Not to be cited without author's permission.

CAFSAC Res. Doc. 78/12

A programing language (APL) programs for the assessment of exploited fish stocks

Co-ordinated and revised by

William H. Marshall

E.D.P. Co-ordinator

Fisheries Systems and Data Processing Group

Marine Fish Division, Resource Branch

Bedford Institute of Oceanography

Dartmouth, Nova Scotia

Fisheries Systems and Data Processing Group contribution No. 20.

Introduction

This document presents 6 programs designed for use in assessing exploited fish stocks. The actual programs contained in the package are:

- cohort analysis (COHORT)
- catch projections (PROJECT)
- Schaefer production model (SCHNUT)
- Von Bertalanffy growth equation (VONB)
- yield per recruit (YIELD)

The original programs were written by various members of the Fisheries Systems and Data Processing Group. All known bugs have been removed from the programs but it is quite likely that there are others waiting to be found. If you do find a problem in a program I would be most grateful if you would let me know about it. I would also like to hear suggestions for improving existing programs or ideas for new programs which might be added.

The documentation in this paper is divided into three parts as follows:

- general instructions for signing on and assessing the package.
- user instructions for each program.
- listings of the acutal programs.

The first two parts should provide adequate information for the average person to start using the programs. The third section containing the actual program listings may be of interest to some people who wish to implement the programs on other computers.

2. General Instructions

These assessment programs are currently implemented on the IBM computer at the University of New Brunswick. In order to use the programs it is necessary to sign on to the APL subsystem of the UNB computer and to load the workspace which contains the programs. The steps to follow if you are using a Decwriter terminal and an acoustic coupler are as follows:

(a) Turn on the power switches on the terminal and on the acoustic coupler.

(b) Set the various option switches on the terminal and acoustic coupler to

- line (as opposed to local)
- half duplex
- 300 baud
- APL character set (sometimes labeled as alternate character set)

(c) Dial the UNB computer (506-453-4873) and wait for the carrier signal which you will hear as a high pitched whistle.

(d) Place the phone handset into the acoustic coupler (be sure its in the right direction) and then press the line feed (LF) key on the terminal keyboard.

(e) The message

TYPE APL FOLLOWED BY RETURN AND THEN YOUR APL ACCOUNT NUMBER

should then be printed on the terminal. If the message doesn't appear then try starting again from step (c).

(f) Type in APL(control A)3 followed by a return. You should then hear a "beep" to prompt you for the next line of input.

(g) Type in your APL account number and password followed by a return. A message indicating that you are signed on should be typed at the terminal.

(h) To load the workspace containing the assessment programs type in)LOAD 2081004 LIBARY followed by a return.

If you are using a terminal other than a Decwriter some of the details in the above procedure may be slightly different, but the general sequence of steps to follow will still be as indicated above.

A typical sign on sequence is shown below.

```
TYPE  APL  FOLLOWED BY RETURN AND THEN YOUR APL ACCOUNT NUMBER

APL3
)2081011:
  9.45.51 OPR:
010) 13.04.07 04/17/78 MARSHALL

   . A P L . S V .
```

The descriptions and instructions for using the individual programs are given in the next section. However before attempting to use any of the programs you should note the following points which apply to all the programs in the workspace:

1) Always press the return key after entering each line of input. No action is taken by the computer until a return has been transmitted

2) When there is no program executing and the computer is waiting for input the terminal print head will be spaced over 6 spaces from the left hand margin. Most terminals will also indicate that they are waiting for input by emitting a single "beep" when they are ready for input.

3) To use any program in the package simply type in the name of the function.

4) If a program is asking for numeric data you will be prompted by a message indicating the type of data expected. The computer will then print ⎕: and space over to column six on the next line and wait for your input. The input to such a request may consist of a number (or series of numbers seperated by blanks) or a variable name which has previously been assigned a numeric value. Typical examples of the procedure may be seen in the sample outputs which are included with the individual program descriptions.

5) If a program is asking for an answer to a particular question you will be prompted by the question followed by .... The answer to the question should be typed on the same line immediately following the prompting dots.

6) If you enter data which the program considers to be unacceptable, the data request will be repeated. In these cases you should check your input to find the mistake and then enter the correct data. Detailed explanations of what the programs consider to be errors are given with the individual program instructions.

7) Occasionally a program will stop for no apparent reason. A message consisting of the program name and a line number in square brackets will be printed (eg COHORT [21]). Such stops are usually caused by noise on the telephone lines or by the program requiring a large amount of central processor time to execute. To restart the program type in a right arrow followed by the line number given in the message (eg→21).

8) If you have entered data which you wish to save for your next APL session then save it in a new workspace before signing off. For example if you wished to call your workspace FISHDATA then you would enter the following commands to save your data.

```
)ERASE PROGS
)WSID FISHDATA
)SAVE
```

To get your data back during your next APL session you would enter the following commands after signing on.

```
)LOAD 2081004 LIBRARY
)COPY FISHDATA
```

9) When you are ready to sign off at the end of your APL session type

```
)OFF
```

Only a very rudimentary knowledge of APL is required to use the assessment programs in this package. However, a better knowledge of APL will enable you to get more out of the computer and to get the most out of your time at the terminal. For those wishing to learn APL, I would recommend the book APL; An Interactive Approach by Gilman and Rose. For a computer book it is quite readable and it contains many examples.

Cohort Analysis (COHORT)

This program calculates population numbers at age and hence the instantaneous rate of fishing mortality at age using the method of cohort analysis described in Pope (1972). The input data required by the program consists of:

- a table of catch numbers at age with rows representing ages and columns representing years. (This table should be created and assigned a name before starting the program).
- the instantaneous rate of fishing mortality for each age group in the final year.
- the instantaneous rate of fishing mortality for the oldest age group in each year.
- the instantaneous rate of natural mortality which may be either a single number to be applied to all age groups or a series of numbers representing a different rate of mortality for each age group.

The program first calculates the population numbers in the final year using the equation:

$$(1)\quad N = \frac{C \times (F + M)}{F \times (1 - \exp(-F-M))}$$

The population numbers for the oldest age group are given by:

$$(2)\quad N = \frac{C \times (F+M)}{F}$$

if fishing is complete for the oldest age group or by (1) if fishing is not complete for the oldest age group. Fishing would be considered to be incomplete if the last row of the catch table (i.e. the oldest age group) includes only the catches from that age group but not the catches of older fish from the same cohort. If the catches of older fish have been added to the last row of the catch table or if there are no older fish then fishing should be considered complete.

The population numbers for the rest of the table are calculated using the approximation:

(3) $${}_{j}N_{i} = {}_{j}C_{i} \exp\left(\frac{M_{i}}{2}\right) + {}_{j+1}N_{i+1} \exp(M_{i})$$

To start the program type in COHORT after which you will be prompted for input as follows:

NAME OF CATCH MATRIX?....

- enter the name you have given the previously created catch matrix.

ERRORS - the name entered must represent an APL variable

- the variable must be a matrix
- there should not be negative values in the matrix

FIRST YEAR AND YOUNGEST AGE GROUP

⎕ :

- enter two numbers representing the first year in the catch table and the youngest age group in the catch table respectively.

ERRORS - you must enter exactly two numbers

- both numbers must be non-negative integers.

NATURAL MORTALITY?

⎕:

- enter a value for the instantaneous rate of natural mortality (either a single number to be applied to all age groups or one number for each row of the catch matrix)

ERRORS: - you must enter exactly one number or a vector of numbers with the same number of elements as there are rows in the catch matrix.
- the natural mortality must be greater than zero.

STARTING F VALUES FOR LAST YEAR (xxxx)?

⎕ :

- enter the value of the instanteous rate of fishing mortality for the last year represented in the catch matrix (either a single number representing a fishing mortality to be applied to all age groups in the final year or a vector of values with one value for each age group)

ERRORS: - same error checks as for the values of natural mortality.

STARTING F VALUES FOR OLDEST AGE GROUP (xx)?

⎕ :

- enter the value of the instantaneous rate of fishing mortality for the oldest age group represented in the catch matrix (either a single value or a seperate value for each year)

ERRORS: - you must enter exactly one number or a vector of numbers with the same number of elements as there are columns in the catch table.

- the fishing mortality must be greater than zero.

IS FISHING COMPLETE FOR THE LAST AGE GIVE (YES OR NO)?.....

- enter either YES or NO indicating whether equation (2) or (1) should be used to calculate the population numbers for the oldest age group.

ERRORS: - you must answer either YES or NO.

The output from the program consists of two tables. The first table gives the population numbers at age. The bottom row of this table gives the total population numbers for each year. The second table shows the instantaneous rate of fishing mortality. This is calculated as:

$$(4) \qquad {}_jF_i = \ln \frac{{}_jN_i}{{}_{j+1}N_{1+1}}$$

The last row of this table gives the weighted average of the fishing mortalities where the weighing is on the basis of population numbers. A sample run of the program is shown below. The catch at age table has been previously entered and assigned the name CODCATCH.

```
      COHORT
NAME OF CATCH MATRIX?.....CODCATCH
FIRST YEAR AND YOUNGEST AGE GROUP?
⎕:
      1968 3
NATURAL MORTALITY?
⎕:
      .2
STARTING F VALUES FOR LAST YEAR (1977)?
⎕:
      .01 .13 .291 .52 .587 .708 .573 .575 .631 .308 .367 .197 .45
STARTING F VALUES FOR OLDEST AGE GROUP (15)?
⎕:
      .28 .28 .28 .28 .04 .09 .03 .4 .13 .45
IS FISHING COMPLETE FOR LAST AGE GIVEN(YES OR NO)?.....NO
```

POPULATION NUMBERS 15/ 4/78

	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977
3	94134	55121	50285	71266	24795	32980	40849	28441	30514	60644
4	85563	76790	44833	41146	58346	18906	26660	32332	21133	24682
5	33525	62688	58404	33634	31447	34836	11501	18959	17511	13629
6	18097	20131	39987	34270	20920	15499	17773	5922	10006	6113
7	12047	10587	13251	21948	19967	10620	7277	5638	2722	3672
8	4825	7225	6392	7054	12426	8676	4630	2672	1745	860
9	4527	2796	3983	3509	3453	5533	3977	1811	1057	674
10	2015	3025	1459	1954	1754	1587	2536	1373	548	482
11	6483	1257	1766	537	1099	785	630	1003	456	250
12	2575	4494	837	931	299	491	299	244	305	244
13	562	1476	3424	360	372	130	284	84	98	154
14	269	226	945	2587	256	221	49	166	26	49
15	292	94	135	648	2025	179	149	17	99	6
	264913	245910	225699	219844	177158	130442	116614	98661	86220	111460

FISHING MORTALITY 15/ 4/78

	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977
3	.004	.007	.001	.000	.071	.013	.034	.097	.012	.010
4	.111	.074	.087	.069	.316	.297	.141	.413	.239	.130
5	.310	.250	.333	.275	.508	.473	.464	.439	.852	.291
6	.336	.218	.400	.340	.478	.556	.948	.577	.803	.520
7	.311	.305	.430	.369	.634	.630	.802	.973	.953	.587
8	.346	.396	.400	.514	.609	.580	.739	.728	.751	.708
9	.203	.450	.512	.494	.577	.580	.863	.995	.584	.573
10	.272	.338	.800	.376	.603	.724	.727	.901	.586	.575
11	.166	.206	.440	.387	.606	.765	.747	.992	.425	.631
12	.356	.072	.645	.719	.635	.347	1.071	.716	.485	.308
13	.712	.247	.080	.138	.320	.771	.340	.965	.486	.367
14	.847	.314	.177	.045	.158	.192	.882	.311	1.266	.197
15	.280	.280	.280	.280	.040	.090	.030	.400	.130	.450
	.136	.144	.233	.178	.398	.369	.367	.402	.390	.132

Catch Projection (PROJECT)

This program uses the Beverton - Holt catch equation to predict population parameters over a number of years. The input required by the program consists of:

- population numbers at age for the starting year.
- catch at age for the starting year if the catch is known.
- weight at age in kilograms.
- partial recruitment values for each age group.
- instantaneous rate of natural mortality.
- an estimate of recruitment for each year to appear in the projection.
- an estimate of the standard deviation of the recruitment values.
- an estimate of either fishing mortality or total catch for each year to appear in the projection.
- maturity at age if you want the program to calculate mature numbers and mature biomass.

At the start of each year in the projection the population numbers for that year as well as either the instantaneous rate of fishing mortality or a quota or the catch at age are known. From these values the program calculates the catch at age and instantaneous rate of fishing mortality for that year if they are unknown. If there are more years to project then the program calculates numbers at age for the next year by applying the expression exp (-Z) to this years numbers and introducing new recruits based on the parameters supplied by the user.

To use the program type in PROJECT after which you will be prompted for input as follows:

FIRST YEAR AND YOUNGEST AGE GROUP?

⎕ :

- enter two numbers representing the year on which the projection is to start and the youngest age to be included in the projection.

ERRORS - you must enter 2 positive integers.

NUMBERS AT AGE FOR XXXX?

⎕ :

- enter the vector of population numbers for the year on which the projection is to start (the first number in the vector should be for the youngest age group as entered above)

ERRORS - the population numbers must not be negative.

CATCH AT AGE KNOWN FOR XXXX? .....

- enter either YES or NO. If you enter YES then the program will next ask for

CATCH AT AGE FOR XXXX?

⎕ :

- enter the vector of catch at age for the first year of the projection.

ERRORS - the catch vector must be the same length as the vector of numbers at age.

- the catch numbers must not be negative.
- the catch numbers must be less than the population numbers

WEIGHT AT AGE?

⎕ :

- enter a vector of weights at age.

ERRORS - the vector must be the same length as the vector of population numbers.

- the weights must be positive

PARTIAL RECRUITMENT?

⎕:

- enter a vector of partial recruitments.

ERRORS - the vector must be the same length as the vector of population numbers.

- the partial recruitments must all be greater than or equal to zero and less than or equal to one.

NATURAL MORTALITY?

⎕ :

- enter either a single number or a vector giving a seperate instantaneous rate of natural mortality for each age group.

ERRORS - the entry must either be a single number or a vector the same length as the vector of population numbers.

- all mortalities must be greater than zero.

NUMBER OF YEARS TO BE PROJECTED?

⎕:

- enter a number representing the number of years for which the projection is to be carried forward. (The starting year counts as one year).

ERRORS - the entry must be a single integer greater than one.

NATURAL LOG OF RECRUITMENT FOR XXXX TO XXXX?

⎕:

- enter a vector containing the natural log of recruitment values for the range of years requested. The program asks for the log of recruitments since in most cases the geometric mean is a better indicator of average recruitment than the arithmetic mean and in most cases the geometric mean is calculated using logs. If you wish to enter estimates of the actual recruitment rather than their logs then enter the vector preceded by the natural log operator ("⍟ ").

ERRORS - the entry must be a vector with number of elements equal to one less than the number of years for which the projection is to run.

STANDARD DEVIATION OF LOG RECRUITMENT? (ZERO IF RECRUITMENT FIXED)

⎕:

- enter a vector containing the standard deviation of the log of the recruitment values. If all zeros are entered recruitment will be fixed at the level determined by the previous entry. If there are any none zero entries the program will put some variation into the recruitment process.

ERRORS - the entry must be a vector the same length as the vector of log recruitment values.

- the entrys must be all greater than or equal to zero.

If any of the standard deviations were greater than zero, then the program will next ask for

NUMBER OF RUNS?

⎕:

- enter a number representing the number of times the projection is to be run.

ERRORS: - the entry must be a single integer greater than or equal to one.

QUOTA OR F VALUES FOR XXXX TO XXXX?

⎕:

- enter a vector of quota and F values for the indicated years. Any number which is less than 10 will be assumed to be an F value and any number greater than or equal to 10 will be interpreted as a quota.

ERRORS - all entries must be greater than or equal to zero.

- the number of entries must be equal to the number of years which are to be projected if catch for the current year was not provided. If catch for the current year was provided then the number of entries must be one less than the number of years to be projected.

MATURITY FIGURES TO BE CALCULATED?....

- enter YES or NO indicating whether or not you wish to get printouts of mature biomass and mature numbers. If you answer YES the program will next ask for:

MATURITY AT AGE?

⎕:

- enter a vector indicating the proportion of mature fish in each age group.

ERRORS - the number of entries must be equal to the number of entries in the number at age vector.

- the maturity values must be greater than or equal to zero and less than or equal to one.

FULL OUTPUT OR SUMMARY? (F OR S)....

- enter either an F or S indicating whether you want full output or only a summary. The full output gives seperate tables showing values at each age in each year. The summary produces a single table giving totals for each year.

When full output is requested the program will produce seperate tables showing population numbers, standard deviation of population numbers, population biomass, mature numbers, mature biomass, catch numbers, catch biomass and fishing mortalities. The summary output consists of a single table with each row representing a total for the year. The columns in the table show population numbers, standard deviation of population numbers, population biomass, mature numbers, mature biomass, catch numbers, standard deviation of catch numbers, catch biomass and mature F. After the projection you are given the option of doing more projections on the same data. If you do wish to perform additional projections the program will return to the point at which partial recruitments are requested and continue from there.

The sample run shown below uses data for Haddock the 1978 assessment.

```
      PROJECT
FIRST YEAR AND YOUNGEST AGE GROUP?
⎕:
      1977 1
NUMBERS AT AGE FOR 1977?
⎕:
      NATAGE
CATCH AT AGE KNOWN FOR 1977?.....YES
CATCH AT AGE FOR 1977?
⎕:
      CATAGE
WEIGHT AT AGE? (KG)
⎕:
      WTATAGE
PARTIAL RECRUITMENT?
⎕:
      S
NATURAL MORTALITY?
⎕:
      .2
NUMBER OF YEARS TO BE PROJECTED?
⎕:
      5
NATURAL LOG OF RECRUITMENT FOR 1978 TO 1981?
⎕:
      4⍴⍟50000
STANDARD DEVIATION OF LOG RECRUITMENT? (ZERO IF RECRUITMENT FIXED)
⎕:
      4⍴1.5
NUMBER OF RUNS?
⎕:
      5
QUOTA OR F VALUES FOR 1978 TO 1981?
⎕:
      .3 .3 25000 25000
MATURITY FIGURES TO BE CALCULATED?.....NO
FULL OUTPUT OR SUMMARY?(F OR S).....S
```

YEAR	POP NUMBERS	STANDARD DEV	POP BIOMASS	CATCH NUMBERS	STANDARD DEV	CATCH BIOMASS	MATURE F
1977	178043	.00	157231	13139	0	21816.20	.2925
1978	181098	37379.38	172202	15555	3	25245.84	.3000
1979	180388	33184.33	176518	15262	230	26097.99	.3000
1980	255607	171148.82	199372	14158	1104	25000.07	.2737
1981	336979	231173.64	234544	13946	1427	25000.14	.2637

```
MORE PROJECTIONS WITH THIS DATA?.....YES
PARTIAL RECRUITMENT?
⎕:
      S
NATURAL MORTALITY?
⎕:
      .2
NUMBER OF YEARS TO BE PROJECTED?
⎕:
      8
```

```
NATURAL LOG OF RECRUITMENT FOR 1978 TO 1984?
⎕:
      7⍴⍟50000
STANDARD DEVIATION OF LOG RECRUITMENT? (ZERO IF RECRUITMENT FIXED)
⎕:
      7⍴1.5
NUMBER OF RUNS?
⎕:
      5
QUOTA OR F VALUES FOR 1978 TO 1984?
⎕:
      .3 .3,5⍴25000
MATURITY FIGURES TO BE CALCULATED?.....NO
FULL OUTPUT OR SUMMARY?(F OR S).....F
```

POPULATION NUMBERS 16/ 4/78

	1977	1978	1979	1980	1981	1982	1983	1984
1	29423	27469	50784	242147	136498	220752	96100	107881
2	67964	24088	22488	41574	198236	111745	180719	78674
3	27757	55036	19557	18257	33776	161034	90635	147177
4	15401	20461	40459	14377	13561	25234	125754	69289
5	16270	10691	14144	27969	10029	9636	18241	97930
6	14772	10478	6842	9052	18139	6531	6539	12698
7	2103	9027	6355	4150	5580	11238	4183	4413
8	2475	1305	5560	3914	2596	3507	7291	2853
9	578	1570	822	3504	2502	1666	2318	5045
10	1124	386	1043	546	2354	1687	1149	1657
11	146	780	267	720	381	1645	1200	841
12	30	99	525	179	490	260	1147	864
	178043	161389	168845	366391	424143	554936	535275	529322

STANDARD DEVIATION OF POPULATION NUMBERS 16/ 4/78

	1977	1978	1979	1980	1981	1982	1983	1984
1	0	27523	42329	427160	212016	318236	85898	88349
2	0	0	22532	34653	349698	173567	260521	70319
3	0	0	0	18293	28171	284056	140622	212221
4	0	0	0	0	13664	21467	224483	108146
5	0	0	0	0	90	9948	15924	177805
6	0	0	0	0	237	274	7036	11382
7	0	0	0	0	89	599	565	4856
8	0	0	0	0	39	189	1004	631
9	0	0	0	0	35	83	301	1077
10	0	0	0	0	26	70	125	313
11	0	0	0	0	3	55	107	131
12	0	0	0	0	5	9	105	126
	0	27523	60971	416973	336200	459137	320381	235178

POPULATION BIOMASS 16/ 4/78

	1977	1978	1979	1980	1981	1982	1983	1984
1	7944.27	7416.65	13711.57	65379.72	36854.56	59603.08	25947.10	29127.99
2	27865.30	9876.04	9219.98	17045.48	81276.70	45815.41	74094.75	32256.23
3	21373.17	42377.42	15058.53	14058.21	26007.71	123996.50	69788.99	113326.42
4	18172.90	24143.51	47741.21	16964.52	16002.20	29776.42	148389.22	81760.45
5	27170.77	17854.24	23620.73	46707.47	16749.08	16092.81	30462.36	163542.56
6	33828.54	23995.59	15668.76	20729.40	41538.27	14955.00	14974.90	29078.63
7	6308.79	27081.90	19066.45	12450.11	16740.16	33715.38	12549.51	13238.32
8	7894.43	4161.40	17735.61	12486.39	8279.68	11187.06	23258.26	9102.52
9	2040.30	5542.44	2902.37	12369.72	8832.42	5882.68	8181.32	17808.24
10	4023.07	1381.49	3733.70	1955.20	8427.36	6038.16	4113.12	5932.05
11	508.98	2720.55	930.16	2513.91	1328.56	5741.91	4188.04	2936.76
12	100.89	329.63	1753.48	599.52	1637.43	868.04	3829.35	2886.20
	157231.42	166880.86	171142.56	223259.64	263674.13	353672.47	419776.94	500996.36

CATCH NUMBERS 16/ 4/78

	1977	1978	1979	1980	1981	1982	1983	1984
1	2	2	4	19	12	20	7	4
2	674	183	171	290	1403	945	867	548
3	2512	5102	1813	1538	2682	6749	5450	4763
4	2129	2895	5724	1933	1627	2684	5573	3681
5	3159	2124	2810	5290	1868	1500	2484	4264
6	3412	2474	1615	2037	4018	1294	1045	1891
7	464	2037	1434	892	1181	2124	635	570
8	507	273	1165	781	510	615	1027	320
9	97	269	141	572	402	239	267	465
10	156	55	148	74	313	201	110	128
11	23	126	43	110	57	223	131	75
12	4	13	71	23	62	29	105	65
	13139	15553	15140	13558	14135	16624	17703	16775

CATCH BIOMASS 16/ 4/78

	1977	1978	1979	1980	1981	1982	1983	1984
1	1	1	1	5	3	5	2	1
2	276	75	70	119	575	388	355	225
3	1934	3929	1396	1184	2066	5196	4197	3668
4	2512	3416	6754	2281	1920	3168	6577	4343
5	5276	3547	4693	8834	3119	2505	4148	7121
6	7813	5665	3699	4664	9201	2964	2394	4331
7	1392	6110	4302	2676	3542	6373	1905	1710
8	1617	872	3716	2491	1626	1963	3277	1022
9	342	951	498	2019	1419	845	943	1643
10	558	196	531	264	1121	719	394	459
11	80	439	150	386	201	777	459	262
12	13	45	238	77	207	99	350	216
	21816	25245	26048	25000	25000	25000	25000	25000

FISHING MORTALITY 16/ 4/78

	1977	1978	1979	1980	1981	1982	1983	1984
1	.000	.000	.000	.000	.000	.000	.000	.000
2	.011	.008	.008	.008	.008	.007	.006	.005
3	.105	.108	.108	.102	.100	.090	.074	.063
4	.165	.169	.169	.160	.158	.142	.116	.099
5	.240	.246	.246	.233	.230	.207	.169	.145
6	.292	.300	.300	.284	.280	.252	.205	.176
7	.277	.285	.285	.269	.265	.239	.195	.167
8	.255	.262	.262	.248	.244	.220	.179	.154
9	.204	.209	.209	.198	.195	.176	.143	.123
10	.166	.170	.170	.161	.159	.143	.116	.100
11	.190	.195	.195	.185	.182	.164	.134	.115
12	.157	.161	.161	.153	.151	.136	.111	.095

MORE PROJECTIONS WITH THIS DATA?.....NO

Schaefer Production Model (SCHNUT)

This program fits a Schaefer production model to a series of catch and catch per unit effort data. The methods used in the program are based on the paper by Schute (1977). Specifically the program uses the linear approximation which allows linear regression techiques to be used to find the least squares estimates of the parameters. The input to the program consists of:

- a vector of catch per unit efforts.
- a vector of efforts.

To use the program type SCHNUT after which the program will prompt for input as follows:

EFFORT VALUES?

⎕:

- enter a vector of effort values.

ERRORS - the effort values must be greater than or equal to zero.

- there must be at least four values so that the four parameters can be estimated.

C/E VALUES?

⎕:

- enter a vector of catch per unit effort values.

ERRORS - there must be the same number of values in this vector as in the effort vector.

- the catch per unit efforts must be greater than or equal to zero.

The output from the program consists of the parameters of the model, namely rate of increase (R), carrying capacity (K) and catchability (Q). The program also gives the failure index which is an indication of how well the model fits the data with zero signifying a perfect fit and one indicating the worst possible fit. The program will also give the predicted MSY and 2/3 MSY values as well as a table of next years predicted catch as a function of next year effort. The input used in the sample output shown below consists of data for Gulf of St. Lawrence Cod.

```
   SCHNUT
EFFORT VALUES?
⎕:
      EFFORT
C/E VALUES?
⎕:
      CPUE
RATE OF INCREASE (R)..........1.156374623
CARRYING CAPACITY (K).........194191.2174
CATCHABILITY (Q)..............1.549602283E¯5
FAILURE INDEX.................0.3989991765
```

	MSY	2/3 MSY
EQUILIBRIUM EFFORT............	37311.981	24874.654
EQUILIBRIUM CUE...............	1.50460	2.00613
EQUILIBRIUM BIOMASS...........	97095.61	129460.81
EQUILIBRIUM CATCH.............	56139.45	49901.73

EFFORT	CATCH	LOWER	UPPER
1063.100	1824.275	1041.339	2952.487
2126.200	3622.547	2073.480	5853.531
3189.300	5395.095	3096.467	8703.753
4252.400	7142.201	4110.343	11503.775
5315.500	8864.142	5115.150	14254.218
6378.600	10561.195	6110.928	16955.707
7441.700	12233.636	7097.715	19608.863
8504.800	13881.738	8075.549	22214.312
9567.900	15505.771	9044.466	24772.678
10631.000	17106.007	10004.500	27284.584
11694.100	18682.713	10955.683	29750.657
12757.200	20236.156	11898.046	32171.522
13820.300	21766.600	12831.619	34547.802
14883.400	23274.308	13756.430	36880.124
15946.500	24759.542	14672.505	39169.111
17009.600	26222.561	15579.869	41415.390
18072.700	27663.622	16478.544	43619.583
19135.800	29082.981	17368.552	45782.314
20198.900	30480.892	18249.914	47904.208
21262.000	31857.607	19122.646	49985.886

Von Bertalanffy growth equation (VONB)

This program calculates the parameters L_∞, K and t_o for Von Bertalanffy's equation given by:

$$(1)\quad L_t = L_\infty \{ 1 - \exp - K(t - t_o)\}$$

The input required by the program consists of:

- a vector of ages
- a vector of lengths

The program calculates a first approximation to the parameters by doing a regression of L_t on L_{t+1}.

The slope of the regression line should be equal to k, hence the first approximation to K is given by - ln k. Also the y-axis intercept of the line should be equal to $L_\infty (1 - k)$ from which an estimate of L_∞ can be calculated. The starting estimate of t_o is given by rearranging equation (1) to give:

$$(2)\quad t_o = t - \frac{\ln \left(1 - \frac{L}{L_\infty}\right)}{K}$$

These parameter estimates are then used as starting values for a non-linear regression routine. The non-linear regression subroutine uses a method suggested in section 10.3 of Draper and Smith (1966).

To use the program type in VONB after which you will be prompted for input as follows:

AGES?

⎕ : - enter a vector of ages

ERRORS:- the ages must be positive

- there must be more than three ages to allow the three parameters to be fit.

LENGTH AT AGE?
⎕ :

- enter a vector of lengths corresponding to the ages entered above

ERRORS: - the number of elements in the length vector must be the same as the number of elements in the age vector
- the lengths must be positive

The output from the program consists of the calculated parameters and a table showing observed versus predicted lengths along with the absolute and relative errors. The program also outputs an analysis of variance table showing the variation accounted for by the regression and the calculated R^2 value.

The sample run for this program uses data for Pollock measured in 1976.

```
      VONB
AGES?
⎕:
      1+⍳15
LENGTH AT AGE?
⎕:
      .79 1.52 2.33 3.01 3.76 4.51 5.37 6.65 7.21 7.75 7.13 8.12 7.82 9.42 9.82

                  VON BERTALANAFFY PARAMETERS                15/ 4/78

L∘∘ -------- 15.51143483
K   -------- 0.06530964117        L=L∘∘(1-EXP(KT∘-KT))
T∘  -------- 1.412217952
```

AGE	OBSERVED LENGTH	PREDICTED LENGTH	ERROR	REL ERROR
2	.790	.584	¯.206	¯.261
3	1.520	1.528	.008	.005
4	2.330	2.412	.082	.035
5	3.010	3.240	.230	.076
6	3.760	4.016	.256	.068
7	4.510	4.743	.233	.052
8	5.370	5.424	.054	.010
9	6.650	6.061	¯.589	¯.089
10	7.210	6.659	¯.551	¯.076
11	7.750	7.219	¯.531	¯.069
12	7.130	7.743	.613	.086
13	8.120	8.234	.114	.014
14	7.820	8.694	.874	.112
15	9.420	9.125	¯.295	¯.031
16	9.820	9.529	¯.291	¯.030

SOURCE	SS	DF	MS
REGRESSION	598	3	199.2
RESIDUAL	2.5	12	.21
TOTAL	600	15	40.0

```
R*2 -------- 97.8614915
```

Virtual Population Analysis (VPA)

This program performs a virtual population analysis of catch at age data to give population numbers at age. The input data required, editing checks and output generated are identical with the cohort analysis program (COHORT) and for a detailed description the user should refer to that section.

The program first calculates the numbers at age for the last year in the catch table using the equation:

$$(1)\quad N = \frac{C\ x(F + M)}{F\ x\ (1 - \exp\ (-F-M))}$$

The population numbers for the oldest age group are also calculated using (1) if fishing is incomplete or by:

$$(2)\quad N = \frac{C\ x\ (F+M)}{F}$$

if fishing is complete for the oldest age group.

The population numbers at age and the instantaneous rate of fishing mortality for the remainder of the table are calculated by fitting the Beverton-Holt catch equation to the available data. A slight rearrangement of the basic equation gives:

$$(3)\quad {}_jC_i = \frac{{}_{j+1}N_{i+1}\ x\ {}_jF_i\ (\exp\ ({}_jF_i + M_i) - 1)}{{}_jF_i + Mi}$$

This equation involves only one unknown, namely ${}_jF_i$, which the program calculates using the Newton-Raphson method of successive approximations. An initial estimate of ${}_jF_i$ is calculated based on the cohort analysis approximation:

(4) $_jN_i = {_jC_i} \exp\left(\frac{Mi}{2}\right) + {_{j+1}N_{i+1}} \exp(Mi)$

giving:

(5) $_jF_i = \ln\left(\frac{_jN_i}{_{j+1}N_{i+1}}\right) - Mi$

Successive values of $_jF_i$ are then calculated, based on previous estimates, until a value is found which satisfied:

(6) $\left| \frac{_jF_i\,(\exp(_jF_i + Mi) - 1)}{_jF_i + M_i} - \frac{_jC_i}{_{j+1}N_{i+1}} \right| < 10^{-5}$

Once a value of $_jF_i$ has been found the population numbers are calculated by:

(7) $_jN_i = {_{j+1}N_{i+1}} \exp(_jF_i + Mi)$

A sample run of the virtual population analysis program is shown below. The catch at age matrix has been previously assigned the name CODCATCH.

```
      VPA
NAME OF CATCH MATRIX?.....CODCATCH
FIRST YEAR AND YOUNGEST AGE GROUP?
⎕:
      1968 3
NATURAL MORTALITY?
⎕:
      .2
STARTING F VALUES FOR LAST YEAR (1977)?
⎕:
      .01 .13 .291 .52 .587 .708 .573 .575 .631 .308 .367 .197 .45
STARTING F VALUES FOR OLDEST AGE GROUP (15)?
⎕:
      .28 .28 .28 .28 .04 .09 .03 .4 .13 .45
IS FISHING COMPLETE FOR LAST AGE GIVEN(YES OR NO)?.....NO
```

POPULATION NUMBERS 15/ 4/78

	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977
3	93258	54592	49759	70440	24576	32726	40502	28401	30513	60644
4	84844	76073	44401	40716	57670	18731	26452	32051	21108	24682
5	33234	62125	57830	33290	31101	34372	11384	18800	17357	13629
6	17932	19943	39592	33897	20680	15318	17496	5859	9926	6113
7	11950	10483	13114	21714	19722	10485	7187	5562	2694	3672
8	4775	7163	6321	6975	12277	8567	4568	2647	1732	860
9	4493	2764	3948	3466	3412	5466	3923	1788	1051	674
10	2002	3001	1441	1939	1729	1567	2503	1358	545	482
11	6476	1249	1752	531	1090	773	623	991	455	250
12	2566	4492	832	924	295	488	294	242	304	244
13	557	1474	3422	359	371	128	283	83	97	154
14	267	225	944	2586	256	221	49	165	26	49
15	292	94	135	648	2025	179	149	17	99	6
	262646	243677	223490	217486	175203	129020	115412	97963	85907	111460

FISHING MORTALITY 15/ 4/78

	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977
3	.004	.007	.001	.000	.072	.013	.034	.097	.012	.010
4	.112	.074	.088	.069	.317	.298	.141	.413	.237	.130
5	.311	.251	.334	.276	.508	.475	.464	.439	.844	.291
6	.337	.219	.401	.342	.479	.557	.946	.577	.794	.520
7	.312	.306	.431	.370	.634	.631	.799	.966	.942	.587
8	.347	.396	.401	.515	.609	.581	.738	.723	.744	.708
9	.204	.452	.511	.495	.578	.581	.861	.987	.579	.573
10	.272	.338	.798	.376	.605	.723	.727	.893	.581	.575
11	.166	.207	.440	.388	.603	.767	.745	.983	.422	.631
12	.354	.072	.640	.713	.635	.345	1.070	.711	.482	.308
13	.707	.246	.080	.138	.318	.769	.338	.964	.482	.367
14	.838	.312	.177	.045	.158	.192	.872	.309	1.250	.197
15	.280	.280	.280	.280	.040	.090	.030	.400	.130	.450
	.136	.145	.234	.179	.399	.370	.366	.400	.385	.132

Yield per Recruit (YIELD)

This program uses Thompson and Bell's method to determine the yield for a single recruit at various levels of fishing mortality. The program also attempts to calculate F_{max} and $F_{0.1}$.

In cases where the yield keeps increasing over a wide range of fishing mortality the program may be unable to find F_{max} or $F_{0.1}$. The input required by the program consists of:

- a vector of weights at age. (The weights should be entered in kilograms.)
- a vector of partial recruitments.
- the instantaneous rate of natural mortality

The program first calculates the yield from a single recruit at levels of F ranging from 0 to 10 in steps of .05. The yield at each age is calculated as the weight at age times the numbers caught, determined from the Beverton-Holt catch equation. The overall yield is the sum of the yields from each age group. The vector of calculated yields is then searched for a maximum to determine F_{max}. To find $F_{0.1}$ the program first calculates the angle of the yield curve at the origin. The calculated yield points are then rotated by one tenth of this angle and the resulting vector is then searched for a maximum which will be the point corresponding to $F_{0.1}$. If the maximum does not occur in the range of F's from 0 to 10, then the program will not be able to find the F_{max} or $F_{0.1}$ points.

To start the program type in YIELD after which you will be prompted for input as follows:

WEIGHT AT AGE? (KG)

⎕:

- enter a vector of weights in kilograms representing the weight of fish at successive ages.

  ERRORS - the weights must all be positive

PARTIAL RECRUITMENT?

⎕:

- enter a vector of partial recruitment values with one value for each age.

  ERRORS: - the vector must be the same length as the weight vector
  - each number in the vector must be greater than or equal to zero and less than or equal to one

NATURAL MORTALITY?

⎕:

- enter either a single number or a separate mortality for each age group

  ERRORS: - you must enter either a single number or a vector the same length as the weight and partial recruitment vectors
  - all mortalities must be greater than zero

The output from the program consists of a table showing number of fish caught from each recruit, weight of fish caught from each recruit, average weight of fish caught from each recruit (ie. column 3 divided by column 2) and yield per unit of effort. The yield per unit of effort assumes that effort is proportional to fishing mortality. The values shown in this column are calculated by dividing column 3 by column 1 and then multiplying the result by an appropriate factor to scale the results on a range of zero to one.

The table gives results over a range of fishing mortalities and the F_{max} and $F_{0.1}$ points are labelled if they were found. The program also produces a plot of yield per recruit vs. fishing mortality.

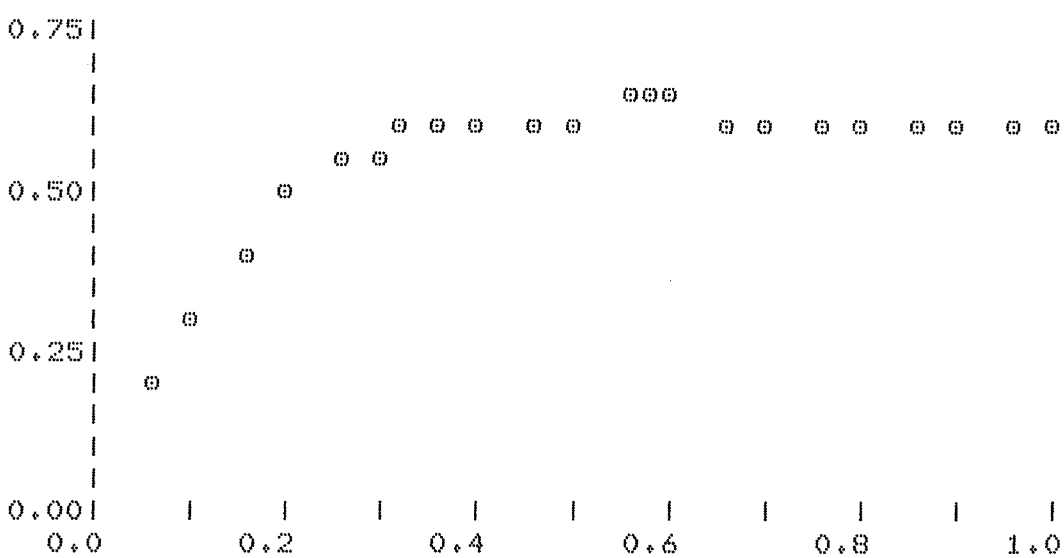
A sample run of the program is shown below using data for Haddock.

YIELD
WEIGHT AT AGE? (KG)
⎕:
WTATAGE
PARTIAL RECRUITMENT?
⎕:
S
NATURAL MORTALITY?
⎕:
.2

YIELD PER RECRUIT 15/ 4/78

	FISHING MORTALITY	CATCH (NUMBER)	YIELD (KG)	AVG. WEIGHT (KG)	YIELD PER UNIT EFFORT
	.050	.08795	.189	2.150	1.000
	.100	.15695	.325	2.068	.858
	.150	.21173	.421	1.990	.743
	.200	.25575	.490	1.916	.648
	.250	.29158	.538	1.846	.569
	.300	.32112	.572	1.780	.504
F0.1---	.328	.33530	.585	1.745	.472
	.350	.34577	.594	1.718	.449
	.400	.36662	.609	1.661	.403
	.450	.38445	.618	1.608	.363
	.500	.39988	.623	1.559	.330
	.550	.41338	.626	1.513	.301
FMAX---	.580	.42069	.626	1.488	.285
	.600	.42531	.626	1.471	.276
	.650	.43594	.624	1.432	.254
	.700	.44550	.622	1.396	.235
	.750	.45416	.619	1.363	.218
	.800	.46205	.616	1.332	.203
	.850	.46929	.612	1.304	.190
	.900	.47596	.608	1.277	.179
	.950	.48215	.604	1.252	.168
	1.000	.48790	.600	1.229	.159

YIELD PER RECRUIT(KG) VS. FISHING MORTALITY



```
.........................15/ 4/78
    ∇ COHORT;J;I;M;MORT;FI;FC;X;YR;AG;Y;TIT;NAM;CATCH
[1]    NAM←⎕NL 2
[2]   ER0:⍞←'NAME OF CATCH MATRIX?.....'
[3]    →(∨/NAM∧.=X←(¯1↑⍴NAM)↑(' '≠X)/X←,⍞)/OK1
[4]    →ER0,0↑⍞←'NAMED MATRIX DOES NOT EXIST, RE-ENTER '
[5]   OK1:→(2=⍴⍴CATCH←⍎X)/OK2
[6]    →ER0,0↑⍞←((' '≠X)/X),' IS NOT A MATRIX.'
[7]   OK2:→(∧/0≤,CATCH)/ER1
[8]    →0,0↑⍞←'NEGATIVE VALUES IN CATCH MATRIX.'
[9]   ER1:'FIRST YEAR AND YOUNGEST AGE GROUP?'
[10]   →(∧/(X=⌈X),(0≤2↑X),2=⍴X←,⎕)/OK3
[11]  →ER1:0↑⍞←'MUST BE 2 NON-NEGATIVE INTEGERS, RE-ENTER '
[12]  OK3:YR←(¯1+X[1])+⍳J←(⍴CATCH)[2]
[13]   AG←(¯1+X[2])+⍳I←(⍴CATCH)[1]
[14]  ER2:'NATURAL MORTALITY?'
[15]   →(∨/(1,I)=⍴M←,⎕)/OK4
[16]   →ER2,0↑⍞←'EITHER 1 OR ',(⍕I),' NUMBERS, RE-ENTER '
[17]  OK4:→(∧/M>0)/OK5
[18]   →ER2,0↑⍞←'MORTALITIES MUST BE POSITIVE, RE-ENTER '
[19]  OK5:MORT←⍉((J,I)⍴M)
[20]   F←(I,J)⍴0
[21]  ER3:'STARTING F VALUES FOR LAST YEAR (',(⍕¯1↑YR),')?'
[22]   →(∨/(1,I)=⍴FI←,⎕)/OK6
[23]   →ER3,0↑⍞←'EITHER 1 OR ',(⍕I),' NUMBERS, RE-ENTER '
[24]  OK6:→(∧/FI>0)/OK7
[25]   →ER3,0↑⍞←'MORTALITIES MUST BE POSITIVE, RE-ENTER '
[26]  OK7:F[;J]←I⍴FI
[27]  ER4:'STARTING F VALUES FOR OLDEST AGE GROUP (',(⍕¯1↑AG),')?'
[28]   →(∨/(1,J)=⍴FC←,⎕)/OK8
[29]   →ER4,0↑⍞←'EITHER 1 OR ',(⍕J),' NUMBERS, RE-ENTER '
[30]  OK8:→(∧/FC>0)/OK9
[31]   →ER4,0↑⍞←'MORTALITIES MUST BE POSITIVE, RE-ENTER '
[32]  OK9:F[I;]←J⍴FC
[33]   POP←(I,J)⍴0
[34]   POP[;J]←(CATCH[;J]×FI+MORT[;J])÷FI×1-*-FI+MORT[;J]
[35]   POP[I;]←(CATCH[I;]×FC+MORT[I;])÷FC
[36]  ER5:⍞←'IS FISHING COMPLETE FOR LAST AGE GIVEN(YES OR NO)?.....'
[37]   →((∧/'YES'=3↑X),~∧/'NO'=2↑X←(' '≠X)/X←,⍞)/SK1,ER5
[38]   POP[I;]←POP[I;]÷1-*-FC+MORT[I;]
[39]  SK1:Y←J-1
[40]  AA:X←MORT[⍳I-1;Y]
[41]   POP[⍳I-1;Y]←(CATCH[⍳I-1;Y]×*X÷2)+(POP[1↓⍳I;Y+1]×*X)
[42]   →(1≤Y←Y-1)/AA
[43]   F[⍳I-1;⍳J-1]←(⍟(¯1 ¯1 ↓POP)÷ 1 1 ↓POP)- ¯1 ¯1 ↓MORT
[44]   TIT←'POPULATION NUMBERS'
[45]   0 OUT POP,[1]+/[1] POP
[46]   TIT←'FISHING MORTALITY'
[47]   3 OUT F,[1]((+/[1] POP×F)÷+/[1] POP)
```

```
.........................15/ 4/78
    ∇ PROJECT;YR;AG;X;M;REF;ANS;REC;SDR;Y;POP;CAT;WGT;SEL;MAT;MEAN;SDEV

[1]   ER1:'FIRST YEAR AND YOUNGEST AGE GROUP?'
[2]    →(∧/(X=⌈X),(0≤2↑X),2=⍴X←,⎕)/ER2
[3]    →ER1,0↑⍞←'MUST BE 2 NON-NEGATIVE INTEGERS, RE-ENTER '
[4]   ER2:'NUMBERS AT AGE FOR ',(⍕X[1]),'?'
[5]    →(∧/0≤POP←,⎕)/ER3
[6]    →ER2,0↑⍞←'POPULATION NUMBERS MUST BE POSITIVE, RE-ENTER '
[7]   ER3:AG←¯1+X[2]+⍳⍴POP
[8]    CAT←(⍴AG)⍴¯1
[9]   ER4:⍞←'CATCH AT AGE KNOWN FOR ',(⍕X[1]),'?.....'
[10]   →((∧/'NO'=2↑Y),∨∧/'YES'=3↑Y←(' '≠Y)/Y←,⍞)/ER6,ER4
[11]  ER5:'CATCH AT AGE FOR ',(⍕X[1]),'?'
[12]   →(∧/(CAT≥0),(⍴AG)=⍴CAT←,⎕)/SK1
[13]   →ER5,0↑⍞←'ONE NON-NEGATIVE CATCH FOR EACH OF ',(⍕⍴AG),' AGES, RE-ENTE
       R '
[14]  SK1:→(∧/CAT≤POP)/ER6
[15]   →ER5,0↑⍞←'CATCHES MUST BE LESS THAN POPULATION NUMBERS, RE-ENTER '
[16]  ER6:'WEIGHT AT AGE? (KG)'
[17]   →(∧/(WGT>0),(⍴AG)=⍴WGT←,⎕)/ER7
[18]   →ER6,0↑⍞←'ONE NON-NEGATIVE WEIGHT FOR EACH OF ',(⍕⍴AG),' AGES, RE-ENT
       ER '
[19]  ER7:'PARTIAL RECRUITMENT?'
[20]   →((⍴AG)=⍴SEL←,⎕)/SK2
[21]   →ER7,0↑⍞←'ONE VALUE FOR EACH OF ',(⍕⍴AG),' AGES, RE-ENTER '
[22]  SK2:→(∧/(SEL≥0),SEL≤1)/ER8
[23]   →ER7,0↑⍞←'PARTIAL RECRUITMENTS MUST BE BETWEEN 0 AND 1, RE-ENTER '
[24]  ER8:'NATURAL MORTALITY?'
[25]   →(∨/(1,⍴AG)=⍴M←,⎕)/SK3
[26]   →ER8,0↑⍞←'EITHER 1 OR ',(⍕⍴AG),' VALUES, RE-ENTER '
[27]  SK3:→(∧/M>0)/ER9
[28]   →ER8,0↑⍞←'MORTALITIES MUST BE POSITIVE, RE-ENTER '
[29]  ER9:'NUMBER OF YEARS TO BE PROJECTED?'
[30]   →(∧/(1=⍴YR),(YR=⌈YR),1<YR←,⎕)/SK4
[31]   →ER9,0↑⍞←'A POSITIVE INTEGER GREATER THAN ONE, RE-ENTER '
[32]  SK4:YR←¯1+X[1]+⍳YR
[33]  ER10:'NATURAL LOG OF RECRUITMENT FOR ',(⍕YR[2]),' TO ',(⍕¯1↑YR),'?'
[34]   →((¯1+⍴YR)=⍴REC←,⎕)/ER11
[35]   →ER10,0↑⍞←(⍕¯1+⍴YR),' VALUES, RE-ENTER '
[36]  ER11:'STANDARD DEVIATION OF LOG RECRUITMENT? (ZERO IF RECRUITMENT FIXE
       D)'
[37]   →(∧/(SDR≥0),(⍴REC)=⍴SDR←,⎕)/SK5
[38]   →ER11,0↑⍞←(⍕⍴REC),' NON-NEGATIVE VALUES, RE-ENTER '
[39]  SK5:REF←1
[40]   →(∧/SDR=0)/SK6
[41]  ER12:'NUMBER OF RUNS?'
[42]   →(∧/(1=⍴REF),(REF=⌈REF),1≤REF←,⎕)/SK6
[43]   →ER12,0↑⍞←'MUST BE A POSITIVE INTEGER, RE-ENTER '
[44]  SK6:Y←YR[1]+∨/¯1≠,CAT
[45]  ER13:'QUOTA OR F VALUES FOR ',(⍕Y),' TO ',(⍕¯1↑YR),'?'
[46]   →(∧/(ANS≥0),(⍴YR)=⍴ANS←((Y≠YR[1])/0),,⎕)/ER14
[47]   →ER13,0↑⍞←(⍕1+YR[2]-Y),' NON-NEGATIVE NUMBERS, RE-ENTER '
[48]  ER14:⍞←'MATURITY FIGURES TO BE CALCULATED?.....'
[49]   MAT←(⍴AG)⍴0
[50]   →((∧/'NO'=2↑Y),∨∧/'YES'=3↑Y←(' '≠Y)/Y←⍞)/SK8,ER14
[51]  ER15:'MATURITY AT AGE?'
[52]   →((⍴AG)=⍴MAT←,⎕)/SK7
[53]   →ER15,0↑⍞←'ONE VALUE FOR EACH OF ',(⍕⍴AG),' AGES, RE-ENTER '
[54]  SK7:→(∧/(MAT≥0),MAT≤1)/SK8
[55]   →ER15,0↑⍞←'MATURITYS MUST BE BETWEEN 0 AND 1, RE-ENTER '
[56]  SK8:PROJ
[57]   PROJOUT
[58]  SK9:⍞←'MORE PROJECTIONS WITH THIS DATA?.....'
[59]   →((∧/'YES'=3↑Y),∨∧/'NO'=2↑Y←(Y≠' ')/Y←,⍞)/ER7,SK9
```

```
........................17/ 4/78
    ∇ SCHNUT;EFF;CUE;Y;X;B;VAR;S;XBAR;SS;FDX;SZ;NEF;UN;NUM;XN1;CN;VAR;RN

[1]   ER1:'EFFORT VALUES?'
[2]    →(∧/0≤EFF←,⎕)/ER2
[3]    →ER1,0↑⍞←'EFFORT MUST BE POSITIVE. RE-ENTER '
[4]   ER2:'C/E VALUES?'
[5]    →(∧/(CUE≥0),(⍴EFF)=⍴CUE←,⎕)/ER3
[6]    →ER2,0↑⍞←'NEED 1 POSITIVE C/E FOR EACH OF ',(⍕⍴EFF),' EFFORT VALUES.
       RE-ENTER '
[7]   ER3:Y←⍟(1↓CUE)÷¯1↓CUE
[8]    X←((⍴Y)⍴1),-(0.5×(¯1↓EFF)+1↓EFF),[1.5] 0.5×(¯1↓CUE)+1↓CUE
[9]    B←(⌹(⍉X)+.×X)+.×(⍉X)+.×Y
[10]   VAR←(S←+/(Y-X+.×B)*2)÷¯3+⍴Y
[11]   X←X+(⍴X)⍴XBAR←-(+/[1] X)÷1↑⍴X
[12]   SS←(⍉X)+.×X
[13]   FDX←S÷SZ←+/(Y-(+/Y)÷⍴Y)*2
[14]   NEF←0.1×(¯1↑EFF)×⍳20
[15]   XN1←(0.5×NEF+¯1↑EFF)-XBAR[2]
[16]   UN←(¯1↑CUE)-XBAR[3]
[17]   NUM←(SS[3;3]×XN1*2)+(SS[2;2]×UN*2)-2×XN1×SS[2;3]×UN
[18]   RN←(((1+⍴Y)÷⍴Y)+NUM÷(SS[2;2]×SS[3;3])-SS[2;3]*2)*0.5
[19]   CN←(B[1]-B[2]×¯1↑EFF)÷B[1]-B[2]×NEF
[20]   CN←CN×((*B[1]-B[2]×NEF)-1)÷1-*-B[1]-B[2]×¯1↑EFF
[21]   CN←(⍉(3,⍴CN)⍴CN)×(1-*-B[3]×¯1↑CUE)×*(VAR*0.5)×RN∘.× ¯1 0 1 ×0.025 STU
       D ¯3+⍴Y
[22]   CN←(⍉(3,⍴NEF)⍴NEF)×(⍟1+CN)÷B[3]
[23]   'RATE OF INCREASE (R)..........',⍕B[1]
[24]   'CARRYING CAPACITY (K).........',⍕B[1]÷B[2]×B[3]
[25]   'CATCHABILITY (Q)..............',⍕B[2]
[26]   'FAILURE INDEX.................',⍕FDX
[27]   ' '
[28]   '                                                 MSY       2/3 MSY'
[29]   '                                                 ---       -------'
[30]   'EQUILIBRIUM EFFORT............', 12 3 ⍕(B[1]÷2×B[2])×1,2÷3
[31]   'EQUILIBRIUM CUE...............', 12 5 ⍕(0.5×B[1]÷B[3])×1,4÷3
[32]   'EQUILIBRIUM BIOMASS...........', 12 2 ⍕(0.5×B[1]÷B[2]×B[3])×1,4÷3
[33]   'EQUILIBRIUM CATCH.............', 12 2 ⍕(0.25×(B[1]*2)÷B[2]×B[3])×1,8÷
       9
[34]   ' '
[35]   '       EFFORT       CATCH        LOWER        UPPER'
[36]   '       ------       -----        -----        -----'
[37]   12 3 ⍕NEF,CN[; 2 1 3]
```

```
........................16/ 4/78
  ∇ VONB;L;AG;X;Y;P;PL;OUTM;DATA
 ER1:'AGES?'
  →(∧/0<AG←,⎕)/SK1
  →ER1,0↑⍞←'AGES MUST BE POSITIVE RE-ENTER '
 SK1:AG←AG[P←⍋AG]
  →(3<+/PL←AG≠1⌽AG)/ER2
  →ER1,0↑⍞←'MUST BE MORE THAN 3 DISTINCT AGES. RE-ENTER '
 ER2:'LENGTH AT AGE?'
  →((⍴AG)=⍴L←,⎕)/OK1
  →ER2,0↑⍞←'ONE LENGTH FOR EACH OF ',(⍕⍴AG),' AGES. RE-ENTER '
 OK1:→(∧/L>0)/OK2
  →ER2,0↑⍞←'LENGTHS MUST BE POSITIVE. RE-ENTER '
 OK2:L←L[P]
  X←((¯1++/PL)⍴1),[1.5] ¯1↓P←PL/L
  Y←1↓P
  X←(⌹(⍉X)+.×X)+.×(⍉X)+.×Y
  P←(X[1]÷1-X[2]),-⍟X[2]
  P←P,((+/AG)÷⍴AG)+(⍟1-((+/L)÷⍴L)÷P[1])÷P[2]
  DATA←L,[1.5] AG
  P←FUNCVONB NONLIN P
  PL←P[1]×1-*-P[2]×AG-P[3]
  OUTM←⍉(5,⍴AG)⍴AG,L,PL,(PL-L),(PL-L)÷L
  ' '
  (62↑(22⍴' '),'VON BERTALANAFFY PARAMETERS'),DAT
  ' '
  'L∘∘ --------- ';P[1]
  'K   --------- ';P[2];'           L=L∘∘(1-EXP(KT∘-KT))'
  'T∘  --------- ';P[3]
  ' '
  'AGE  OBSERVED  PREDICTED  ERROR   REL'
  '       LENGTH    LENGTH           ERROR'
  '---  --------  ---------  -----  -----'
  2 0 11 3 10 3 8 3 7 3 ⍕OUTM
  ' '
  '  SOURCE          SS     DF         MS'
  '  ------          --     --         --'
  X←+/[1] OUTM[; 2 4]*2
  Y←((+/OUTM[;2])*2)÷⍴AG
  'REGRESSION', 10 0 5 0 11 1 ⍕(X[1]-X[2]),3,(X[1]-X[2])÷3
  'RESIDUAL  ', 10 1 5 0 11 2 ⍕X[2],(¯3+⍴AG),X[2]÷¯3+⍴AG
  'TOTAL     ', 10 0 5 0 11 1 ⍕X[1],(⍴AG),X[1]÷⍴AG
  ' '
  'R*2 --------- ';100×(X[1]-X[2]+Y)÷X[1]-Y
```

```
........................15/ 4/78
    ∇ VPA;CATCH;J;I;M;MORT;FI;FC;X;YR;AG;Y;TIT;FT;TMP;EXP;FFM;DIF
[1]    NAM←⎕NL 2
[2]   ER0:⍞←'NAME OF CATCH MATRIX?.....'
[3]    →(∨/NAM∧.=X←(¯1↑⍴NAM)↑(' '≠X)/X←,⍞)/OK1
[4]    →ER0,0↑⍞←'NAMED MATRIX DOES NOT EXIST, RE-ENTER '
[5]   OK1:→(2=⍴⍴CATCH←⍎X)/OK2
[6]    →ER0,0↑⍞←((' '≠X)/X),' IS NOT A MATRIX.'
[7]   OK2:→(∧/0≤,CATCH)/ER1
[8]    →0,0↑⍞←'NEGATIVE VALUES IN CATCH MATRIX.'
[9]   ER1:'FIRST YEAR AND YOUNGEST AGE GROUP?'
[10]   →(∧/(X=⌈X),(0≤2↑X),2=⍴X←,⎕)/OK3
[11]  →ER1:0↑⍞←'MUST BE 2 NON-NEGATIVE INTEGERS, RE-ENTER '
[12]  OK3:YR←(¯1+X[1])+⍳J←(⍴CATCH)[2]
[13]   AG←(¯1+X[2])+⍳I←(⍴CATCH)[1]
[14]  ER2:'NATURAL MORTALITY?'
[15]   →(∨/(1,I)=⍴M←,⎕)/OK4
[16]   →ER2,0↑⍞←'EITHER 1 OR ',(⍕I),' NUMBERS, RE-ENTER '
[17]  OK4:→(∧/M>0)/OK5
[18]   →ER2,0↑⍞←'MORTALITIES MUST BE POSITIVE, RE-ENTER '
[19]  OK5:MORT←⍉((J,I)⍴M)
[20]   F←(I,J)⍴0
[21]  ER3:'STARTING F VALUES FOR LAST YEAR (',(⍕¯1↑YR),')?'
[22]   →(∨/(1,I)=⍴FI←,⎕)/OK6
[23]   →ER3,0↑⍞←'EITHER 1 OR ',(⍕I),' NUMBERS, RE-ENTER '
[24]  OK6:→(∧/FI>0)/OK7
[25]   →ER3,0↑⍞←'MORTALITIES MUST BE POSITIVE, RE-ENTER '
[26]  OK7:F[;J]←I⍴FI
[27]  ER4:'STARTING F VALUES FOR OLDEST AGE GROUP (',(⍕¯1↑AG),')?'
[28]   →(∨/(1,J)=⍴FC←,⎕)/OK8
[29]   →ER4,0↑⍞←'EITHER 1 OR ',(⍕J),' NUMBERS, RE-ENTER '
[30]  OK8:→(∧/FC>0)/OK9
[31]   →ER4,0↑⍞←'MORTALITIES MUST BE POSITIVE, RE-ENTER '
[32]  OK9:F[I;]←J⍴FC
[33]   POP←(I,J)⍴0
[34]   POP[;J]←(CATCH[;J]×FI+MORT[;J])÷FI×1-*-FI+MORT[;J]
[35]   POP[I;]←(CATCH[I;]×FC+MORT[I;])÷FC
[36]  ER5:⍞←'IS FISHING COMPLETE FOR LAST AGE GIVEN(YES OR NO)?.....'
[37]   →((∧/'YES'=3↑X),∨∧/'NO'=2↑X←(' '≠X)/X←,⍞)/SK1,ER5
[38]   POP[I;]←POP[I;]÷1-*-FC+MORT[I;]
[39]  SK1:Y←J-1
[40]  AA:X←MORT[⍳I-1;Y]
[41]   POP[⍳I-1;Y]←(CATCH[⍳I-1;Y]×*X÷2)+(POP[1↓⍳I;Y+1]×*X)
[42]   F[⍳I-1;Y]←FT←(⍟POP[⍳I-1;Y]÷POP[1↓⍳I;Y+1])-X
[43]  BB:TMP←(EXP←*FT+X)×FFM←FT÷FT+X
[44]   →(∧/1E¯5>|DIF←TMP-FFM+CATCH[⍳I-1;Y]÷POP[1↓⍳I;Y+1])/CC
[45]   F[⍳I-1;Y]←FT←FT-DIF÷TMP+X×(EXP-1)÷(FT+X)*2
[46]   →BB
[47]  CC:POP[⍳I-1;Y]←POP[1↓⍳I;Y+1]×*X+FT
[48]   →(1≤Y←Y-1)/AA
[49]   F[⍳I-1;⍳J-1]←(⍟(¯1 ¯1 ↓POP)÷ 1 1 ↓POP)- ¯1 ¯1 ↓MORT
[50]   TIT←'POPULATION NUMBERS'
[51]   0 OUT POP,[1]+/[1] POP
[52]   TIT←'FISHING MORTALITY'
[53]   3 OUT F,[1]((+/[1] POP×F)÷+/[1] POP)
```

```
∇ YIELD;M;OUTM;FMX;I;MAX;SL;ROT;WGT;SEL;FLG;FLG1
ER0:'WEIGHT AT AGE? (KG)'
 →(∧/0<WGT←,⎕)/ER1
 →ER0,0↑⍞←'WEIGHTS MUST BE POSITIVE, RE-ENTER '
ER1:'PARTIAL RECRUITMENT?'
 →((⍴WGT)=⍴SEL←,⎕)/OK1
 →ER1,0↑⍞←'ONE VALUE FOR EACH OF ',(⍕⍴WGT),' AGES, RE-ENTER '
OK1:→(∧/(SEL≥0),SEL≤1)/ER2
 →ER1,0↑⍞←'PARTIAL RECRUITMENTS MUST BE BETWEEN 0 AND 1, RE-ENTER '
ER2:'NATURAL MORTALITY?'
 →(∨/(1,⍴WGT)=⍴M←,⎕)/OK2
 →ER2,0↑⍞←'EITHER 1 OR ',(⍕⍴WGT),' NUMBERS, RE-ENTER '
OK2:→(∧/M>0)/OK3
 →ER2,0↑⍞←'MORTALITIES MUST BE POSITIVE, RE-ENTER '
OK3:M←(⍴WGT)⍴M
 OUTM←(1 3 ⍴0),[1] YIE 0.0001 0.0002
 FMX←¯40+⍳40
AA:OUTM←OUTM,[1] YIE 0.05×FMX←FMX+40
 →(FMX[1]≤200)/BB
 '***** F MAX NOT FOUND *****'
 FLG←0
 OUTM[;3]←OUTM[;3]×1E¯6
 →CC
BB:→((1↑⍴OUTM)<2↓I← ¯1 0 1 +(OUTM[;3]=⌈/OUTM[;3])/⍳1↑⍴OUTM)/AA
 OUTM[;3]←OUTM[;3]×1E¯6
 FMX←LAGR OUTM[I; 1 3]
 MAX←¯0.5×FMX[2]÷FMX[1]
 FMX←+/(MAX* 2 1 0)×FMX
 OUTM←OUTM,[1] MAX,(+/(MAX* 2 1 0)×LAGR OUTM[I; 1 2]),FMX
 FLG←1
CC:SL←0.1×¯3○1↑1↓LAGR OUTM[1 2 3 ; 1 3]
 ROT←((-FLG,0)↓OUTM[; 1 3])+.× 2 2 ⍴(2○SL),(-1○SL),(1○SL),2○SL
 I← ¯1 0 1 +(ROT[;2]=⌈/ROT[;2])/⍳1↑⍴ROT
 FLG1←1
 →(I[3]≤1↑⍴ROT)/DD
 FLG1←0
 '***** F OPT NOT FOUND *****'
 I←I-1
DD:FMX←LAGR ROT[I;]
 MAX←¯0.5×FMX[2]÷FMX[1]
 FMX←+/(MAX* 2 1 0)×FMX
 FMX←(MAX,FMX)+.× 2 2 ⍴(2○SL),(1○SL),(-1○SL),2○SL
 FMX←FMX[1],(+/(FMX[1]* 2 1 0)×LAGR OUTM[I; 1 2]),FMX[2]
 OUTM←(¯1 3 ↑OUTM),[1] FMX
 OUTM[;3]←OUTM[;3]×1000000
 I←⌊10⍟0.1×OUTM[1;1]
 OUTM←OUTM,[1] YIE(⍳20)×0.5⌊(10*I)×⌊OUTM[1;1]×10*-I+1
 OUTM←OUTM[I←⍋OUTM[;1];]
 OUTM←OUTM,[2](OUTM[;3]÷OUTM[;2]),[1.5] OUTM[;3]÷OUTM[;1]
 OUTM[;5]←OUTM[;5]÷⌈/OUTM[;5]
 ((¯8+⎕PW)↑((⌊0.5×¯17+⎕PW)⍴' '),'YIELD PER RECRUIT'),DAT
 ' '
 '          FISHING     CATCH     YIELD    AVG. WEIGHT    YIELD PER'
 '         MORTALITY  (NUMBER)    (KG)        (KG)        UNIT EFFORT'
 '         ---------  --------   -----    -----------    -----------'
 ROT← 22 7 ⍴' '
 ROT[I⍳2;]←'F0.1---'
 ROT[I⍳1;]←'FMAX---'
 FMX←22⍴1
 FMX[I⍳ 2 1]←FLG1,FLG
 FMX/[1] ROT,[2] 7 3 11 5 9 3 10 3 13 3 ⍕OUTM
 ' '
 ((⌊0.5×¯45+⎕PW)↑' '),'YIELD PER RECRUIT(KG) VS. FISHING MORTALITY'
 (15,¯5+⎕PW) PLOT FMX/[1] OUTM[; 1 3]
```

```
........................15/ 4/78
    ∇ X←DAT
     X←1↓ 3 0 ⍕100⊤⌽3↑⎕TS
     X[3 6]←'/'
```

```
........................15/ 4/78
    ∇ Y←GAUSS X;Z;T
     Z←0.5-|0.5-X
     T←(⍟÷Z*2)*0.5
     Y←(2×¯0.5+Z≠X)×T-(2.515517+T×0.802853+T×0.010328)÷1+T×1.432788+T×0.18
     9269+T×0.001308
```

```
........................15/ 4/78
    ∇ Y←LAGR X;TMP
     TMP←X[;1]∘.*(1↑⍴X)-⍳1↑⍴X
     Y←(⌹TMP)+.×X[;2]
```

```
........................15/ 4/78
    ∇ OP←F NONLIN P;ITER;DERIV;SS;OSS;Z;ZP;FACT;MULT;DELTA;X;X2;Y;N
     X←⎕FX F
     ITER←DERIV←0
     SS←+/(Y←DATA[;1]-,P FUNC 0 1 ↓DATA)*2
    TOP:→(1=ITER←ITER+1)/SKP
     →(∧/1E¯6>|(P-OP)÷P)/OUT
    SKP:OP←P
     OSS← 1 2 ⍴0,SS
     DERIV←1
     Z←P FUNC 0 1 ↓DATA
     DERIV←0
     DELTA←(⌹ZP+.×Z)+.×(ZP←⍉Z)+.×Y
     DELTA←DELTA×(⌈/10000⌊|DELTA)÷⌈/|DELTA
     P←OP+(MULT←1)×DELTA
     FACT←2
     →(OSS[1;2]>SS←+/(DATA[;1]-,P FUNC 0 1 ↓DATA)*2)/CORR
     FACT←0.5
    CORR:N←1↑⍴OSS←OSS,[1] MULT,SS
     →(∧/(N≥3),(OSS[N;2]≥OSS[N-1;2]),~∧/,OSS[1;2]< 1 1 ↓OSS)/STOP
     P←OP+(MULT←MULT×FACT)×DELTA
     SS←+/(Y←DATA[;1]-,P FUNC 0 1 ↓DATA)*2
     →CORR
    STOP:X2←(X←, ¯3 1 ↑OSS)*2
     Y←, ¯3 ¯1 ↑OSS
     MULT←(+/Y×(2⌽X2)-1⌽X2)÷2×+/Y×(2⌽X)-1⌽X
     P←OP+MULT×DELTA
     →(OSS[N-1;2]>SS←+/(Y←DATA[;1]-,P FUNC 0 1 ↓DATA)*2)/TOP
     P←OP+OSS[N-1;1]×DELTA
     SS←+/(Y←DATA[;1]-,P FUNC 0 1 ↓DATA)*2
     →TOP
    OUT:OP←P
     X←⎕EX 'FUNC'
```

```
⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄15/ 4/78
    ∇ A OUT B;C;D;W;Y;FW
[1]    A←,⍉(2,¯1↑⍴B)⍴(5⌈1+(A≠0)+A+⌈10⍟1⌈⌈/[1] B),(¯1↑⍴B)⍴A
[2]    FW←(20+⍴TIT)⌈(⎕PW)⌊4++/((⍴A)⍴ 1 0)/A
[3]    Y←YR
[4]    ' '
[5]    ((¯8+FW)↑((⌊0.5×FW-⍴TIT)⍴' '),TIT),DAT
[6]   SK1:' '
[7]    C←¯1+(⎕PW<4++\A[¯1+2×⍳0.5×⍴A])⍳1
[8]    D←(2×C⌊⍴Y)↑A
[9]    D[2×⍳C⌊⍴Y]←0
[10]   '     |',D⍪(C⌊⍴Y)↑Y
[11]   '----+',(+/A[¯1+2×⍳C])⍴'-'
[12]   (2 0 ⍕((⍴AG),1)⍴AG),(((⍴AG),2)⍴' |'),((2×C)↑A)⍕((⍴AG),C)↑B
[13]   ((((1↑⍴B)-⍴AG),4)⍴'    |'),((2×C)↑A)⍕(((⍴AG)-1↑⍴B),C)↑B
[14]   A←(2×C)↓A
[15]   B←(0,C)↓B
[16]   Y←C↓Y
[17]   →(0≠⍴A)/SK1
```

```
⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄15/ 4/78
    ∇ START
[1]    'MAXIMUM PRINTING WIDTH?'
[2]    ⎕PW←⎕
[3]    NEWS
```

```
⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄15/ 4/78
    ∇ Y←P STUD DF;X;X3;X5;X7;X9;G1;G2;G3;G4
[1]    X←GAUSS 1-P
[2]    G1←((X3←X*3)+X)÷4
[3]    G2←((5×X5←X*5)+(16×X3)+3×X)÷96
[4]    G3←((3×X7←X*7)+(19×X5)+(17×X3)-15×X)÷384
[5]    G4←((79×X*9)+(776×X7)+(1482×X5)+(¯1920×X3)-945×X)÷92160
[6]    Y←X+(G1÷DF)+(G2÷DF*2)+(G3÷DF*3)+G4÷DF*4
```

```
⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄⋄15/ 4/78
    ∇ Y←YIE X;NUM;CAT;YLD;I;FT;TMP
[1]    NUM←(⍴X)⍴1
[2]    CAT←YLD←(⍴X)⍴0
[3]    I←1
[4]   AA:FT←X×SEL[I]
[5]    CAT←CAT+TMP←NUM×FT×(1-*-FT+M[I])÷FT+M[I]
[6]    YLD←YLD+TMP×WGT[I]
[7]    NUM←NUM×*-FT+M[I]
[8]    →((⍴WGT)≥I←I+1)/AA
[9]    Y←⍉(3,⍴X)⍴X,CAT,YLD
```

```
.........................16/ 4/78
    ∇ PROJOUT;X;OUTM;RED;A;TIT

ER1:⍞←'FULL OUTPUT OR SUMMARY?(F OR S).....'
 →(('F'=X),'S'≠X←1↑(X≠' ')/X←,⍞)/SK1,ER1
 OUTM←YR,[1] MEAN[1;1+⍴AG;],[1] SDEV[1;1+⍴AG;],[1] WGT+.× ¯1 0 ↓MEAN[1
 ;;]
 OUTM←OUTM,[1](MAT+.× ¯1 0 ↓MEAN[1;;]),[0.5](MAT×WGT)+.× ¯1 0 ↓MEAN[1;
 ;]
 OUTM←OUTM,[1] MEAN[2;1+⍴AG;],[1] SDEV[2;1+⍴AG;],[1](WGT+.× ¯1 0 ↓MEAN
 [2;;]),[0.5]⌈/[1] ¯1 0 ↓MEAN[3;;]
 RED←∧/[1] 0=OUTM←⍉OUTM
 A←+\4,9⍴10
 RED←~∨/[1]((RED/A)∘.≥⍳94)∧(RED/0,¯1↓A)∘.<⍳94
 TIT←'YEAR      POP     STANDARD      POP       MATURE     MATURE     CATCH
   STANDARD    CATCH     MATURE'
 TIT←TIT,[0.5] '          NUMBERS     DEV        BIOMASS   NUMBERS    BIOMASS
    NUMBERS    DEV       BIOMASS      F    '
 RED/((2,⍴RED)↑TIT),[1](⍴RED)⍴'-'
 ' '
 RED/ 4 0 10 0 10 2 10 0 10 0 10 2 10 0 10 0 10 2 10 4 ⍕OUTM
 →0
SK1:TIT←'POPULATION NUMBERS'
 0 OUT MEAN[1;;]
 TIT←'STANDARD DEVIATION OF POPULATION NUMBERS'
 →(∧/0=,SDEV[1;;])/N1
 0 OUT SDEV[1;;]
N1:TIT←'POPULATION BIOMASS'
 OUTM←(¯1 0 ↓MEAN[1;;])×⍉((⍴YR),⍴AG)⍴WGT
 2 OUT OUTM,[1]+/[1] OUTM
 TIT←'MATURE NUMBERS'
 →(∧/0=MAT)/N2
 OUTM←(¯1 0 ↓MEAN[1;;])×⍉((⍴YR),⍴AG)⍴MAT
 0 OUT OUTM,[1]+/[1] OUTM
 TIT←'MATURE BIOMASS'
 OUTM←(¯1 0 ↓MEAN[1;;])×⍉((⍴YR),⍴AG)⍴MAT×WGT
 2 OUT OUTM,[1]+/[1] OUTM
N2:TIT←'CATCH NUMBERS'
 0 OUT MEAN[2;;]
 TIT←'CATCH BIOMASS'
 OUTM←(¯1 0 ↓MEAN[2;;])×⍉((⍴YR),⍴AG)⍴WGT
 0 OUT OUTM,[1]+/[1] OUTM
 TIT←'FISHING MORTALITY'
 3 OUT ¯1 0 ↓MEAN[3;;]
```

```
......................16/ 4/78
  ∇ PROJ;C;C1;C2;CURR;X;Y
   Y←1
   MEAN←SDEV←(3,(1+⍴AG),⍴YR)⍴0
  LP1:CURR←(3,(⍴AG),⍴YR)⍴¯1
   CURR[1;;1]←POP
   CURR[2;;1]←CAT
   X←1
  LP2:→((∨/CURR[2;;X]≠¯1),ANS[X]≥10)/SK1,SK2
   CURR[3;;X]←SEL×ANS[X]
   CURR[2;;X]←(CURR[3;;X]×CURR[1;;X]÷CURR[3;;X]+M)×1-*-CURR[3;;X]+M
   →SK3
  SK1:CURR[3;;X]←SEL
   →(∧/CURR[1;;X]>CURR[2;;X])/LP3
   'CATCH GREATER THAN POPULATION'
   CURR[2;;X]←CURR[2;;X]⌊CURR[1;;X]
  LP3:C←(CURR[3;;X]×CURR[1;;X]÷CURR[3;;X]+M)×1-*-CURR[3;;X]+M
   →(0.01≥⌈/|C-CURR[2;;X])/SK3
   C1←-(CURR[3;;X]×CURR[1;;X]÷CURR[3;;X]+M)×*-CURR[3;;X]+M
   C2←-(CURR[1;;X]×M÷(CURR[3;;X]+M)*2)×1-*-CURR[3;;X]+M
   CURR[3;;X]←CURR[3;;X]-(CURR[2;;X]-C)÷C1+C2
   →LP3
  SK2:CURR[3;;X]←SEL
   →(ANS[X]<+/WGT×CURR[1;;X])/LP4
   'QUOTA GREATER THAN POPULATION BIOMASS'
   CURR[3;;X]←10×SEL
   CURR[2;;X]←(CURR[3;;X]×CURR[1;;X]÷CURR[3;;X]+M)×1-*-CURR[3;;X]+M
   →SK3
  LP4:CURR[2;;X]←(CURR[3;;X]×CURR[1;;X]÷CURR[3;;X]+M)×1-*-CURR[3;;X]+M
   →(0.4>|ANS[X]-+/WGT×CURR[2;;X])/SK3
   CURR[3;;X]←CURR[3;;X]×ANS[X]÷+/WGT×CURR[2;;X]
   →LP4
  SK3:→((⍴YR)<X←X+1)/SK4
   CURR[1;;X]←CURR[1;;X-1]×*-CURR[3;;X-1]+M
   C←*REC[X-1]+SDR[X-1]×GAUSS 1E¯6×?999999
   CURR[1;;X]←C,¯1↓CURR[1;;X]
   →LP2
  SK4:CURR←0⌈CURR,[2]+/[2] 0⌈CURR
   MEAN←MEAN+CURR⌈0
   SDEV←SDEV+CURR*2
   →(REP≥Y←Y+1)/LP1
   MEAN←MEAN÷REP
   →(1=REP)/SK5
   SDEV←(0⌈(SDEV-REP×MEAN*2)÷REP-1)*0.5
   →0
  SK5:SDEV←(⍴MEAN)⍴0
```