE(1,NC)=SE(1,NC)+SUMX
10110              II=I+1
10120              SE(II,NC)=SUMX
10130              SE(II,1)=SUMN
10140              SE(1,II)=SUMN
10150              SE(II,II)=SUMN
10160              DO 140 J=1,NF3(I)
10170                  JJ=JJ+1
10180                  SE(II,JJ)=CELLS(I,J,2)
10190                  SE(JJ,II)=CELLS(I,J,2)
10200                  SE(1,JJ)=CELLS(I,J,2)
10210                  SE(JJ,1)=CELLS(I,J,2)
10220                  SE(JJ,JJ)=CELLS(I,J,2)
10230      140         SE(JJ,NC)=CELLS(I,J,1)
10240          DO 145 I=1,NF2
10250              II=I+1
10260              DO 145 J=2,NR
10270      145         SE(J,II)=SE(J,II)+1
10280          DO 160 N=1,NF2
10290              I1=II+1
10300              I2=II+NF3(N)
10310              DO 155 I=I1,I2
10320                  DO 155 J=I1,I2
10330      155             SE(I,J)=SE(I,J)+1
10340      160     II=I2
10350          CALL SOLVE(SE,ISE,EST,NR)
10360          IF(WIDTH.LT.100) WRITE(6,617) TITLE
10370          IF(WIDTH.GE.100) WRITE(6,618) TITLE
```

```
10380          CALL ANOVAT(MNUM,NR,EST,SE,YPY,WIDTH,VAR,ANOVA)
10390          IF(WIDTH.LT.100) WRITE(6,619) MODEL
10400          IF(WIDTH.GE.100) WRITE(6,620) MODEL
10410          CALL LABEL(KEY(1),VF1,VLABEL)
10420          BIAS=.5*ISE(1,1)*VAR
10430          CPUE=EXP(EST(1)+BIAS)
10440          WRITE(6,625) FACTORS(1),VF1,VLABEL,EST(1),CPUE
10450          JJ=NF2+1
10460          II=1
10470          DO 165 I=1,NF2
10480               II=II+1
10490               BIAS=.5*(ISE(1,1)+ISE(II,II)+2*ISE(1,II))*VAR
10500               CPUE=EXP(EST(1)+EST(II)+BIAS)
10510               CALL LABEL(KEY(2),VF2(I),VLABEL)
10520               WRITE(6,630) FACTORS(2),VF2(I),VLABEL,EST(II),CPUE
10530               DO 165 J=1,NF3(I)
10540                    CELLS(I,J,1)=0.0
10550                    CELLS(I,J,2)=0.0
10560                    JJ=JJ+1
10570                    BIAS=.5*(ISE(1,1)+ISE(II,II)+ISE(JJ,JJ)+
10580         &          2*(ISE(1,II)+ISE(1,JJ)+ISE(II,JJ)))*VAR
10590                    CPUE=EXP(EST(1)+EST(II)+EST(JJ)+BIAS)
10600                    CALL LABEL(KEY(3),VF3(I,J),VLABEL)
10610       165          WRITE(6,635) FACTORS(3),VF3(I,J),VLABEL,
10620            &                        EST(JJ),CPUE
10630          IF(DONE) GO TO 400
10640       170 VF1=F1
10650          NF1=NF1+1
10660          NF2=1
10670          VF2(1)=F2
10680          NF3(1)=1
10690          VF3(1,1)=F3
10700          YPY=0.0
10710          MNUM=0
10720       173 IF(F2.EQ.VF2(NF2)) GO TO 175
10730          NF2=NF2+1
10740          VF2(NF2)=F2
10750          NF3(NF2)=1
10760          VF3(NF2,1)=F3
10770       175 IF(F3.EQ.VF3(NF2,NF3(NF2))) GO TO 180
10780          NF3(NF2)=NF3(NF2)+1
10790          VF3(NF2,NF3(NF2))=F3
10800       180 IF(NF2.LE.25.AND.NF3(NF2).LE.25) GO TO 185
10810          WRITE(6,640)
10820          GO TO 400
10830       185 CELLS(NF2,NF3(NF2),1)=CELLS(NF2,NF3(NF2),1)+SUM
10840          CELLS(NF2,NF3(NF2),2)=CELLS(NF2,NF3(NF2),2)+NUM
10850          YPY=YPY+SUM2
10860          MNUM=MNUM+NUM
10870          GO TO 110
10880     C
10890     C
10900       200 CONTINUE
10910     C
10920     C-DESCRIPTION--------
10930     C                THIS SECTION CCOMPUTE MODEL 2 STANDARAIZATION
10940     C
10950          VF1=0
```

```
10960          NF1=0
10970      210 READ(15,1500,END=215) F1,F2,F3,NUM,SUM,SUM2
10980          IF(F1.EQ.VF1) GO TO 267
10990          IF(NF1.EQ.0) GO TO 265
11000          GO TO 220
11010      215 DONE=.T.
11020      220 NR=NF2+NF3(1)+1
11030          NC=NR+1
11040          DO 225 I=1,NR
11050              DO 225 J=1,NC
11060      225         SE(I,J)=0.0
11070          DO 235 I=1,NF2
11080              SUMX=0.0
11090              SUMN=0.0
11100              DO 230 J=1,NF3(1)
11110                  SUMX=SUMX+CELLS(I,J,1)
11120      230         SUMN=SUMN+CELLS(I,J,2)
11130              SE(1,1)=SE(1,1)+SUMN
11140              SE(1,NC)=SE(1,NC)+SUMX
11150              II=I+1
11160              SE(II,NC)=SUMX
11170              SE(II,1)=SUMN
11180              SE(1,II)=SUMN
11190              SE(II,II)=SUMN
11200              DO 235 J=1,NF3(1)
11210                  JJ=NF2+1+J
11220                  SE(II,JJ)=CELLS(I,J,2)
11230      235         SE(JJ,II)=CELLS(I,J,2)
11240          DO 245 J=1,NF3(1)
11250              SUMX=0.0
11260              SUMN=0.0
11270              DO 240 I=1,NF2
11280                  SUMX=SUMX+CELLS(I,J,1)
11290      240         SUMN=SUMN+CELLS(I,J,2)
11300              JJ=NF2+1+J
11310              SE(JJ,NC)=SUMX
11320              SE(JJ,1)=SUMN
11330              SE(1,JJ)=SUMN
11340      245     SE(JJ,JJ)=SUMN
11350          II=NF2+1
11360          DO 250 I=2,II
11370              DO 250 J=2,II
11380      250         SE(I,J)=SE(I,J)+1
11390          JJ=II+NF3(1)
11400          II=II+1
11410          DO 255 I=II,JJ
11420              DO 255 J=2,JJ
11430      255         SE(I,J)=SE(I,J)+1
11440          CALL SOLVE(SE,ISE,EST,NR)
11450          IF(WIDTH.LT.100) WRITE(6,617) TITLE
11460          IF(WIDTH.GE.100) WRITE(6,618) TITLE
11470          CALL ANOVAT(MNUM,NR,EST,SE,YPY,WIDTH,VAR,ANOVA)
11480          IF(WIDTH.LT.100) WRITE(6,619) MODEL
11490          IF(WIDTH.GE.100) WRITE(6,620) MODEL
11500          CALL LABEL(KEY(1),VF1,VLABEL)
11510          BIAS=.5*ISE(1,1)*VAR
11520          CPUE=EXP(EST(1)+BIAS)
11530          WRITE(6,625) FACTORS(1),VF1,VLABEL,EST(1),CPUE
```

```
11540           II=1
11550           DO 260 I=1,NF2
11560               II=II+1
11570               BIAS=.5*(ISE(1,1)+ISE(II,II)+2*ISE(1,II))*VAR
11580               CPUE=EXP(EST(1)+EST(II)+BIAS)
11590               CALL LABEL(KEY(2),VF2(I),VLABEL)
11600               WRITE(6,630) FACTORS(2),VF2(I),VLABEL,EST(II),CPUE
11610               JJ=NF2+1
11620               DO 260 J=1,NF3(1)
11630                   CELLS(I,J,1)=0.0
11640                   CELLS(I,J,2)=0.0
11650                   JJ=JJ+1
11660                   BIAS=.5*(ISE(1,1)+ISE(II,II)+ISE(JJ,JJ)+
11670          &        2*(ISE(1,II)+ISE(1,JJ)+ISE(II,JJ)))*VAR
11680                   CPUE=EXP(EST(1)+EST(II)+EST(JJ)+BIAS)
11690                   CALL LABEL(KEY(3),VF3(1,J),VLABEL)
11700       260         WRITE(6,635) FACTORS(3),VF3(1,J),VLABEL,
11710          &                     EST(JJ),CPUE
11720           IF(DONE) GO TO 400
11730       265 VF1=F1
11740           NF1=NF1+1
11750           NF2=1
11760           VF2(1)=F2
11770           NF3(1)=1
11780           VF3(1,1)=F3
11790           YPY=0.0
11800           MNUM=0
11810       267 IF(F2.EQ.VF2(NF2)) GO TO 270
11820           NF2=NF2+1
11830           VF2(NF2)=F2
11840       270 DO 280 I=1,NF3(1)
11850       280     IF(F3.EQ.VF3(1,I)) GO TO 285
11860           I=I+1
11870           NF3(1)=NF3(1)+1
11880           VF3(1,NF3(1))=F3
11890       285 IF(NF2.LE.25.AND.NF3(1).LE.25) GO TO 290
11900           WRITE(6,640)
11910           GO TO 400
11920       290 CELLS(NF2,I,1)=CELLS(NF2,I,1)+SUM
11930           CELLS(NF2,I,2)=CELLS(NF2,I,2)+NUM
11940           YPY=YPY+SUM2
11950           MNUM=MNUM+NUM
11960           GO TO 210
11970       C
11980       C
11990         400 WRITE(6,645) TITLE
12000             IF(BATCH) GO TO 410
12010             IF(FILEIN.EQ.BLANK) GO TO 410
12020             CALL DETACH(15,ISTAT,)
12030         410 RETURN
12040       C
12050       C-FORMATS-----------
12060         500 FORMAT(I1)
12070         600 FORMAT(1H1////' ***SUBROUTINE STANDARD EXECUTING***',A80)
12080         605 FORMAT(1H ,'READY')
12090         610 FORMAT(1H ,'*****ILLEGAL FACTOR = ',A6)
12100         615 FORMAT(1H ,'*****ILLEGAL MODEL NUMBER = ',I1)
12110         616 FORMAT(1H ,'*****REPORT READY---POSITION PAPER')
```

```
12120      617 FORMAT(1H1//////1H ,70(1H-)/1X,A70/1H ,70(1H-))
12130      618 FORMAT(1H1//////1H ,100(1H-)/1X,A80/1H ,100(1H-))
12140      619 FORMAT(47X,'MODEL ',I1,' STANDARDIZATION'/
12150         &        47X,'COEFFICIENT',8X,'CPUE'/1H ,70(1H-)//)
12160      620 FORMAT(47X,'MODEL ',I1,' STANDARDIZATION'/
12170         &        47X,'COEFFICIENT',8X,'CPUE'/1H ,99(1H-)//)
12180      625 FORMAT(1X,A6,'=',I5,A17,16X,F8.4,8X,F7.1)
12190      630 FORMAT(//6X,A6,'=',I5,A17,11X,F8.4,8X,F7.1)
12200      635 FORMAT(/11X,A6,'=',I5,A17,6X,F8.4,8X,F7.1)
12210      640 FORMAT(1H ,'*****NUMBER OF LEVELS EXCEEDS 25*****')
12220      645 FORMAT(//////1H '***SUBROUTINE STANDARD TERMINATING***',A80)
12230     1500 FORMAT(3I5,5X,F5.0,30X,2E10.4)
12240          END
12250          SUBROUTINE INVERT(MAT,IMAT,RANK)
12260    C
12270    C-PURPOSE-----------INVERT A MATRIX
12280    C
12290    C-ARGUMENTS----------
12300          REAL       MAT(51,51),IMAT(51,51)
12310          INTEGER    RANK
12320    C     DEFINITIONS
12330    C          MAT        MATRIX TO BE INVERTED
12340    C          IMAT       INVERSE OF MAT
12350    C          RANK       RANK OF MAT
12360    C
12370    C-LOCAL VARIABLES---
12380          REAL       S,WMAT(51,51)
12390          INTEGER    NR,NC
12400    C     DEFINITIONS
12410    C          S          DIAGONAL ELEMENT
12420    C          WMAT       WORKING MATRIX
12430    C          NR,NC      ROW,COLUMN INDEXES
12440    C
12450    C-INITIALIZATION---
12460    C
12470          NR=RANK
12480          NC=RANK+1
12490    C
12500    C-START-------------
12510    C
12520          DO 500 IROW=1,RANK
12530               DO 50 I=1,RANK
12540                    WMAT(I,NC)=0.0
12550                    DO 50 J=1,RANK
12560       50                WMAT(I,J)=MAT(I,J)
12570               WMAT(IROW,NC)=1.0
12580               DO 200 I=1,NR
12590                    S=WMAT(I,I)
12600                    DO 100 J=1,NC
12610       100               WMAT(I,J)=WMAT(I,J)/S
12620                    K=I
12630       125          IF(K.EQ.NR) GO TO 200
12640                         K=K+1
12650                         S=WMAT(K,I)
12660                         DO 150 J=1,NC
12670       150                    WMAT(K,J)=WMAT(K,J)-S*WMAT(I,J)
12680                         GO TO 125
12690       200          CONTINUE
```

```
12700              II=NR-1
12710              DO 300 I=1,II
12720                   JJ=NR-I+1
12730                   KK=NR-I
12740                   DO 250 K=1,KK
12750                        S=WMAT(K,JJ)
12760                        DO 250 J=1,NC
12770        250                  WMAT(K,J)=WMAT(K,J)-S*WMAT(JJ,J)
12780        300           CONTINUE
12790              DO 400 I=1,NR
12800        400           IMAT(IROW,I)=WMAT(I,NC)
12810        500      CONTINUE
12820          RETURN
12830          END
12840          SUBROUTINE SOLVE(XPX,IXPX,EST,RANK)
12850     C
12860     C-PURPOSE-----------MATRIX SOLUTION OF MULTIPLE REGRESSION,
12870     C                    NO INTERCEPT MODEL
12880     C
12890     C-ARGUMENTS---------
12900     C
12910          REAL       XPX(51,51),IXPX(51,51),EST(50),VAR(50)
12920          INTEGER    RANK
12930     C      DEFINITIONS
12940     C         XPX         X PRIME X MATRIX
12950     C         IXPX        INVERT OF X PRIME X MATRIX
12960     C         EST         ESTIMATES OF COEFFICIENTS
12970     C         RANK        RANK OF XPX
12980     C
12990     C-LOCAL VARIABLES---
13000          INTEGER    NC
13010     C      DEFINITIONS
13020     C         NC          RANK+1 COLUMN OF XPX IS X PRIME Y VECTOR
13030     C
13040     C-INITIALIZATION---
13050     C
13060          NC=RANK+1
13070     C
13080     C-START-------------
13090     C
13100          CALL INVERT(XPX,IXPX,RANK)
13110          DO 10 I=1,RANK
13120               EST(I)=0.0
13130               DO 10 J=1,RANK
13140                    EST(I)=EST(I)+IXPX(I,J)*XPX(J,NC)
13150       10           CONTINUE
13160          RETURN
13170          END
13180           SUBROUTINE ANOVAT(NOBS,NEST,BP,XPY,YPY,WIDTH,VAR,ANOVA)
13190     C
13200     C-PURPOSE-----------PRINT ANOVA TABLE FOR MODEL
13210     C
13220     C-ARGUMENTS---------
13230          REAL       BP(50),XPY(51,51),YPY,VAR
13240          INTEGER    NOBS,NEST,WIDTH
13250          LOGICAL    ANOVA
13260     C       DEFINITIONS
13270     C          NOBS        TOTAL NUMBER OF OBSERVATIONS
```

```
13280   C          NEST       NUMBER OF ESTIMATES
13290   C          BP         B PRIME MATRIX
13300   C          XPY        X PRIME Y MATRIX
13310   C          YPY        Y PRIME Y MATRIX
13320   C          WIDTH      NUMBER OF CHARACTERS PER OUTPUT LINE
13330   C          VAR        VARIANCE
13340   C          ANOVA      .T. TO PRINT ANOVA TABLE
13350   C
13360   C-LOCAL VARIABLES---
13370   C
13380         INTEGER    DFE
13390         REAL       SSR,MSR,SSE,MSE,SST,F,R2
13400   C     DEFINITIONS
13410   C          DFE        DEGREES OF FREEDOM-ERROR
13420   C          SSR        SUMS OF SQUARES-MODEL
13430   C          MSR        MEAN SQUARE-MODEL
13440   C          SSE        SUMS OF SQUARES-ERROR
13450   C          MSE        MEAN SQUARE-ERROR
13460   C          SST        SUMS OF SQUARES-TOTAL
13470   C          F          F-STATISTIC
13480   C          R2         MULTIPLE REGRESSION COEFFICIENT
13490   C
13500   C-INITIALIZATION---
13510   C
13520         SST=YPY
13530         SSR=0.0
13540         SSE=0.0
13550         DFE=NOBS-NEST
13560   C
13570   C-START-------------
13580   C
13590         J=NEST+1
13600         DO 10 I=1,NEST
13610      10     SSR=SSR+BP(I)*XPY(I,J)
13620         SSE=SST-SSR
13630         R2=SSR/SST
13640         MSR=SSR/NEST
13650         MSE=SSE/DFE
13660         F=MSR/MSE
13670         VAR=MSE
13680         IF(.NOT.ANOVA) RETURN
13690         IF(WIDTH.GE.100) GO TO 20
13700         WRITE(6,600) NEST,SSR,MSR,F,DFE,SSE,MSE,NOBS,SST,R2
13710         RETURN
13720      20 WRITE(6,605) NEST,SSR,MSR,F,DFE,SSE,MSE,NOBS,SST,R2
13730         RETURN
13740   C
13750   C-FORMATS-----------
13760     600 FORMAT(//1X,70(1H-)/16X,'ANALYSIS OF VARIANCE FOR ',
13770        &   'FITTING MODEL'/1X,70(1H-)/1X'SOURCE',10X,'D.F.',
13780        &   5X,'SUMS OF SQUARES',3X,'MEAN SQUARE',4X,
13790        &   'F-STATISTIC'/1X,70(1H-)//1X,'MODEL',
13800        &   10X,I5,7X,F10.2,6X,F10.2,5X,F10.4/1X,'ERROR',10X,
13810        &   I5,7X,F10.2,6X,F10.2/1X,'TOTAL',10X,I5,7X,F10.2//
13820        &   1X,70(1H-)/1X,'MULTIPLE CORRELATION COEFFICIENT',3X,
13830        &   F6.3/1X,70(1H-))
13840     605 FORMAT(//1X,100(1H-)/16X,'ANALYSIS OF VARIANCE FOR ',
13850        &   'FITTING MODEL'/1X,100(1H-)/1X'SOURCE',10X,'D.F.',
```

```
13860          &    5X,'SUMS OF SQUARES',3X,'MEAN SQUARE',4X,
13870          &    'F-STATISTIC'/1X,100(1H-)//1X,'MODEL',
13880          &    10X,I5,7X,F10.2,6X,F10.2,5X,F10.4/1X,'ERROR',10X,
13890          &    I5,7X,F10.2,6X,F10.2/1X,'TOTAL',10X,I5,7X,F10.2//
13900          &    1X,100(1H-)/1X,'MULTIPLE CORRELATION COEFFICIENT',3X,
13910          &    F6.3/1X,100(1H-))
13920           END

*
```