# An adaptation of "APL programs for stock assessments" for an 8088-based microcomputer

by

D. Rivard and M. Joly

Resource Research Branch
Department of Fisheries and Oceans
240 Sparks Street, 6th Floor West
Ottawa, Ontario
K1A 0E6

## ABSTRACT

A series of standard APL functions used for fish stock assessments have been adapted to run on a microcomputer, namely a 8088-based system. The functions have been modified to take advantage of the computing environment provided by the microcomputer system. Most operations can be accessed via a main menu and the output can easily be routed to any output device such as a disk drive, a printer, the screen display or a terminal.

## RÉSUMÉ

Les fonctions APL utilisées pour l'évaluation des stocks de poissons ont été modifiées pour fonctionner sur un micro-ordinateur, soit un système basé sur le microprocesseur 8088. Les modifications apportées mettent à profit le milieu créé par ce système informatique. La plupart des opérations est maintenant accessible au moyen d'un menu principal et les résultats peuvent être facilement dirigés vers tout périphérique disponible, comme un lecteur de disques, une imprimante, un écran cathodique ou un terminal.

## TABLE OF CONTENTS

The functions described in "APL programs for stock assessments" (Rivard, 1982) are adapted here for a microcomputer environment, namely a 8088-based microcomputer with screen graphics, serial and parallel interfaces. A number of new functions have been included and others have been rewritten to minimize memory requirements, optimize the use of floppy disks and to facilitate the routing of output to various devices (such as disks, printers, screen, terminals). These functions are described below. This description assumes that the user is already familiar with APL and with the disk operating system.

## System configuration and software requirements

The following hardware configuration was used for this implementation :

- -IBM-PC™ with 512K RAM, two floppy disk drives, color/graphics adapter, serial interface (RS-232) and parallel interface;
- -dot matrix printer;
- -APL terminal (optional);
- -8087 coprocessor (optional).

Technical notes on RS-232-C cable wiring for various output devices are provided in Appendix C.

This implementation uses WATCOM APL version 2.0 (WATCOM Publications, P.O Box 880, Waterloo, Ontario N2J 4C3), under PC-DOS 2.0 . In this version of WATCOM APL, workspace size is only limited by the size of available RAM. The calculations are performed with an accuracy of 15 digits. Speed of execution is adequate for most tasks but can be improved by using an arithmethic coprocessor, i.e. the 8087.

## Installation

First, make a working copy of your original diskette in the following manner. Put the DOS 2.0 diskette in drive A; turn on the computer and wait for the DOS prompt (A>). Then put a blank diskette in drive B: and type

```
A>FORMAT B:/s                    return
```

The diskette in drive B: will be formatted and the DOS system command file will be copied on your diskette. Then put the distribution diskette in drive A and type

```
A>copy A*.* B:                   return
```

The distribution diskette will then be copied on drive B: . Remove the distribution diskette from drive A and put the WATCOM APL 2.0 diskette in drive A. Then load WATCOM APL 2.0 as indicated in the user's guide. Remove the WATCOM APL 2.0 diskette from drive A , put the working copy of FISH in drive A and type

```
)LOAD FISH                       return
```

When the cursor is back on the screen, press the function key **F3** to see a list of APL functions and variables that are now defined in your active workspace.

It is advised to make a back-up of your working copy of the distribution diskette. Finally, Appendix D contains suggestions on how to set up hard disk files if you have access to a hard disk system.

## The files

This package includes a number of workspaces which are stored on the distribution diskette in the following files:

| FILE NAME | CONTENT |
|---|---|
| FISH | main workspace (must be loaded first); |
| FILE | utility functions to input/output from/to file; |
| BEVHOLT | Beverton and Holt yield-per-recruit analysis; |
| PALOHEIM | estimation of total mortality rates (Z); |
| PRODMDL | surplus production models; |
| PROJECT | projection programs: age-structured projections; |
| PROJVAR | variances for catch projections; |
| SPA | cohort analysis programs; |
| SPASA | cohort analysis and sensitivity analysis; |
| SURVIVOR | estimation of survivors for the current year; |
| YPR | Thompson and Bell yield per recruit; |
| PLOT | XY-graphs for screen; |
| REGR | linear regression analysis; |
| COMM | functions for using the PC as a terminal. |

FISH, the main workspace, must always be loaded first as follows:

)LOAD FISH return

In order to use a function defined in any of the files, you must first activate it by using the command )COPY. For example,

)COPY *name* return

will activate all variables and functions stored in the file identified by *name*. The variable *name*∆DIR contains the names of all functions which have been activated by )COPY. You can erase from your active workspace all functions that have been activated by a )COPY statement by using the function ∆EXP:

∆EXP *name*∆DIR return

## Workspace FISH

FISH contains a series of functions which are needed by a number of APL functions contained in this package. These functions, which are useful for preparing matrices and formatting output, are described in detail in Rivard (1982). Thus their description is not repeated here and only the following list is provided:

| | |
|---|---|
| A A∆PLUS B | sums the values contained in B for the range of age-groups specified in the vector A; |
| A BECOMES B | corrects erroneous entries of the matrix MAT; |
| DAT | prints the current date in a standard format; |
| HEADER B | prints the character vector B and underlines it; |
| INPUT∆MAT | used to create a matrix of data (result is stored in matrix MAT); |
| A OUT B | prints with A decimal places the matrix B; |
| A OUT∆F B | prints a matrix B of fishing mortalities with A decimal places; |
| RESET∆INDICES NAME | resets the indices (year, age) of a matrix NAME (must be used before using OUT); |
| A R∆CAT B | catenates the character vector B along the rows of matrix A; |
| A TABLE B | tabulates a matrix B with A decimal places; |
| UNDER B | prints and underlines all non-blank characters of vector B. |

The following new functions have been added to facilitate workspace management:

| | |
|---|---|
| MENU | this function lists a menu outlining the various options which are available and prompts the user to input one or more numbers corresponding to the selections which will be automatically copied in the active workspace. After typing GO return, the "main" function of the last option selected will be invoked. |

LIST *namelist* — utility function to list programs and variables. The argument *namelist* can be a vector of function or variable name(s), enclosed in quotes (e.g. LIST 'A B FONC'), or the name of a variable containing such a list (e.g. LIST FISH∆DIR).

∆EXP *namelist* — utility function to erase specific functions or variables from the active workspace. The argument of ∆EXP is similar to that of LIST above (e.g. ∆EXP 'F1 F2' or ∆EXP PLOT∆DIR). ∆EXP outputs a vector of 0's and 1's, where each 1 means that the corresponding name in the input list has been erased. Note that you can erase all variables from the workspace by using ⎕EX ⎕NL 2 .

When the function MENU is activated, a series of prompts are issued to guide the user. The menu options will appear as follows:

```
WORKSPACE FISH: VERSION XX/XX/XX
OPTIONS:   1. VIRTUAL POPULATION ANALYSIS: CALCULATIONS
           2.                              SENSITIVITY ANALYSIS
           3. YIELD PER RECRUIT ANLAYSIS : THOMPSON AND BELL
           4.                              BEVERTON AND HOLT
           5. CATCH PROJECTIONS:           CALCULATIONS
           6.                              VARIANCES
           7. SURVIVOR ANALYSIS
           8. PALOHEIMO METHOD
           9. PRODUCTION MODELS:    SCHAEFER
          10.                       PELLA-TOMLINSON
          11. GRAPHICS FUNCTIONS:   SCREEN
          12.                       H.P. (NOT IMPLEMENTED YET)
          13. LINEAR REGRESSION
          14. COMMUNICATION (TERMINAL MODE)
          15. CREATE MATRIX OF DATA
          16. EXIT: TO ACTIVE WORKSPACE
          17.       TO CLEAR WORKSPACE
          18.       TO DISK OPERATING SYSTEM (I.E. LEAVE APL)

INPUT OPTION NUMBER(S):
  ⎕: 11 1 16                    (Example of user's response)
```

If multiple option numbers are entered, options 15 and 16 must appear last. Options 17 and 18 will destroy the content of your active workspace and will thus cancel the effect of previous selections.

The utility functions and variables used by the MENU function are listed in the variable MENU∆DIR. Similarly, the remaining functions and variables related to workspace FISH are listed in the variable FISH∆DIR. You can erase these functions from your active workspace by typing

```
∆EXP FISH∆DIR                  return
∆EXP MENU∆DIR                  return
```

**Routing the output**

Output can be routed to any output device by activating this device before entering an APL function. For example,

---

| | |
|---|---|
| CRT | will send output to screen; |
| PRT | will send output to the printer; |
| RS232 | will send output to an APL terminal attached to the serial port; |
| TOFILE | will divert output to any disk file (this function is stored in worskpace FILE). |

---

Then the function QPUT can be used to send output to the active device. For example,

```
PRT                                    return
QPUT 'HELLO'                           return
```

will print HELLO on the printer, and (if MAT is a matrix of data)

```
QPUT MAT                               return
```

will print the content of the matrix MAT on the printer. Output is always sent to the device which has been activated last. Note also that a remote host computer (see Appendix B) can be considered as an output device. The user should note that the screen display is activated as output device when FISH is first loaded.

The function TOFILE can be used to divert output to a disk file. TOFILE will ask for a filename. If the filename specified does not correspond to an existing file, then a new file is created. If the filename corresponds to an existing file, then its content may be deleted or new output may be appended to it. The user can now proceed with the desired action (e.g. cohort analysis, virtual population analysis, etc.). Output will be diverted to the file specified. In order to terminate the routing of output to file, you **must activate another device** by using CRT, PRT or RS232: this will force the system to update the file directory on your diskette. If you do not give control to another device, your output may be lost. The function READFILE can be used to read a file which has been created with TOFILE. READFILE will output the file content on the active device.

## Utility functions.

### Functions for graphical output (workspace PLOT).

This workspace contains APL-functions for producing graphical output on the CRT. Graphs can be stored on a disk file and can be recalled from disk if needed. The main function, XYPLOT, must be called as follows:

```
XMAT XYPLOT YMAT                    return
```

where XMAT and YMAT are data matrices. Up to three data series may be plotted in a single graph. The variables XMAT or YMAT must have dimensions (*nser*, *nobs*), or *nobs* where

*nser* is the number of data series to be included; can be omitted if there is only one data series;
*nobs* is the number of observations per series.

Consequently, XMAT and YMAT are matrices, for which each row of YMAT is plotted against the corresponding row of XMAT, or vectors. The function XYPLOT will prompt you to specify the minimum and the maximum for each axis, as well as the number of "tic" marks to draw on each axis. In its default mode, the function identifies each observation with a symbol and links each point of a given series with a line having a specific pattern.

The following global variables must be used to specify a general title, labels for each axis or a legend:

| Variable | Description |
|---|---|
| GENTIT | general title for the graph; |
| XTIT | label for the x axis; |
| YTIT | label for the y axis; |
| LEGEND | legend :dimension (*nser*, *length of string*). |

In addition, the global variables OBI and LIN may be used to specify how each data series will be identified. OBI controls the identification of observed points by symbols; LIN controls the presence/absence of a line linking the points for a given series. OBI and LIN have dimension 3 and each entry can take the value 'Y', for indicating the presence, or 'N', for indicating the absence. The following table summarizes the outcome of each option:

| OBI | LIN: N | LIN: Y |
|---|---|---|
| N | | ------- |
| Y | ∆ | ---∆--- |

Finally, the global variables FL∆SCALX and FL∆SCALY can be used to control the calculation of the lower and upper limits of the axes. FL∆SCALX←'MANU' and FL∆SCALY←'MANU' will force the function XYPLOT to prompt you for input but FL∆SCALX←'AUTO' and FL∆SCALY←'AUTO' will force the function to calculate the lower and upper limits of each axis from input data.

The result of XYPLOT appears on the screen but is also stored on a disk file (the global variable PIXFILE contains the *filename* which will be used but if PIXFILE is not defined, then the graph is stored on file BSAVED.PIX). It is good practice to assign the extension .PIX to any file containing a screen image. You can merge at any time a screen image that has been stored on a disk file by using BLOAD: for exemple,

BLOAD 'BSAVED.PIX' return

or, in general

BLOAD 'drive:filename.extension' return

If GRAPHICS.COM has been loaded prior to loading APL, you can print any graph appearing on the screen by using the PrtSc key. A sample of graphical output on the dot matrix printer is provided in Appendix A.

The following table summarizes the options available:

| Global variable | Default value (option) | Description |
|---|---|---|
| GENTIT | *blank* | general title for the graph |
| XTIT | 'X AXIS' | label for the X-axis |
| YTIT | 'Y AXIS' | label for the Y-axis |
| LEGEND | *blank* | legend |
| OBI | 'YYY' ('NNN') | controls the identification of observations by symbols |
| LIN | 'YYY' ('NNN') | controls presence/absence of a line linking the observations |
| PIXFILE | 'BBSAVED.PIX' | name of file for storing screen image |
| FL∆SCALX | 'MANU' ('AUTO') | controls calculation of lower and upper limits for X axis |
| FL∆SCALY | 'MANU' ('AUTO') | controls calculation of lower and upper limits for Y axis |

**Functions for statistical analysis (workspace REGR).**

This workspace contains the APL-functions for estimating the coefficients of linear regression models. The main function, REGR, must be called as follows:

```
Y REGR X
```

where Y and X are matrices of data. The dimensions of Y are (*nobs*,1) and the dimensions of X are (*nobs*, *nind*), where

*nind* is the number of independent variables to be included (+1 if the origin is to be estimated);
*nobs* is the number of observations per variable.

For the origin of the regression line to be estimated, the first column of X must be a column of ones. The result of REGR may be presented in a tabular form by using

```
PRINT∆REGR
```

In addition, the results are available in the following global variables:

| | |
|---|---|
| B | *estimated regression coefficients;* |
| VAR∆B | *variance-covariance matrix of regression coefficients;* |
| CORR | *multiple correlation coefficient;* |
| DF | *number of observations.* |

## Stock assessment functions.

These functions are described in Rivard (1982). Only a brief description of the main functions appears below. For producing graphical output, you will need to merge the functions of PLOT in your active workspace. If the functions of PLOT are not defined in your active workspace, the function will abort and you will be prompted to copy workspace PLOT in your active workspace. If workspace size is limited, it is recommended to store your workspace and to copy in a "CLEAR" workspace only the functions and variables which are needed for producing the graph.

### YPR

This workspace contains the functions which are needed to perform a yield-per-recruit analysis by the method of Thompson and Bell.

| | |
|---|---|
| YIELD | asks for input, calculates and outputs results; |
| YIELD∆CONTROL | displays the current and default values of the control variables used by YIELD; |
| PLOT∆YIELD | prepares graphical output (needs the functions of PLOT). |

### BEVHOLT

This workspace contains the functions which are needed to perform a yield-per-recruit analysis by the method of Beverton and Holt.

| | |
|---|---|
| BEVHOLT | asks for input, calculates and outputs results; performs a sensitivity analysis (optional). |

### PROJECT

This workspace contains the functions which are needed to perform catch projections.

| | |
|---|---|
| MPROJECT | asks for input data and calls the different subroutines necessary to perform the catch projections and to output the results; |
| MPROJECT∆CONTROL | displays the current and the default values of the control variables used by the function MPROJECT. |

**PROJVAR**

This workspace contains the functions which are needed to calculate the variance of catch projections. Variance estimates are approximated by the first terms of a Taylor expansion from the variance of input parameters (Rivard, 1982).

| | |
|---|---|
| PROJECT∆VAR | asks for input data and calls the different subroutines necessary to perform the catch projections, to calculate variances and to output the results. |

**SPA**

This workspace contains the functions which are needed to perform sequential population analysis: cohort analysis or virtual population analysis. Before using these functions, data matrices must be defined in your active workspace; you can prepare input data by using the utility functions of FISH (e.g. INPUT∆MAT).

| | |
|---|---|
| INPUT∆COHORT | prepares the input for either COHORT or VPA; |
| COHORT | calls the subroutines that calculate population numbers at age and the instantaneous rate of fishing mortality at age by using the method of cohort analysis described in Pope (1972); |
| VPA | calls the subroutines that calculate population numbers at age and the instantaneous rate of fishing mortality at age by using a virtual population analysis (Gulland, 1965); |
| AUTOF | computes the starting F for the oldest age-groups in the sequential population analysis; |
| GRAPH∆PRODUCTION | must be used after a call to the function COHORT or VPA in order to generate a graph showing the variations of the components of production through time; the graph is stored in the file BPROD.PIX; (needs the functions of PLOT); |
| GRAPH∆NET∆PROD | must be used after a call to the function COHORT or VPA in order to generate a graph showing the fluctuations of annual net production through time; the graph is stored in the file BPRODYLD.PIX; (needs the functions of PLOT). |

After the completion of INPUT∆COHORT, your catch matrix is defined in the global variable CATCH. If you need more space for executing COHORT or VPA, erase INPUT∆COHORT and the initial catch matrix before executing them.

**SPASA**

Same as SPA but allows also the calculation of sensitivity coefficients for certain calculated quantities (Rivard, 1982).

**PRODMDL**

This workspace contains the functions which are needed to estimate the coefficients of the Schaefer model or the Pella-Tomlinson model by various methods.

| | |
|---|---|
| INPUT∆PROD | used to input data and to initialize the variables CATCH, EFFORT, CPUE and MEAN∆YR which are necessary to execute the functions SCHAEFER and PROD∆FIT;. |
| SCHAEFER | estimates the parameters of the Graham-Schaefer model by Gulland's approximation; |
| PROD∆FIT | estimates the parameters of the Pella-Tomlinson model by Gulland's approximation; |
| GRAPH∆PROD∆MODEL | produces a graph of catch against effort by using the variables CATCH, EFFORT, C∆CAL and Y∆CAL previously defined by SCHAEFER or PROD∆FIT; the graph is stored in the file BSURPROD.PIX ; needs the functions of PLOT. |

**SURVIVOR**

This workspace contains the functions which are needed to calculate survivors in the current year by the method of Doubleday (1981).

| | |
|---|---|
| SURVIVOR | inputs data and calls the subroutines necessary to perform calculations and to output the results. |

If workspace size is limited, answer "NO" to the prompt "ANALYSIS OF VARIANCE TO BE PERFORMED (YES OR NO)?"

**PALOHEIM**

This workspace contains the functions which are needed to calculate the instantaneous rate of total mortality (Z) by using the method of Paloheimo. A new function, GRAPH∆PAL, is provided for producing graphical output.

| | |
|---|---|
| PALOHEIMO | asks for input, performs the calculations and outputs the results (requires the functions of REGR); |
| GRAPH∆PAL | produces graphical output (requires the functions of PLOT). |

**References**

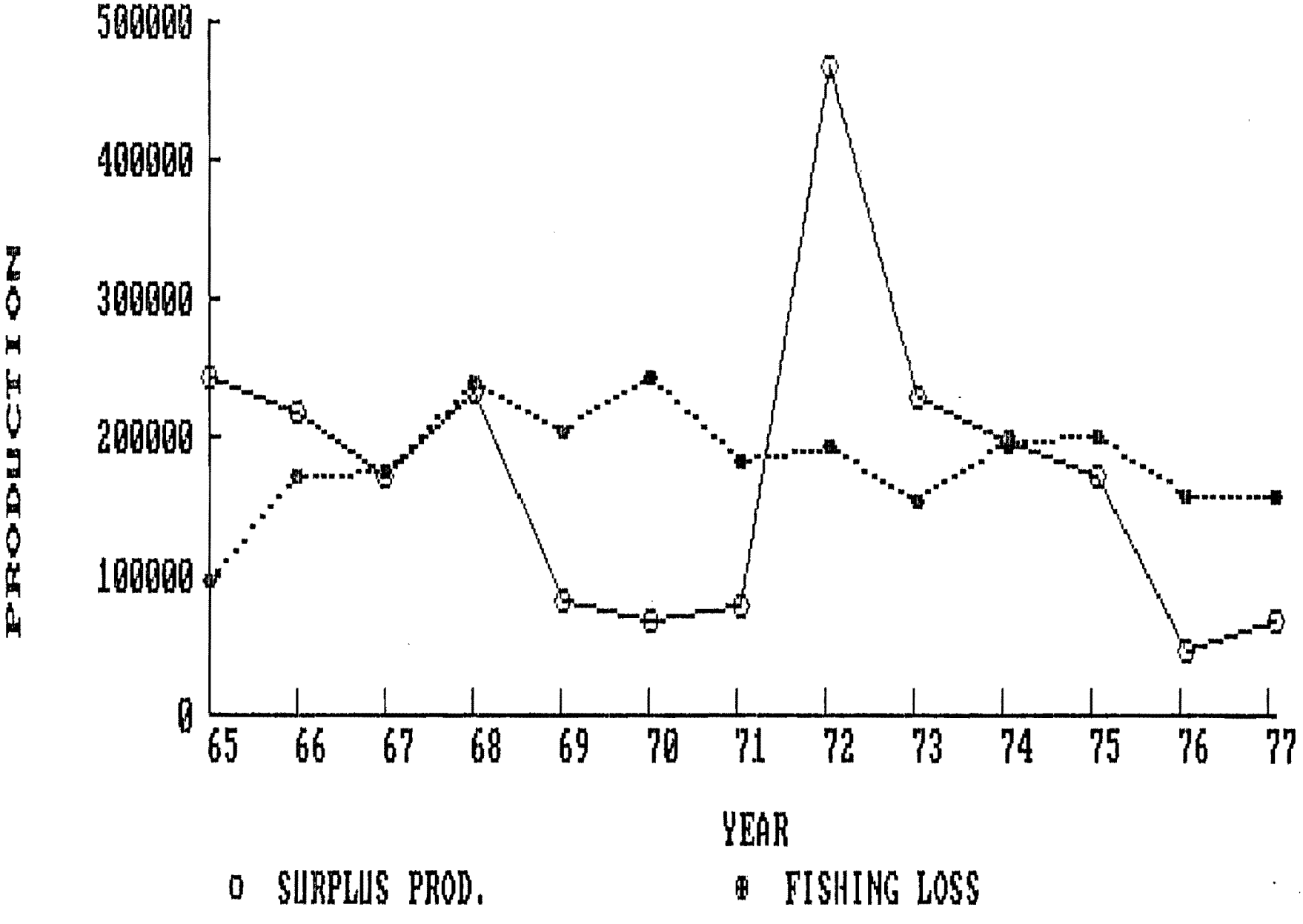
Doubleday, W.G. 1981. *A method of estimating the abundance of survivors of an exploited fish population using commercial fishing catch at age and research vessel abundance indices.* In Doubleday, W.G. and D. Rivard (Ed.). Bottom trawl surveys. Can. Spec. Publ. Fish. Aquat. Sci. 58: 273 p.

Gulland, J.A. 1965. *Estimation of mortality rates.* Annex to Arctic Fisheries Working Group Report (meeting in Hamburg, January 1965). ICES, C.M. 1965, Doc. No. 3 (mimeographed)

Pope, J.G. 1972. *An investigation of the accuracy of virtual population analysis using cohort analysis.* Int. Comm. Northwest Atl. Fish. Res. Bull. 9: 65-74.

Rivard, D. 1982. *APL Programs for stock assessment (revised).* Can. Tech. Rep. Fish. Aquat. Sci. 1091: 146 p.

Appendix A. Graphical output: an example.



## APPENDIX B

## APL functions for using the PC as a terminal

The workspace COMM contains APL-functions to use the IBM-PC as a computer terminal. In order to use the functions of COMM, simply )LOAD or )COPY COMM into your workspace.

### Using the IBM.PC as a terminal.

For using the IBM-PC as a primitive terminal, simply type in

SHARP∆LINK return

The system will respond with a summary of the logon procedure for accessing the host (I.P. Sharp Ltd is assumed). You must dial the local telephone number of the host-system, connect the receiver to the acoustic coupler, and pursue with the logon procedure. SHARP∆LINK leaves you in terminal mode, connected to the host system. At any time,

ALT/F10 will return command to the active workspace on the IBM-PC;
ALT/F2 will return command to the active workspace on the host system.

You can go back and forth between the two systems, as needed. ALT/F1 can be used to "toggle" between the APL and ASCII character sets, while in terminal mode (i.e. after entering ALT/F2). This can be used to perform the logon procedure in ASCII characters and revert to APL characters once the APL system has been activated on the host.

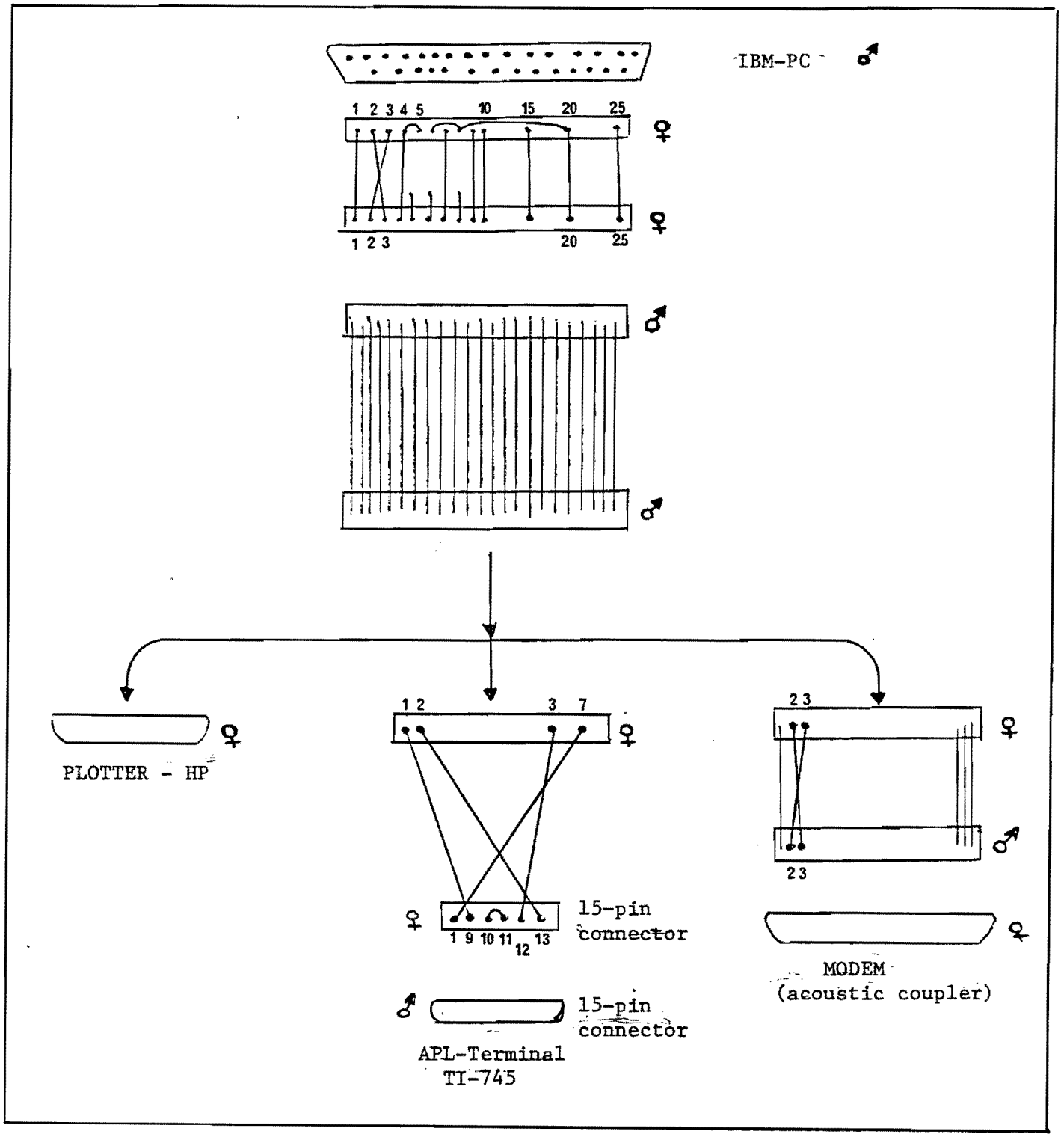
The workspace COMM was developed for linking with the I.P. Sharp system through a local phone hook-up. If you are using a different host system or communications set-up (e.g. through DATAPAC...), the functions SHARP∆LINK, SET∆TERM and LOGON may require modifications. Check the ⍝SETUP for lines in SET∆TERM and LOGON and the LOGON procedure itself.

The function HELP is a reminder of the ALT/F key combinations to be used with the COMM facility.

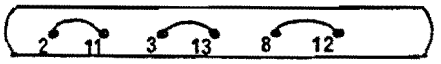
### Uploading/downloading objects.

A special program HOSTCM (available from WATCOM Publications) must be used to download functions from the remote host computer, or upload them to the host, via the HOST device provided in WATCOM APL. Unfortunately, this program is available only for "traditional" operating systems (VM/CMS, VAX/VMS, PDP-11/RSTS) and not for APL-based host systems like I.P. Sharp; the operation of HOSTCM on the latter systems has proven to be too expensive for practical use.

# Appendix C. RS-232-C cable wiring for the IBM-PC.




APL-Terminal Jumper connector (allows using TI-745 as an independent terminal):




15-pin connector

## APPENDIX D. Managing hard disk files.

This package has also been used with a hard disk, namely a TALLGRASS™ Model 3020, configured to give four additional disk drives (C, D, E and F). Drive F was used as a scratch pad while E was used to store the WATCOM APL 2.0 and the APL workspaces described herein. The following outlines the set-up and the .BAT files which were adopted to facilitate operations with the hard disk system.

### SET-UP

| Drive | Path | Files |
|---|---|---|
| C: | DOS2 | all .COM files of DOS 2.0; |
| E: | APLCODE | WATCOM APL 2.0 files; |
| | FISH | all workspaces associated with the main workspace FISH (see page 2); |
| | *username1* | working files for user 1; |
| | *username2* | working files for user 2; |
| | ..... | ..... |
| | *usernamen* | working files for user *n*; |
| F: | undefined or defined at execution time | scratch area. |

### OPERATING SYSTEM

The DOS command file GOAPL.BAT is used to tell DOS which sub-directory of drive F to use as "scratch" directory and which user files/workspaces to copy from drive E into this scratch area. The possibility of specifying which scratch directory to use on drive F allows the user to maintain and access a work directory, different from the default F:\SCRATCH.

In order to invoke APL, simply type the following after the DOS prompt:

GOAPL *path1 path2*

where

*path1* is the specification for the path of the directory in drive E containing the workspaces of the user or the files to be used from APL. If no user files are required, you must input a minus (-) character.

*path2* is the complete path specification of the scratch directory to be used in F. The directory will be created if it does not already exist. If no path is indicated, the default directory F:\SCRATCH will be used.

After entering the specified scratch directory, GOAPL copies all APL code files, the workspace FISH and its associated workspaces, and all user files identified by *path1*. Then APL is automatically started. When the user leaves APL by the )OFF command, the GOAPL.BAT file resumes and warns the user that a CLEAN-UP will be performed. The clean up consists of:

1. erasing the APL code files and the FISH workspaces;

2. copying the user files (including any created by APL) to the specified user directory (*path1*) on drive E;

3. erasing all remaining files.

The user can choose to STOP this process by pressing the CTRL/BREAK keys before steps 1 or 3.

When specified, *path1* and *path2* must be complete paths (from the root directory), since they will be prefixed as E:*path1* and F:*path2* by GOAPL.

**EXAMPLES.**

| | |
|---|---|
| >GOAPL | the directory F:\SCRATCH is activated and no user directory is specified on drive E. |
| >GOAPL JACK WORK | the directory F:\WORK is activated as the scratch area, and all user files contained in E:\JACK are copied to the scratch area; when leaving APL, the user files are copied back to E:\JACK. |
| >GOAPL - WORK | the directory F:\WORK is activated as the scratch area but no user files are copied; this form can be used to resume work in the scratch directory F:\SCRATCH without affecting pre-defined user files/workspaces. |
| >GOAPL JILL\DATA | the directory F:\SCRATCH (default) is activated as the scratch area, and all user files contained in the directory E:\JILL\DATA are copied to the scratch area; when leaving APL, the user files are copied back to E:\JILL\DATA. |

A complete listing of the batch file GOAPL.BAT follows:

```
ECHO OFF
 ASSIGN
 F:
 CD \
 IF .%1 == . GOTO SCRATCH
 IF .%2 == . GOTO SCRATCH
     MKDIR %2
     ECHO  CHANGING TO DIRECTORY  %2  ON DRIVE F:
     CHDIR %2
     GOTO  APLCODE
 :SCRATCH
     MKDIR \SCRATCH
     ECHO  CHANGING TO SCRATCH DIRECTORY ON DRIVE F:
     CHDIR \SCRATCH
 :APLCODE
     IF NOT EXIST APL.COD  COPY E:\APLCODE  F:
     IF NOT EXIST FISH     COPY E:\FISH F:
     IF .%1 == .-  GOTO RUN
     IF NOT .%1 == .       COPY E:\%1  F:
  :RUN
     ECHO ON
     WRUN APL,APLGR,APLRAF,APLEF
  REM *** CLEAN-UP --SCRATCH DIRECTORY %2 WILL NOW BE SAVED AND CLEANED-UP...
  PAUSE *** HIT  CTRL/BREAK  if you DO NOT want a clean up.
     ERASE *.COD
     ERASE APL*.DAT
     ERASE APL.BAT
     ECHO OFF
  ECHO  ... now erasing the FISH workspaces ....
     FOR %%F IN (FISH PLOT YPR SURVIVOR BEVHOLT PROJVAR) DO ERASE %%F
     FOR %%F IN (PROJECT SPA PALOHEIM REGR COMM) DO ERASE %%F
     FOR %%F IN (SPASA PRODMDL PACKIT FILE) DO ERASE %%F
     IF .%1 == .  GOTO FIN
     IF .%1 == .-  GOTO FIN
  ECHO ON
     COPY *.* E:\%1
     REM *** your files will now be erased from the scratch directory...
     PAUSE   Hit CTRL/BREAK if you wish to retain them
     ERASE *.*
     CD \
     IF .%2 == .  RMDIR SCRATCH
     IF NOT .%2 == .  RMDIR %2
   :FIN
```