NOTES

AN ALGORITHM TO ESTIMATE THE TERMINAL EXPLOITATION IN THE LENGTH COHORT ANALYSIS

ABSTRACT

An algorithm is developed to estimate the terminal rate of exploitation for use in the length cohort analysis. A computer program in BASICA is also presented to execute the algorithm.

THE LENGTH cohort analysis (Jones 1984) is a derived form of age based cohort analysis of Pope (1972). This approach is basically same as the length converted catch curve where the age frequency data is transformed to length frequency data *via* a growth function in length. The assumptions for the validity of length cohort analysis are almost the same as that of the age based analysis except that the growth function is the von Bertalanffy's Growth Formula (vBGF). The aim is to make use of the length structure of the catch to estimate the population structure under certain assumptions and for some known growth and natural mortality parameters. This would facilitate in estimating the stock size, recruitment, spawning stock, fishing mortality etc. which are essential for fish stock assessment.

Assuming that the growth equation in length is vBGF, the basic equation in the length cohort analysis (following the notations of Sparre and Venema (1992) is given by

$$N(L_1) = [N(L_2) * H(L_1, L_2) + C(L_1, L_2)] * H(L_1, L_2) \quad \text{... I}$$

where $N(L_1)$ = number of fish that attain length L_1

$N(L_2)$ = number of fish that attain length L_2

$C(L_1, L_2)$ = number of fish caught in the length group (L_1, L_2) and

$$H(L_1, L_2) = ((L_\infty - L_1)/(L_\infty - L_2))^{(M/2K)}$$

where L_∞, K are the parameters of vBGF and M is the instantaneous rate of natural mortality.

The calculations are started from the last length group in the length frequency data and use the length based catch equation

$$C(L_1, L_2) = N(L_1) (F/Z) (1 - \exp(-Z \Delta t))$$

where F is the fishing mortality rate in the length group (L_1, L_2)

$Z = F + M$ and Δt is time taken to grow from L_1 to L_2 (time interval).

In the last length class corresponding to the catch in numbers larger than L_1 (the lower limit of the last length group) Δt is considered relatively large so that it tends to ∞ and the above equation reduces to

$$C(L_1, \infty) = N(L_1) F/Z$$

from which we get $N(L_1) = C(L_1, \infty) Z/F$

using this and the equation I, the numbers in the sea are recursively computed. Once these are available, the computation of mortality rates, standing stock etc. become straightforward and needs no elaboration and the procedure is given in Sparre and Venema (1992).

Usually for estimating the numbers at sea in the terminal length group, a value for F/Z (terminal exploitation rate) is assumed. According to Jones (1984) the choice of F/Z depends on the extent of exploitation of the stock under study. For moderate to heavily exploited stocks a choice of $F/Z \geq 0.50$ ensures convergence of mortality rates. In this section an algorithm is developed which carries out the length cohort analysis not from the terminal length group but from a chosen length group and such that the (F/Z) values at the larger length groups are more or less homogeneous.

In heavily exploited stocks and also in the short lived species, constrained by the selectivity of the gear, the catches of the length classes at the fully vulnerable length class or in its neighbourhood are likely to be more representative than those that are far away. The following algorithm is based on this consideration only.

Algorithm

For given values of L_∞, K and M

(1) Choose a starting length group from where the calculations are to begin.

(2) Specify the range of F/Z along with increment in F/Z in this range

(3) Start from the smallest value in the above range, since M is given calculate Z

(4) Calculate the numbers in the sea in the chosen length class from

$Z\,C_L/[F\,(1 - \exp(-Z\,\Delta t))]$ where C_L is the numbers caught in the chosen length group and Δt is the time interval for the length group

(5) Back calculate the number in sea using the equation I

(6) Calculate numbers forward in length groups

(7) If the calculated numbers in sea are negative or zero stop the calculations go to step (8) else increment F/Z go to step (4).

(8) Check F/Z values at the larger length groups, if there is more or less concordance, stop the routine and print results, if not, repeat the routine in the neighbourhood of the F/Z obtained in (7)

(9) Check again the F/Z values at the larger length groups, if there is no further improvement possible, stop the calculations and print the results.

The computer program written in *Basica* is given hereunder. After loading the program and giving the run command it will prompt for input file. The input file should contain line by line the following particulars L_∞ K, number of length classes, lower limit of the first length group, class width and catch in numbers one by one.

Program to estimate the terminal F/Z in the length cohort analysis

```
10 CLS
20 KEY OFF
40 DIM N(50, C(50), L(50), DL(50), X1(50),
   X2(50), F(50), Z(20)
50 DIM E(50), MNS(50)
70 INPUT 'FILE NAME:', F$
80 OPEN F$ FOR INPUT AS #1
90 WHILE NOT EOF (1)
100 INPUT #1, L8
110 INPUT #1, K
120 INPUT #1, NL
130 INPUT #1, LMIN
140 INPUT #1, WID
150 FOR I=1 TO NL
160 INPUT #1, C(I)
170 NEXT I
180 WEND
190 CLOSE #1
200 CLS
210 PRINT TAB(30); ''INPUT DATA''
```

```
220 PRINT TAB (3); ‘‘----------’’
230 PRINT: PRINT
240 PRINT TAB(10); ‘‘L8 =’’; L8; TAB (3);
    ‘‘K = ’’; K
250 PRINT : PRINT
260 FOR I = 1 TO NL
270 L(I) LMIN + (I-1)* WID
280 NEXT I
290 FOR I = 1 TO NL
300 L (NL + 1) = L (NL) + WID
310 DL(I) = LOG ((L8 - L(I)) / (L8 - L
    (I + I)))
320 DL(I) = DL (I)/K
330 NEXT I
340 PRINT TAB (10); ‘‘SL. NO’’; TAB (20);
    ‘‘L-LEN’’; TAB (30); ‘‘CATCH’’
350 PRINT
360 FOR I = 1 TO NL
370 PRINT TAB (10); I; TAB (20); L(I); TAB
    (30); C(I)
380 NEXT I
390 PRINT: PRINT TAB (10); ‘‘STARTING
    LENGTH CLASS (SL.NO); ‘‘;:INPUT’’’’, SL
400 IF SL < 1 OR SL > NL THEN 200
410 INPUT ‘‘M/K =’’, MK: M = MK* K
420 FOR 1 = 1 TO NL
430 X1 (I) = EXP (-.5* M* DL (I))
440 X2 (I) = 1/XI(I)
450 NEXT I
460 INPUT ‘‘MIN F/Z = ’’, EMIN
470 INPUT ‘‘MAX F/Z = ’’, EMAX
480 INPUT ‘‘INCR IN F/Z = ’’, EINC
490 FOR J = EMIN TO EMAX STEP EINC
500 MZ = 1 - J
510 Z = M / (1 - J)
520 Z = Z* DL(SL)
530 U = J* (1-EXP (-Z))
540 N(SL) = C(SL) / U
550 GOSUB 930
560 FOR I = SL + 1 TO NL
570 N(I) = X1 (I-1)* (X1 (I-1)*N(I-1)-C(I-1))
580 NEXT I

590 N(NL+1)=X1(NL)*(X1(NL)*N(NL)-C(NL))
600 IF N(NL + 1) < = 0 THEN GOSUB 980
610 CLS
620 IF ESTOP <> J THEN LOCATE 12,20:
    PRINT ‘‘---PLEASE WAIT--’’: GOTO 870
630 PRINT TAB(30); ‘‘RESULTS FOR F/Z = ’’;
    PRINT USING’’ #.###’’; J:PRINT:PRINT
640 PRINT TAB(5); ‘‘M/K =’’;: PRINT USING’’
    #.##’’; MK;
650 PRINT TAB (25); ‘‘S.L. = ’’;: PRINT
    USING ‘####.##’; L(SL)
660 PRINT
670 GOSUB 1050
680 FOR T = 1 TO 65: PRINT TAB
    (4 + T); ‘‘=’’;: NEXT: PRINT
690 PRINT TAB (5);’’ LENGTH’’; TAB (20);’’
    CATCH’’; TAB (35);’’ POPLIN’’; TAB
    (50);" F ‘‘:TAB (6);’’ F/Z’’
700 FOR T = 1 TO 65 : PRINT TAB
    (4 + T); ‘‘=’’;: NEXT: PRINT
710 PRINT
720 FOR I = 1 TO NL
730 E(SL) = J
740 PRINT TAB(5);: PRINT USING ‘‘####.#’’;
    L(I);
750 PRINT TAB (20);: PRINT USING ‘‘######.
    ###’’; C(I);
760 PRINT TAB (35);: PRINT USING ‘‘#######.
    ###’’; N(I);
770 PRINT TAB (50);: PRINT USING ‘##.
    ####’’; F(I);
780 PRINT TABE (60);: PRINT USING ‘‘#.###’’;
    E(I)
790 NEXT I
800 FOR T = 1 TO 65: PRINT TAB
    (4 + T); ‘‘=’’;: NEXT : PRINT
810 A$ = INPUT$(1)
820 PRINT : PRINT ‘DO YOU WANT TO
    TRY NEAR F/Z =’’;:.PRINT USING’’
    #.###’’; J
830 INPUT ‘‘IF YES TYPE ENTER ELSE N
    OR n’’ Y$
840 IF Y$ = ‘‘THEN GOSUB 1150
850 IF Y$ = ‘‘N’’ OR Y$ = ‘‘n’’ THEN END
860 IF J = ESTOP THEN END
870 NEXT J
880 PRINT ‘‘NO PROBLEM UPTO F/Z =’’
    ;:PRINT USING’’ #.####’’; J-EINC: PRINT
890 PRINT ‘‘TRY AGAIN BEYOND F/Z =
    ’’;: PRINT USING’’ #.####’’; J-EINC
900 C$ = INPUT$ (1)
```

```
910 GOTO 200
920 END
930 '
940 FOR I = SL-1 TO 1 STEP-1
950 N(I) = (X2 (I + 1) *N(I + 1) + C(I) *
    X2 (I + 1)
960 NEXT I
970 RETURN
980 '
990 PRINT ‘‘NUMBERS CAUGHT IS MORE
     THAN POPULATION!’’
1000 PRINT
1010 PRINT ‘‘FOR F/Z = ’’ ;:PRINT USING
     ‘‘#.####’; J
1020 A$ = INPUT$(1)
1030 CLS: GOSUB 1150
1040 RETURN
1050 '
1060 FOR I = 1 TO NL-1
1070 E(I) = C(I) / (N(I) - N (I + I))
1080 E(NL) = C(NL)/N(NL)
1090 F(I) = M*E(I)/(1-E(I))
1100 Z(I)=F(I)+M
1110 NEXT I
1120 F(NL) = M* E(NL)/(1-E(NL))
1130 Z(NL) = F(NL) + M
1140 RETURN
1150 '
1160 CLS
1170 ESTOP = J-EINC
1180 DELX = ABS (E (NL)- ESTART)
1190 'PRINT DELX:D$ = INPUT$ (1)
1200 IF DELX < = .0001 THEN J = ESTOP:
     GOTO 630
1210 IF DELX > .0001 THEN 1220
1220 EMIN = J
1230 EMAX = J + EINC
1240 EINC = .0001
1250 ESTART = E(NL)
1260 GOTO 490
1270 RETURN
```

Central Marine Fisheries Research Institute, Cochin M. SRINATH

REFERENCES

JONES, R 1984. *FAO Fish. Tech. Pap.* 256: 118p.

POPE, J.G. 1972. *Res. Bull. ICNAF* 9 : 65-74.

SPARRE, P AND S.C. VENEMA 1992. *FAO Fish. Tech. Pap.* No. 306. 1. Rev. 1. Rome. FAO. 376 p.