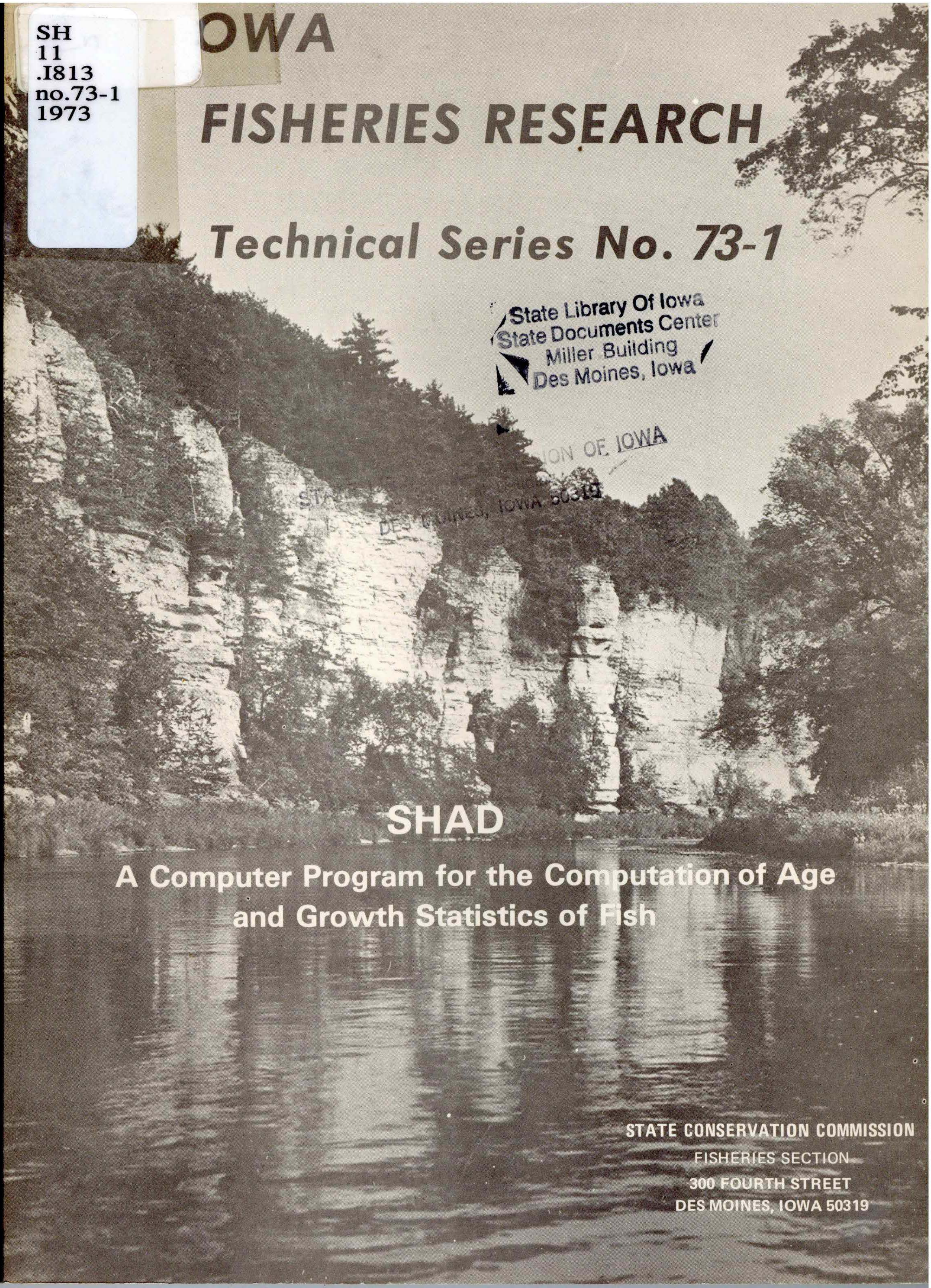
IOWA FISHERIES RESEARCH

Technical Series No. 73-1



SHAD

A Computer Program for the Computation of Age and Growth Statistics of Fish

STATE CONSERVATION COMMISSION
FISHERIES SECTION
300 FOURTH STREET
DES MOINES, IOWA 50319

SHAD

A COMPUTER PROGRAM FOR COMPUTATION OF AGE AND GROWTH STATISTICS OF FISH

By

James Mayhew
Fisheries Research Supervisor

Technical Series 73-1

Fisheries Section

Jerry M. Conley
Superintendent of Fisheries

TABLE OF CONTENTS

INTRODUCTION

The original version of SHAD was written in 1970 to aid fisheries biologists in this department with tedious computations of mathematical equations associated with age and growth studies of fishes. Age and growth statistics are essential parts of life history studies of fishes, and are particularly useful for evaluation of population surveys. Fish grow throughout their life span so growth is a primary function of age. Stress in fish is quickly reflected in growth and body condition. The basic question asked by biologists is, "How long does it require for a fish to reach a certain size, either in terms of body length or weight?" Reliable answering of this question forms the basis for evaluating many biological parameters of a fish population. More important it defines growth characteristics, age structure and the relationship between age and body measurements.

Mathematical computation of age and growth statistics are both tiresome and error prone. Without mechanical or electronic computing devices the solution of most equations are virtually impossible without gross error. High speed electronic computers are far superior to all other, significantly reducing both time and cost. Several age and growth programs have been developed, but most handled only portions of complete studies (Gerking, 1965; Voigtlander and Roochvarg, 1967; Dahlberg, 1966; Chadwick, 1966; Swingle, 1964). Others would require extensive modification before they would function in a computer system available to this department.

SHAD is a complete age and growth program and contains routines for length-weight regression, body condition factors, body-scale regression, length-frequency distribution and back calculation of body length at the end of each year of life. The program provides routines for single and multiple regression analysis, analysis of variance, plotting and obtaining frequency distribution. The main routine controls the type and sequence of data input and serves as the supervisor for extrapolating all analyses, plots, frequency counts, transformations and means.

The original program was written by the author and Kenneth Merritt, Statistical Numerical Analysis Section, Statistical Laboratory, Iowa State University. Revisions were made by the author and William Kennedy of the ISU laboratory. SHAD has been placed in disk storage in the IBM 360-65 system at ISU and can be executed from location by STM2.ME(SHAD). FORTRAN IV, LEVEL G language is used for the program. Present core requirements for execution are 128K bytes. The greatest change from the original version of SHAD was made for computation of body length at the end of each year of life by Dahl-Lea direct proportion method. For implementation of English and metric values several adjustments were required for the input and output formats. Program changes are also required for statements 0032, 0033, 0102 and 0107 when converting English values to metric values (see Appendix A). Both programs utilize the same main supervision routine and four subroutines for regressions analysis, plotting, plot border and plot labelling.

PROGRAM RESTRICTIONS

Limitations of problematic parameters for present version of SHAD are the following:

1.	Maximum number of fish in sample	400
2.	Maximum number of variables (NAGE + 3)	12
3.	Maximum number of length intervals	30

These limitations may be increased by enlarging arrays and requiring additional core.

INPUT REQUIREMENTS

Data are entered for punching on standard IBM coding forms or similar column paper paper. The listings of input data are made in the following sequence.

Column 1 = serial number

Column 2 = age of fish

Column 3 = total length (TL) in .1 inches or 1 mm

Column 4 = weight in .01 lbs or 1 gm

Column 5 = scale radius in .1 inches or 1 mm

Column 6 =

.

.

.

n = radius in .1 inches or 1 mm at NAGE

Any column can be used so long as the format card reflects the number of the columns used. No particular order is required for data cards, although it is often easier to read the input record if ages are listed in ascending order.

Identification at the start of the program is limited to 80 columns per line including spaces, letters and other characters. As many lines as needed may be used to identify the data. An example of proper identification is as follows:

Line 1 - AGE AND GROWTH OF WHITE CRAPPIE - RED HAW - 1972

Line 2 - DATA WERE COLLECTED FROM SEINE HAUL JUNE AND JULY

Line 3 - DATA ARE FOR COMBINED SEXES

Line 4 - COLLECTOR WAS SQUIBB AND AGED BY PUTNAM

One of several jobs can be executed during each computer run. For each job the following header cards must accompany the job deck in the order indicated.

Card 1: Title card(s)

Column	Meaning
1-80	Identification (may also be blank)

Card 2: Control card for reading data

1-5	NOBS number of observations input
6-10	NAGE maximum age of fish
11-80	(blank)

Card 3: Format card for data input

1-80 Card must contain I-format for the first two elements and F-format for NAGE + 3 variables

Card 4: Data cards

1-80 The total number of variables input from each data card is three plus the maximum age (NAGE + 3). These variables follow the serial number and age which is input using I-format conversion codes. The order for these variables should be punched and should follow the example under INPUT REQUIREMENTS.

PROGRAM OUTPUT

The title (Card 1) and data (Card 4) are printed for documentation and identification of the problem.

LENGTH FREQUENCY AND CONDITION FACTORS

Class intervals for obtaining frequency distribution of TL is either .5 or 1.0 inches for the English version or 15 or 25 mm for the metric version. If any fish in the sample exceeds 20 inches or 500 mm the frequency interval is set at the higher value, otherwise the interval will be .5 inches or 15 mm. This arrangement was necessary to eliminate the very small number of groups resulting from species of fish not attaining great body length, such as crappie, perch and bluegill. The body condition factor, C or K and means for TL, weight and scale radius are printed for each size interval. Default values for means having zero frequency counts is zero.

Length-frequency distribution, mean TL, mean wgt, mean scale radius and C or K factors are printed under the heading format

NO.	GROUPS	FREQ.	LENGTH	WEIGHT	SCALE RADIUS	C-K FACTOR

The condition factor C and K for the i^{th} class interval is a truncated integer

$$C(K) = (1/L_i)^3(W_i) \times 10^4$$

for zero frequency the condition factor is not computed.

Class intervals for the length-frequency distribution are listed as truncated integers, where interval ranges are written in exponential notation. The value 6.0 falls within the 5.5-6.0 class interval, but would be written as

5.50001E00 - 6.00000E00

and followed by intervals

6.00001E00 - 6.50000E00

6.50001E00 - 7.00000E00.

PLOTTING OF LENGTH-WEIGHT AND BODY-SCALE REGRESSIONS

For each pair of variables plotted, maximum and minimum values are obtained and used to determine the range for the ordinate and abscissa. This insures enough space will be allowed to include all points while utilizing maximum scaling. The dependent variable is always plotted on the vertical axis against the independent variable on the horizontal axis. Default values in the grouped body-scale regression are entered as 0,0 in both raw data and predicted plots. All intervals on the plot are printed in exponential notation.

A list of plots included in SHAD are as follows:

	Independent variable (X)		Dependent variable (Y)
1.	Raw body length	vs	Raw body weight
2.	Log_{10} body length		Log_{10} body weight
3.	Grouped mean scale radius		Grouped mean body length
4.	Grouped mean scale radius		Predicted linear grouped mean body length
5.	Grouped mean scale radius		Predicted quadratic grouped mean body length

LENGTH-WEIGHT AND BODY-SCALE REGRESSION ANALYSIS

Output from the regression algorithm of m independent (X) variables and one dependent (Y) variable was expressed as

N	Regression coefficient	T	S.D.
1	$\hat{\beta}_0$	t_0	S_{β_1}
2	$\hat{\beta}_1$	t_1	S_{β_2}
.	.	.	.
	.	.	.
m + 1	$\hat{\beta}_m$	t_m	S_{β_m}

where i estimated the i^{th} coefficient in the linear model

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_m X_{im} + \varepsilon_i, \; i = 1, 2, \cdots, n,$$

$S^2\beta_j$ was the estimated variance of β_j and t_i was the test statistic for testing the null hypothesis $\beta_j = 0$.

Regression analysis, means and the multiple coefficient of determination (r^2) output was computed and expressed as

Analysis of variance

Source of variation	df	SS	MS	F
Regress	m-1	$\sum_{i-1}^{m} \beta_i \sum_{j=1}^{n} X_{ij}Y_j$	R	R/S^2
Error	n-m	Subtraction	S^2	
Total	n-1	$\sum_{i=1}^{n} (Y_i-\overline{Y})^2$		

Multiple $r^2 = R(m-1)/\sum_{i=1}^{n}(Y_i-\overline{Y})^2$

X means (independent variable)

1	$\overline{X}_1$
2	$\overline{X}_2$
1	.
.	.
.	.
m	$\overline{X}_m$

Y means (dependent variable)

1	$\overline{Y}$

Predicted values ($\hat{Y}_i$) of the dependent variable, Y_i, was extrapolated from the computed regression and had the generalized form

$$Y_i = \hat{\beta}_0 + \hat{\beta}_1 X_{i1} + \cdots + \hat{\beta}_m X_{im},$$

where the i^{th} residual was the difference $Y_i - \hat{Y}_i$.

LENGTH-WEIGHT RELATIONSHIP

The length-weight relationship was expressed as the logarithmic transformed regression equation

$$\log_{10} W_i = \gamma_0 + \gamma_1 \log_{10} L_i + \varepsilon_i,$$

where W_i is the observed body weight, L_i is the observed body length and ε_i is the residual.

This analysis can be extended to include a test of allometric growth. Estimated variance and the test statistic (T) for the length-weight regression are the same as before except the null hypothesis is tested for $\beta_j = 3$.

Comparison of length-weight regressions for two or more samples can be tested by two procedures taken directly from the printout. Confidence interval for the 95% level can be set around the regression coefficients by the usual fashion using a t distribution. For the slope coefficient, b, the procedure is

$$b - t_{.05}\, S_b \leq \beta \leq b + t_{.05}\, S_b$$

where n - 2 is required for degrees of freedom in a two-tailed distribution. If these intervals overlap there is little reason to believe slope of the two regression differ significantly. Analysis of covariance in slope and elevation can be tested in a t distribution by

$$t = \frac{b_1 - b_2}{\sqrt{V_{(1)} + V_{(2)}}}$$

However, this method would be restricted to two samples. Procedures for analysis of covariance of regression coefficients for more than two samples can be found in Snedecor and Cochran (1967:419). Extension of the analysis of variance will greatly enhance the meaning of length-weight relationship and lead to a superior knowledge of fish population characteristics.

BODY-SCALE REGRESSION

Faced with a choice of linear or non-linear regression for body-scale relationships, particularly for catfishes the regressions are computed in three different forms. The first is a simple linear regression equation for grouped data in the model

$$L_i = \beta_0 + \beta_1 S_i + \varepsilon_i,$$

where L_i is the mean body length, S_i is the mean scale radius and ε_i is the residual. Second, a quadratic model is computed in the form

$$L_i = \alpha_0 + \alpha S_i + \alpha S_i^{\ 2} + \varepsilon_i,$$

where L and S are the same as before. Last, both linear and quadratic models with the intercept through the origin is computed from the models

$$L_i = \beta_0 S_i + \varepsilon_i \text{ and } L_i = \alpha_0 S_i + \alpha S_i^{\ 2} + \varepsilon_i$$

where L and S are again identical with previous forms.

There are several procedures for comparing the fit of linear and quadratic equations. The simplest is to compare T statistics in the printout, which would infer a difference existed, but not the significance of the difference. If it is obvious, which is rare, the comparison need not be carried further. In many instances a linear fit is satisfactory, but the accuracy of back calculations of length can be greatly improved by using the quadratic equation. Comparisons are made by using the analysis of variance table computed and printed for each model. Deviations sums of squares accounted for by the simple linear equation for the term of one degree is printed in the AOV table. The difference between this quantity and the deviations sums of squares for the second order equation are directly comparable with the error term from the latter equation. As an example:

Source of variation	df	SS	MS	F
Second order equation	2	(from quadratic equation table)		
First order equation	1	(from linear equation table)		
Difference	1	(subtraction)		
Error from second order equation	m	(from quadratic equation table)		

The value of F at 1 and m df will determine the level of significants. Most use of the linear model with the zero intercept would be made when intercepts for the simple linear model were negative values.

BACK CALCULATION OF LENGTH AT EACH ANNULUS

The final output is tables containing the grouping of fish by age and estimating body length at each annulus by measurements of annulus distance from the focus of the scale. For each NAGE group, mean scale radius for annuli 1, 2, ..., NAGE are computed. The age frequency is printed along with the mean scale radius under the following format.

GROWTH COMPUTATION GROUPING BY AGE AND MEAN RADIUS

Age	No.	1	2	3	NAGE
1	f_{11}	$\overline{X}_{11}$			
2	f_{21}	$\overline{X}_{21}$	$\overline{X}_{22}$		
3	f_{31}	$\overline{X}_{31}$	$\overline{X}_{32}$	$\overline{X}_{33}$	
.	.	.	.	.	
.	.	.	.	.	
NAGE	f_{NAGE_1}	$\overline{X}_{NAGE_1}$	$\overline{X}_{NAGE_2}$	$\overline{X}_{NAGE_3}$	$\overline{X}_{NAGE,NAGE}$,

where

$$\overline{X}_{ij} = \sum_{k=1}^{f_{ij}} X_{ijk}/f_{ij},$$

and f_{ij} is the frequency of the i^{th} age at annulus j.

When the body-scale relationship is linear with an intercept at the origin, scale growth is directly proportional to body growth and the estimated body length is computed as

$$L_{ij} = \frac{\overline{S}_{ij}}{\overline{S}_i} L_j \qquad i,j,\ldots, \text{NAGE}.$$

For fish where the body-scale relationship is linear but not necessarily directly proportional the estimated body length is computed as

$$L_{ij} - \hat{\beta}_0 = \frac{\overline{S}_{ij}}{\overline{S}_i} (L_j - \overline{S}_j) \qquad i,j,\ldots, \text{NAGE}.$$

In case of a non-linear body-scale relationship estimated body length is extrapolated from the quadratic function

$$L_{ij} = \hat{\alpha}_0 + \hat{\alpha}_1 \overline{S}_{ij} + \alpha_2 \overline{S}_{ij}^{\,2} \qquad i,j,\ldots, \text{NAGE}.$$

The $\overline{S}_{ij}$ in all equations was the mean scale radius at age i, annulus j.

Increments for the estimates are computed from

$$\hat{L}_{i,j+1} - \hat{L}_{i,j} \qquad i,j,\ldots, \text{NAGE}$$

with

$$\hat{L}_{i,1} = \hat{L}_{i,1} \qquad i,j,\ldots, \text{NAGE}$$

Estimated body length at each annulus can also be computed by successive summation of the grand averaged increments.

LITERATURE CITED

Chadwick, H.
1966. Age and growth program, UNIVAC 1107, FORTRAN IV. Trans. Am. Fish. Soc., Vol. 95, No. 2, 229-30.

Dahlberg, M.
1966. Length-frequency analysis, IBM 7090/7094, FORTRAN II or FORTRAN IV. Trans. Am. Fish. Soc., Vol. 95, No. 3, 331.

Gerking, S.
1965. Two computer programs for age and growth studies. Prog. Fish. Cult. 27:59-66.

Snedecor, G. and W. Cochran
1967. Statistical methods. 6th Ed. Iowa State University Press, Ames, Iowa. 593 pp.

Swingle, W.
1964. Length-weight relationships 1 and 2, IBM 1620, FORTRAN/FORMAT. Trans. Am. Fish. Soc. 93:318-319.

Voigtlander, C. and A. Roochvarg
1967. Age and growth program, control data 3600, FORTRAN 63-FORTRAN IV. Trans. Am. Fish. Soc., Vol. 96, No. 3, 364-66.

APPENDIX

Table 1

Source listing of main routines and sub-routines for SHAD using format for English values.

```
0001          CCMMON  A(400),U(400),N, L, DET, X3(400),Y3(400),B(10)
0002          CCMMON XXY(400,4)   ,YY(400)
0003          DIMENSION IDENT(400,2),D1(400,14)
0004          DIMENSION  FMT(20), F(99), Z(99),SUM(99,20),ICNT(99,20),PD1(25,20)
0005          DIMENSION  PD2(25,20),PD3(25,20),PD4(25,20),IPX(99),BX(10,3)
0006          DIMENSION X33(100),Y33(100),BB(10)
0007      100 CONTINUE
0008          READ(1,2,END=101) FMT
0009          WRITE(3,115) FMT
0010      115 FORMAT('1',20X,20A4,//)
0011          READ(1,1)    NOBS,         NAGE
0012        1 FORMAT(10I5)
0013          NVAR=NAGE+3
0014          N=NOBS
0015          N2 = 2
0016          READ(1,2)      FMT
0017        2 FORMAT(20A4)
        C
        C       READ  RAW  DATA INTO D1(I,J)  UNDER  FMT
0018          M12  = 12
0019          NVAR1= NVAR+1
0020          DO 3 I =1,NOBS
0021          READ(1,FMT)(IDENT(I,J),J=1,N2),(  D1(I,J),J=1, NVAR )
0022          WRITE(3,116)  (IDENT(I,J),J=1,N2),(D1(I,J),J=1,NVAR)
0023      116 FORMAT(' ',I5,I3,2X,10(10F11.4,/11X))
0024        3 CONTINUE
0025          DO 4  I = 1, NOBS
0026          X3(I)= D1( I,1)
0027        4 Y3(I)= D1( I,2)
        C
        C       PART 1:  LENGTH-WEIGHT REGRESSION USING RAW DATA
        C          PLOT LENGTH(X),VS. WEIGHT(Y)
        C
0028          ICODE = 1
0029          WRITE(3,400)
0030      400 FORMAT('1',20X,'LENGTH-WEIGHT PLOT OF RAW DATA',
             1 '          DEPENDENT VARIABLE: WEIGHT',/)
0031          CALL PLOT(ICODE,XMAX,XMIN,YMAX, YMIN)
0032          XINT =.5
0033          IF( XMAX.GT.20) XINT= 1.
0034          ICT = 0
0035          IST = XMIN
0036          M10  = 10
0037          M25 = 99
0038          Z(1)=  IST
0039          DO 18 I = 2, M25
0040          II= I- 1
0041          F(II) = 0
```

```
      Z(I)  = Z(II)  + XINT
      ICT = ICT+ 1
      IF( Z(I).GT. XMAX)  GO TO 19
   18 CONTINUE
   19 CONTINUE
      I =ICT
      IC =ICT+1
      II=ICT+2
      I1=ICT+3
      Z(IC )=Z(I)  +XINT
      Z(II)=Z(IC )+XINT
      Z(I1)=Z(II)+XINT
C
C        THERE ARE  ICT INTEVERALS OF LENGTH  XINT EACH (.5 OR 1.0 )
C
      DO 33 I = 1,M25
      DO 33 J = 1,M10
      SUM(I,J)= 0
   33 ICNT(I,J)=0
      M3 =  3
      M4 =  4
      M1=1
      IC=ICT+1
      DO 20 II= 1, NOBS
      DO 200 I= 1, IC
      I1=I-1
C
      IF( D1(II,1).GT.Z(I))  GO TO 200
      DO 201 J= 1, M3
      SUM(I1,J)= SUM(I1,J)+ D1(II, J)
      ICNT(I1,J)=ICNT(I1,J)  + 1
  201 CONTINUE
      GO TO 20
  200 CONTINUE
   20 CONTINUE
      DO  82 I = 1,ICT
      DO  82 J = 1,M3
      IF(ICNT(I,J).EQ.0)  GO TO 82
      SUM(I,J)  = SUM(I,J)  / ICNT(I,J)
   82 CONTINUE
      WRITE(3,21)
   21 FORMAT('1',30X,'LENGTH-FREQUENCY DISTRIBUTION',// )
      KK=0
      DO 353 I=1,ICT
      IF(ICNT(I,1).NE.0)  GO TO 354
  353 KK=KK+1
  354 CONTINUE
      DO 359 I=1,NOBS
```

```
  359 A(I)=0
      K1=KK
      DO 355 I=1,ICT
      KK=KK+1
      Z(I)=Z(KK)
      ICNT(I,1)=ICNT(KK,1)
      DO 356 J=1,M3
  356 SUM(I,J)=SUM(KK,J)
  355 CONTINUE
      DO 350 I=1,ICT
      IPX(I)=10000000
      IF(SUM(I,1).EQ.0) GO TO 350
      A(I)=(1/SUM(I,1)**3    ) * SUM(I,2)*100000
      IPX(I)=A(I)
  350 CONTINUE
      WRITE(3,500)
  500 FORMAT('0', 21X,'GROUPS',16X,'FREQ.',7X,'LENGTH',9X,'WEIGHT',6X,
     1 'SCALE-RADIUS',5X,'C-FACTOR',/)
      ICT=ICT-K1
      DO 23 I= 1, ICT
      II = I+ 1
      WRITE(3,22)  I,  Z(I),Z(II), ICNT(I,1),(SUM(I,J),J=1,M3),IPX(I)
   22 FORMAT(' ', I5,10X, F8.1,'-',F8.1, 10X,  I5,3F15.4,5X,I5)
   23 CONTINUE
C
C        LENGTH WEIGHT REGRESSION      Y = A * X** N
C        X = RAW  LENGTH , Y = RAW WEIGHT
C        TRANSFORM TO  LOG Y =  A + B*LOG X + E
C
      DO  9 I = 1, NOBS
      XXY(I,2) = ALOG10(D1(I,1) )
    9 XXY(I,3) = ALOG10(D1(I,2) )
      N = NOBS
      M = NOBS
      NX= 1
      LD= 0
      LIX= 0
      LCX= 0
      LRX= 0
      LCXY=0
      LPD= 1
      ICODE= 0
      K =  0
      CALL  REGRES ( M,NX,LD,LRX,LCX,LIX,LPD,LCXY,JOBS)
      JOBS=0
C
      DO 24  I= 1, NOBS
   24 Y3(I)  = YY(I)
```

```
            ICODE = 2
            WRITE(3,401)
        401 FORMAT('1',20X,' PREDICTED LENGTH-WEIGTH PLOT OF RAW DATA',
           1 10X,'DEPENDENT VARIABLE: LOG WEIGHT',/)
            CALL  PLOT(ICODE,XMAX,XMIN, YMAX,YMIN )
      C
      C        BEGIN  SECOND  REGRESSION:  BODY - SCALE
      C
         31 FORMAT(' ', 2I5,' ICNT(I,J) WAS ZERO' )
            DO  30 I = 1,ICT
            X3(I)= SUM(I,3)
            Y3(I)= SUM(I,1)
            X33(I)=X3(I)
            Y33(I)=Y3(I)
            M =  ICT
            XXY(I,2)  =X3(I)
         30 XXY(I,3)  =Y3(I)
            ISWTCH=1
        994 CONTINUE
            IF(ISWTCH.EQ.1) GO TO 996
            WRITE(3,991)
        991 FORMAT('1',3X,'LINEAR AND QUADRATIC ESTIMATION BASED ON',/)
            WRITE(3,990)
        990 FORMAT(' ',3X,'FITTING THE BODY-SCALE REGRESSION MODEL',/)
            WRITE(3,989)
        989 FORMAT(' ',3X,'CENTERED TO THE ORIGIN',///)
            SUM1=0.0
            SUM2=0.0
            DO 999 I=1,M
            SUM1=SUM1+X33(I)*X33(I)
        999 SUM2=SUM2+X33(I)*Y33(I)
            D=SUM2/SUM1
            BB(1)=0.
            BB(2)=D
            WRITE(3,998) D
        998 FORMAT('0',3X,'NEW REGRESSION COEFF.',2X,F8.4,//)
            GO TO 995
        996 CONTINUE
            CALL  REGRES ( M,NX,LD,LRX,LCX,LIX,LPD,LCXY,JOBS)
            N =  ICT
            ICODE = 1
            WRITE(3,402)
        402 FORMAT('1',20X,'LINEAR BODY-SCALE PLOT',
           1'OF GROUPED MEANS',10X,'DEPENDENT VARIABLE: BODY LENGTH',/)
            CALL PLOT(ICODE, XMAX,XMIN,YMAX,YMIN )
            DO 93  I= 1,ICT
         93 Y3(I)  =  YY(I)
            ICODE = 2
```

```
          WRITE(3,403)
      403 FORMAT('1',20X,'LINEAR PREDICTED BODY-SCALE PLOT',
         1'OF GROUPED MEANS',10X,'DEPENDENT VARIABLE: BODY LENGTH',/)
          CALL PLOT(ICODE,XMAX,XMIN,YMAX, YMIN )
          M2 = 2
          ICODE = 1
          DO 94  I= 1,ICT
       94 Y3(I) = YY(I)
C         BEGIN PART 3: BODY-SCALE BY POLYNOMIAL FIT
C
          DO 126 I=1,M2
      126 BX(I,1) = B(I)
C
          DC 40 I = 1,ICT
          XXY(I,4) = SUM(I,1)
          XXY(I,3) = SUM(I,3) * SUM(I,3)
       40 XXY(I,2) = SUM(I,3)
C
          NX = 2
          M  = ICT
C
          CALL  REGRES ( M,NX,LD,LRX,LCX,LIX,LPD,LCXY,JOBS)
          ICODE = 2
          DO 41 I = 1,ICT
       41 Y3(I)= YY(I)
          ICODE = 2
          WRITE(3,404)
      404 FORMAT('1',20X,'QUADRATIC PREDICTED BODY-SCALE PLOT',
         1'OF GROUPED MEANS',10X,'DEPENDENT VARIABLE: BODY LENGTH',/)
          CALL PLOT(ICODE,XMAX,XMIN,YMAX, YMIN )
C
          M3  = 3
          M10 =10
          M25 =25
          DO 102 I =1, M3
      102 BX(I,2) = B(I)
C
C         BEGIN PART 4:  GROWTH  COMPUTATION
C                        GROUPING BY AGE MEAN RADIUS AT EACH ANNULUS
          DO  50 I= 1, 25
          DO  50 J= 1, 20
          SUM(I,J) = 0
       50 ICNT(I,J)= 0
          NVAR2=NVAR-3
          DO 51 J = 1, NVAR2
          DC 51 I = 1, NOBS
          K= J+3
          II = IDENT(I,2)
```

```
      IF( D1(I,K).EQ.0) GO TO  51
      SUM(II,J) = SUM(II,J) +   D1(I,K)
      ICNT(II,J) = ICNT(II,J) + 1
   51 CONTINUE
      M12 =12
      DO 53 I =1,NAGE
      DO 53 J=1,NAGE
C
      IF( ICNT(I,J).EQ.0) GO TO 53
      SUM(I,J) =  SUM(I,J) / ICNT(I,J)
   53 CONTINUE
      IF(ISWTCH.EQ.1) GO TO 992
  995 BX(1,1)=BB(1)
      BX(2,1)=BB(2)
  992 CONTINUE
      WRITE(3,98)
   98 FORMAT('1',20X,'LINEAR GROWTH COMPUTATION GROUPING BY AGE',1X,
     1'MEAN RADIUS',/)
      WRITE(3,412)  (I,I=1,NAGE)
  412 FORMAT(' ' ,2X,'AGE',2X,'NO.',7X,12(5X,I5)  ,/)
      DO  54  I = 1, NAGE
      WRITE(3,56)  I, ICNT(I,1),  (SUM(I,J),J=1,I  )
   56 FORMAT(' ',   I5,    I5,10X, 12F10.4 )
   54 CONTINUE
      WRITE(3,58)
   58 FORMAT('1',10X, 'ESTIMATED BODY LENGTH AND INCREMENTS AT ',
     1 'EACH ANNULUS GROWTH',///)
      WRITE(3,59)
   59 FORMAT('0',20X,'LINEAR  ESTIMATED BODY LENGTH AT ANNULUS',//)
C
C         FOR ESTIMATED TOTAL BODY LENGTH AT NTH ANNULUS USE  BEST  FIT
C         FROM  BODY LENGTH - SCALE REGRESSION
C         CALCULATE AND  PRINT  BOTH LINEAR  AND QUADRATIC .
C
      DO 60 I = 1, NAGE
      DO 60 J = 1, NAGE
      PD1(I,J) = 0
      PD2(I,J) = 0
      PD3(I,J) = 0
   60 PD4(I,J) = 0
      NX1 = NX +1
      DO 61 I = 1, NAGE
      DO 61 J = 1, I
      PD1(I,J)  =BX(1,1)+ SUM(I,J)* BX(2,1)
   61 PD2(I,J)  =BX(1,2)+ SUM(I,J)*BX(2,2)+SUM(I,J)*SUM(I,J)* BX(3,2)
      NC1 = NAGE - 1
      WRITE(3,412)  (I,I=1,NAGE)
      DO 62  I= 1,NAGE
```

```
C
   62 WRITE(3,56) I , ICNT(I,1)            , ( PD1(I,J),J=1,I)
C
      DO 63 I = 1, NAGE
      PD3(I,1) =  PD1(I,1)
      PD4(I,1)  = PD2(I,1)
      DO 63 J = 1, I
      JJ = J +1
      PD4(I,JJ)=PD2(I,JJ)-PD2(I,J)
      PD3(I,JJ) =PD1(I,JJ)- PD1(I,J)
      IF(PD4(I,JJ).LT.0)  PD4(I,JJ)= 0
   63 IF(PD3(I,JJ).LT.0)  PD3(I,JJ)= 0
      DO 64 I= 1,NAGE
   64 Z(I)= 0
      KK = NAGE+1
      DO 65 J = 1,NAGE
      DO 65 I = 1,NAGE
   65 Z(J)  = Z(J)  + PD1(I,J)
C
      DO 66 I= 1, NAGE
      KK = KK-1
   66 Z(I)  = Z(I)  / KK
      WRITE(3,67)      (Z(I),I=1,NAGE )
   67 FORMAT('0', 3X,'GRAND AVE. LENGTH', 10F10.4 )
      WRITE(3,68)
   68 FORMAT('0',//  ,30X,'INCREMENTS',//)
      WRITE(3,412)  (I,I=1,NAGE)
      DO  69 I = 1, NAGE
   69 WRITE(3,56)  I, ICNT(I,1),           (PD3(I,J),J=1,I)
      DO 105 I=1,NAGE
  105 Z(I)=0
      DO 107 J=1,NAGE
      DO 107 I=1,NAGE
  107 Z(J)=Z(J)+PD3(I,J)
      KK=NAGE+1
      DO 112 I=1,NAGE
      KK=KK-1
  112 Z(I)=Z(I)/KK
      WRITE(3,67)      (Z(I),I=1,NAGE )
      WRITE(3,70)
   70 FORMAT('1',20X,'QUADRATIC ESTIMATED BODY LENGTH AT ANNULUS',// )
C
      WRITE(3,412)  (I,I=1,NAGE)
      DO 71 I = 1, NAGE
   71 WRITE(3,56)  I, ICNT(I,1)           , (PD2(I,J),J=1,I)
      DO 72 I= 1,NAGE
   72 Z(I)= 0
      DO 73 J = 1,NAGE
```

```
      DO 73 I = 1, NAGE
   73 Z(J)  = Z(J)+ PD2(I,J)
      KK = NAGE+ 1
      DO 74 I = 1, NAGE
      KK =KK-1
   74 Z(I)=  Z(I)/ KK
      WRITE(3,67)  ( Z(I),I=1,NAGE)
      WRITE(3,75)
   75 FORMAT('0',//,30X, 'INCREMENTS',// )
      WRITE(3,412)  (I,I=1,NAGE)
      DO 76 I = 1,NAGE
   76 WRITE(3,56) I, ICNT(I,1),            (PD4(I,J),J=1,I)
      DO 106 I=1,NAGE
  106 Z(I)=0
      DO 108 J=1,NAGE
      DO 108 I=1,NAGE
  108 Z(J)=Z(J)+PD4(I,J)
      KK=NAGE+1
      DO 111 I=1,NAGE
      KK=KK-1
  111 Z(I)=Z(I)/KK
      WRITE(3,67)      (Z(I),I=1,NAGE )
      IF(ISWTCH.NE.1) GO TO 993
      ISWTCH=2
      GO TO 994
  993 CONTINUE
      GO TO 100
  101 CONTINUE
      STOP
      END
```

```
*OPTIONS IN EFFECT*  NOID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
*OPTIONS IN EFFECT*  NAME = MAIN    , LINECNT =       50
*STATISTICS*    SOURCE STATEMENTS =       320,PROGRAM SIZE =     61612
*STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
      SUBROUTINE    REGRES ( M,NX,LD,LRX,LCX,LIX,LPD,LCXY,JOBS)
      COMMON  A(400),U(400),N, L, DET, X3(400),Y3(400),B(10)
      COMMON XXY(400,4)   ,YY(400)
      DIMENSION               XP(10,10), X(10)        ,R(10,10)           06020
      DIMENSION XC(10,10), T(10), SED(10)  ,V(10)
      DIMENSION    XX(10,10),                    RY(400)
      DIMENSION ISORT(20),NUM(10,3)
      DIMENSION FMT(20)
      JOBS = 0
      K=0
  100 CONTINUE
      N1 = NX + 1                                                     0602019C
      N2 = NX + 2                                                     0602019C
      N3=N2+K
C                                                                     0602019C
C     M = NO.OBS; NX = NO.OF X'S; LD=1 IF DATA PRINTED;               0602019C
C     LRX= 1  IF RAW S.S. PRINTED ; LCX= 1 IF  CORRECTED S.S. PRINTED; 0602019C
C     LIX= 1  IF  INVERSE PRINTED ; LPD = 1  IF PREDICTED Y'S PRINTED. 0602019C
C     LCXY= 1 IF  CORRELATION COMPUTED AND PRINTED; JOBS = NO.JOBS.   0602019C
C                                                                     0602019C
C                                                                     0602019C
C         READ  DATA  INTO   XXY(I,J)                                 0602019C
      KCNT=0
      N1=N1+K
      N2=N3
      NX=NX+K
      N= NX                                                           0602019C
      L= N*(N+1)/2
C                                                                     0602019C
      IF(LD-1) 98,97,98                                               0602019C
   97 WRITE(3,65)
   65 FORMAT ('1 DATA',//)
      DO 96  I= 1, M
   96 WRITE(3,95)  (XXY(I,J), J= 2, N2)                               0602019C
C                                                                     0602019C
   95 FORMAT(' ',8E15.6)
C                                                                     0602019C
   98 DO 94 I = 1,M                                                   0602019C
      XXY(I,1) = 1                                                    0602019C
   94 CONTINUE                                                        0602019C
C                                                                     0602019C
C          ZERO OUT  X'X ARRAY                                        0602019C
C                                                                     0602019C
      DO 93  I= 1,N2                                                  0602019C
      DO 93  J= 1,N2                                                  0602019C
      XP(I,J)  = 0.                                                   0602019C
   93 CONTINUE                                                        0602019C
C                                                                     0602019C
```

```
            C       COMPUTE  RAW  SUM SQUARES                                        0602019C
            C                                                                        0602019C
0034              DO 3  K = 1, M                                                     0602019C
0035              DO 92 II= 1, N2                                                    0602019C
0036           92 X(II) = XXY(K,II)                                                  0602019C
0037              DO 3  I = 1, N2                                                    0602019C
0038              DO 3  J = 1, N2                                                    0602019C
            C                                                                        0602019C
0039              XP(I,J) = XP(I,J) + X(I)* X(J)                                     0602019C
0040            3 CONTINUE                                                           0602019C
            C                                                                        0602019C
0041              IF(LRX- 1)  91,90,91                                               0602019C
            C                                                                        0602019C
0042           90 WRITE(3,69)
0043           69 FORMAT ('1 RAW SUM OF SQUARES',//)
0044              DO 87   I = 1, N2
0045           87 WRITE(3,89)  ( XP(I,J),J = 1,I )                                   0602019C
            C                                                                        0602019C
0046           89 FORMAT ('                          ',//, 10( 6(F20.7,1X),/))
            C                                                                        0602019C
            C       APPLY  JORDAN  REDUCTION  TO GET CORRECTED  SUM SQUARES          0602019C
            C                                                                        0602019C
0047           91 DO 4  J = 1, N2                                                    0602019C
0048            4 XP(1,J) = XP(1,J)/ M
            C                                                                        0602019C
            C                                                                        0602019C
0049              DO 5  I = 1 , N1                                                   0602019C
0050              II =    I + 1                                                      0602019C
0051              DX = XP(II,1)
            C                                                                        0602019C
0052              DO 5  J = 1 ,N1                                                    0602019C
0053              JJ =  J + 1                                                        0602019C
0054              XC(I,J) =  XP(II,JJ) - XP(1,JJ) * DX                               0602019C
0055            5 CONTINUE                                                           0602019C
            C                                                                        0602019C
            C        OUTPUT  OF  CORRECTED  SUM SQUARES                              0602019C
            C                                                                        0602019C
0056              IF(LCX- 1) 85,86,85                                                0602019C
0057           86 WRITE(3,54)
0058           54 FORMAT('1 CORRECTED SUM OF SQUARES',//)
0059              DO 83  I =  1, N1
0060           83 WRITE(3,84)  ( XC(I,J), J= 1, I )                                  0602019C
0061           84 FORMAT('                          ',//,10(6(F20.7,1X),/))
            C                                                                        0602019C
            C       COMPUTE  AND  OUTPUT  SIMPLE CORRELATION COEFFICIENTS            0602019C
            C                                                                        0602019C
0062           85 CONTINUE
0063              IF(LCXY- 1) 81,82,81                                               0602019C
```

```
0064        82 DO 6  I = 1, N1                                                     0602019C
0065           DO 6  J = 1, N1                                                     0602019C
0066           R(I,J) =  XC(I,J) / SQRT(XC(I,I)* XC(J,J))                          0602019C
0067         6 CONTINUE                                                            0602019C
0068           WRITE(3,59)
0069        59 FORMAT('1 SIMPLE CORRELATION COEFFICIENTS',//)
0070           DO 80 I = 1, N1                                                     0602019C
0071        80 WRITE(3,79)  ( R(I,J) , J= 1,I)                                     0602019C
0072        79 FORMAT('                                  ',//,10(6(F20.7,1X),/))  0602019C
0073        81 K= 0                                                                0602019C
0074           DO 7  I = 1, NX                                                     0602019C
0075           DO 7  J = 1, I                                                      0602019C
0076           K= K+1                                                              0602019C
0077           A(K) = XC(I,J)                                                      0602019C
0078         7 CONTINUE                                                            0602019C
0079           CALL BORDER                                                         0602019C
      C                                                                            0602019C
0080           K= 0                                                                0602019C
0081           DO 8  I = 1, NX                                                     0602019C
0082           DO 8  J = 1, I                                                      0602019C
0083           K= K+1                                                              0602019C
0084           XX(I,J) = A(K)                                                      0602019C
0085         8 CONTINUE                                                            0602019C
      C                                                                            0602019C
0086           DO 9  I = 1,NX                                                      0602019C
0087           DO 9  J = I,NX                                                      0602019C
0088           XX(I,J) =  XX(J,I)                                                  0602019C
0089         9 CONTINUE                                                            0602019C
0090           IF(LIX- 1) 77,78,77                                                 0602019C
      C                                                                            0602019C
      C          OUTPUT  OF  X'X  INVERSE                                          0602019C
      C                                                                            0602019C
0091        78 WRITE(3,55)
0092        55 FORMAT('1 OUTPUT OF X''X INVERSE', //)
0093           DO 76  I  =  1, NX
0094        76 WRITE(3,75)  ( XX(I,J),J = 1, I  )                                  0602019C
0095        75 FORMAT('                  ', //, 10(6(F20.7,1X),/))                 0602019C
0096        77 CONTINUE                                                            0602019C
      C                                                                            0602019C
      C          COMPUTE   B'S  OF   X'XB =  X'Y                                   0602019C
      C                                                                            0602019C
0097           DO 10  I = 1, N1                                                    0602019C
0098           B(I)= 0.                                                            0602019C
0099        10 CONTINUE
      C                                                                            0602019C
0100           DO 11  I = 1, NX                                                    0602019C
0101           II =  I  + 1                                                        0602019C
0102           DO 11  J = 1, NX                                                    0602019C
```

```
0103          B(II) = B(II) + XX(I,J) * XC( J,N1)
0104       11 CONTINUE                                                    0602019C
        C                                                                 0602019C
        C      COMPUTE   B(1)                                             0602019C
        C                                                                 0602019C
0105          SUM= 0.                                                     0602019C
        C                                                                 0602019C
0106          DO 12  J = 2,N1                                             0602019C
0107       12 SUM = SUM  + XP(1,J)* B(J)                                  0602019C
        C                                                                 0602019C
0108          B(1) = XP(1,N2)- SUM
        C                                                                 0602019C
        C     COMPUTE  VAR OF B(1) STROE IN  VB1                          0602019C
        C                                                                 0602019C
0109          DO 13  I= 1, NX                                             0602019C
0110       13 V(I) = 0.                                                   0602019C
0111          VB1  = 0.                                                   0602019C
0112          DO 14  I = 1, NX                                            0602019C
0113          DO 14  J = 1, NX                                            0602019C
0114          JJ = J+ 1                                                   0602019C
0115          V(I) = V(I)  + XP(1,JJ)* XX(I,J)
0116       14 CONTINUE
        C                                                                 0602019C
0117          DO 15  I = 1, NX                                            0602019C
0118          II = I + 1                                                  0602019C
0119       15 VB1 = VB1  + V(I)*XP(1,II)                                  0602019C
        C                                                                 0602019C
        C      COMPUTE   B'X'Y                                            0602019C
        C                                                                 0602019C
0120          BXY = 0                                                     0602019C
0121          DO 16 J = 1, NX                                             0602019C
0122          JJ = J+1
0123       16 BXY = BXY + B(JJ)*XC(J,N1)
0124          N=M
        C                                                                 0602019C
        C      COMPUTE   ERROR SS =  Y'Y - B'X'Y =  ERR  AND ERR M.S. AND VB1  0602019C
        C                                                                 0602019C
0125          ERR =  XC(N1,N1) - BXY
0126          IFERR  = N- NX- 1
0127          ERRMS =  ERR/ IFERR
0128          VB1 = (1./N+ VB1)* ERRMS
        C                                                                 0602019C
        C      COMPUTE  STANDARD ERROR  OF  B'S, = SEB                    0602019C
        C                                                                 0602019C
0129          DO 17  I =  1, NX                                           0602019C
0130          II = I + 1                                                  0602019C
0131       17 SED(II) = SQRT(XX(I,I)* ERRMS)                              0602019C
0132          SED(1) = SQRT(VB1)
```

```
              C                                                                   0602019C
              C        COMPUTE   T- VALUES   = T                                  0602019C
              C                                                                   0602019C
              C                                                                   0602019C
0133                 DO 18  I = 1, N1
0134                 II=I
0135              18 T(II)=  B(II)/ SED(II)
              C                                                                   0602019C
              C        COMPUTE  MULTIPLE  R = RX                                  0602019C
              C                                                                   0602019C
0136                 RX =  ERR / XC(N1,N1)                                        0602019C
0137                 RX=1-RX
0138                 SBXY = BXY /NX                                               0602019C
0139                 IFTOT = N-1
0140                 F1 = SBXY / ERRMS                                            0602019C
              C                                                                   0602019C
              C       OUT PUT  B'S,  T'S, AND  S.D.                               0602019C
              C                                                                   0602019C
0141                 WRITE(3,50)
0142              50 FORMAT('1',12X,'REGRESSION COEFF.',16X,'T',17X,'S.D.',//)
0143                 DO 21 I = 1, N1                                              0602019C
0144              21 WRITE(3,20) I,B(I), T(I), SED(I)
0145              20 FORMAT(' ',5X,I5,3F20.4)
              C                                                                   0602019C
              C       COMPUTE ANALYSIS  OF  VARIANCE  AND  OUTPUT                 0602019C
              C                                                                   0602019C
0146                 WRITE(3,38)   RX                                             0602019C
0147              38 FORMAT('0MULTIPLE R2=', F20.7,//)                            0602019C
0148                 WRITE(3,108)       (I,XP(1,I),I=2,N1)
0149             108 FORMAT('0',10X,'X MEANS (INDEPENDENT)',//,25(I5,F20.4,/))
0150                 WRITE(3,109)      XP(1,N2)
0151             109 FORMAT('0',10X,'Y MEAN (DEPENDENT)',//,5X,F20.4)
0152                 WRITE(3,22)
0153              22 FORMAT('1 ANALYSIS OF VARIANCE  TABLE',// )
0154                 WRITE(3,23)
0155              23 FORMAT('0 SOURCE',14X,'D.F',17X,'SUM SQ.',14X,'MEAN SQ',13X,'F',/) 0602019C
0156                 WRITE (3,24)   NX, BXY , SBXY, F1
0157              24 FORMAT('0 REGRESS',13X,I5,07X,3(F20.8))
0158                 WRITE (3,25)  IFERR, ERR,  ERRMS                             0602019C
0159              25 FORMAT('0ERROR',15X,I5,07X, 2(F20.8))
0160                 WRITE (3,26)  IFTOT,  XC(N1,N1)
0161              26 FORMAT ('0TOTAL',15X,I5, 07X,F20.7)
              C                                                                   0602019C
              C        COMPUTE  AND OUTPUT  OBS, PREDICTED, AND RESIDUAL          0602019C
              C                                                                   0602019C
0162                 IF(LPD - 1) 28,27,28                                         0602019C
0163              27 DO  29 I = 1,M                                               0602019C
0164                 YY(I) = 0.                                                   0602019C
```

```
            C                                                                          0602019C
            C        COMPUTE   T- VALUES   = T                                         0602019C
            C                                                                          0602019C
            C                                                                          0602019C
0133                DO 18  I = 1, N1
0134                II=I
0135             18 T(II)=  B(II)/ SED(II)
            C                                                                          0602019C
            C        COMPUTE  MULTIPLE  R = RX                                         0602019C
            C                                                                          0602019C
0136                RX =  ERR / XC(N1,N1)                                              0602019C
0137                RX=1-RX
0138                SBXY = BXY /NX                                                     0602019C
0139                IFTOT = N-1
0140                F1 = SBXY / ERRMS                                                  0602019C
            C                                                                          0602019C
            C       OUT PUT  B'S,  T'S, AND  S.D.                                      0602019C
            C                                                                          0602019C
0141                WRITE(3,50)
0142             50 FORMAT('1',12X,'REGRESSION COEFF.',16X,'T',17X,'S.D.',//)
0143                DO 21 I = 1, N1                                                    0602019C
0144             21 WRITE(3,20) I,B(I), T(I), SED(I)
0145             20 FORMAT(' ',5X,I5,3F20.4)
            C                                                                          0602019C
            C       COMPUTE ANALYSIS  OF  VARIANCE  AND  OUTPUT                        0602019C
            C                                                                          0602019C
0146                WRITE(3,38)  RX                                                    0602019C
0147             38 FORMAT('OMULTIPLE R2=', F20.7,//)                                  0602019C
0148                WRITE(3,108)      (I,XP(1,I),I=2,N1)
0149            108 FORMAT('0',10X,'X MEANS (INDEPENDENT)',//,25(I5,F20.4,/))
0150                WRITE(3,109)     XP(1,N2)
0151            109 FORMAT('0',10X,'Y MEAN  (DEPENDENT)',//,5X,F20.4)
0152                WRITE(3,22)
0153             22 FORMAT('1 ANALYSIS OF VARIANCE  TABLE',// )
0154                WRITE(3,23)
0155             23 FORMAT('0 SOURCE',14X,'D.F',17X,'SUM SQ.',14X,'MEAN SQ',13X,'F',/) 0602019C
0156                WRITE (3,24)  NX, BXY , SBXY, F1
0157             24 FORMAT('0 REGRESS',13X,I5,07X,3(F20.8))
0158                WRITE (3,25)  IFERR, ERR,  ERRMS                                   0602019C
0159             25 FORMAT('OERROR',15X,I5,07X, 2(F20.8))
0160                WRITE (3,26)  IFTOT,  XC(N1,N1)
0161             26 FORMAT ('OTOTAL',15X,I5, 07X,F20.7)
            C                                                                          0602019C
            C        COMPUTE  AND OUTPUT  OBS, PREDICTED, AND RESIDVAL                 0602019C
            C                                                                          0602019C
0162                IF(LPD - 1) 28,27,28                                               0602019C
0163             27 DC  29 I = 1,M                                                     0602019C
0164                YY(I) = 0.                                                         0602019C
```

```
                                                                          0602019C
0165        29 CCNTINUE                                                   0602019C
0166           DO 30  I = 1,M                                             0602019C
0167           DC 30  J = 1,N1                                            0602019C
0168           YY(I) = YY(I)+ XXY(I,J) * B(J)                             0602019C
0169        30 CONTINUE                                                   0602019C
       C         COMPUTE   RESIDVALS                                      0602019C
       C
0170           IF(JOBS.EQ.1) GO TO 101                                    0602019C
0171           DC 31  I = 1,M                                             0602019C
0172           RY(I) = XXY(I,N2) - YY(I)                                  0602019C
0173        31 CCNTINUE
0174           WRITE(3,43)
0175        43 FCRMAT('1',16X,'OBSERVED',12X,'PREDICTED',11X,'RESIDUAL',//)   0602019C
0176           DC 32 I = 1, M
0177        32 WRITE(3,105) I, XXY(I,N2), YY(I), RY(I)
0178           GC TO 28
0179       101 CONTINUE
0180           WRITE(3,102)
0181       102 FORMAT('1',16X,'LOG OBS', 12X,'LOG PREDICT',10X,'OBSERVED',11X,
              1 'PREDICTED',10X,'RESIDUAL',/)
0182           DO 103 I=1,M
0183           A(I)=10**XXY(I,N2)
0184           U(I)=10**YY(I)
0185       103 RY(I)=A(I)-U(I)
0186           DO 104 I=1,M
0187           WRITE(3,105) I,XXY(I,N2),YY(I),A(I),U(I),RY(I)
0188       105 FORMAT(' ',I5,5F20.4)
0189       104 CONTINUE
0190        28 CONTINUE
0191           RETURN                                                     0602019C
0192           END
```

```
*OPTIONS IN EFFECT*  NOID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
*OPTIONS IN EFFECT*  NAME = REGRES  , LINECNT =       50
*STATISTICS*    SOURCE STATEMENTS =      192,PROGRAM SIZE =     9880
*STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
      SUBROUTINE  BORDER                                              0602019B
      COMMON  A(400),U(400),N, L, DET, X3(400),Y3(400),B(10)
      CCMMON XXY(400,4)   ,YY(400)
C                                                                     0602019B
      DET = 1.                                                        0602019B
      KKK = 1
C                                                                     0602019B
      IF (A(1))   3,75,3
    3 A(1)    =   1.0/ A(1)
      IF(N.LT.2) GO TO 11
      DO 22 K = 2,N                                                   0602019B
      KK = K-1                                                        0602019B
      KKK = KKK +KK
C                                                                     0602019B
C        STORE KTH COL OF  A  TEMP  IN U .                            0602019B
C                                                                     0602019B
      DO 4  I= 1,KK                                                   0602019B
      J = KKK+ I - 1                                                  0602019B
    4 U(I)  = A(J)                                                    0602019B
C                                                                     0602019B
C        ZERO  OUT KTH  COL  OF A                                     0602019B
C                                                                     0602019B
      DO 5  I= 1,KK                                                   0602019B
      J= KKK +I - 1                                                   0602019B
    5 A(J) = 0.                                                       0602019B
C                                                                     0602019B
C        COMPUTE  (INVERSE  A(K-1))* U(K); STORE IN  A                0602019B
C                                                                     0602019B
      DO 6  I= 1,KK                                                   0602019B
      L1=KKK + I - 1
      LL= I*(I-1)/2 + 1
      LL1= I*(I+1)/2
      LP = LL -1
      NCX =0
      DO 6  J = 1,KK                                                  0602019B
      NCX = NCX +1
      LP = LP + 1
      IF(LP - LL1) 50,50,51
   50 NZ = LP
      GO TO 60
   51 NZ = NZ + NCX -1
   60 A(L1)= A(L1) + A(NZ)* U(J)
    6 CCNTINUE                                                        0602019B
C                                                                     0602019B
C        COMPUTE  U'(K)* INV A(K-1)* U(K)  ; STORE IN ALPHA           0602019B
C                                                                     0602019B
      ALPHA = 0.                                                      0602019B
      DO 7  I =  1,KK                                                 0602019B
```

```
0035             J =  KKK + I - 1                                               0602019B
0036           7 ALPHA = ALPHA + U(I)* A(J)                                     0602019B
           C                                                                    0602019B
           C        COMPUTE   ALPHA                                             0602019B
           C                                                                    0602019B
0037             JJ = KKK + K - 1                                               0602019B
0038             ALPHA =  A(JJ) - ALPHA                                         0602019B
           C                                                                    0602019B
0039             IF(ALPHA) 8,75,8                                               0602019B
           C                                                                    0602019B
           C        COMPUTE  DETERMINANT  OF  A                                 0602019B
           C                                                                    0602019B
0040           8 DET = DET * ALPHA                                              0602019B
           C                                                                    0602019B
           C        COMPUTE:    INV A(K-1) + INV  A(K-1)*U(K)* U(K)'* INV A(K-1)   0602019B
           C                                                                    0602019B
0041             DO 41 I= 1,KK
0042             J = KKK+ I - 1
0043          41 U(I)  = A(J)
           C                                                                    0602019B
0044             DO 55 I = 1,KK
0045             LL = I*(I+1)/2
0046             NX = I
           C                                                                    0602019B
0047             DO 55 J = NX, KK
0048             A(LL)  = A(LL)  + U(J)  *U(I)  / ALPHA
0049             LL = LL + J
0050          55 CONTINUE
           C                                                                    0602019B
           C        COMPUTE:  - INV A(K-1)* U(K)  / ALPHA                       0602019B
           C                                                                    0602019B
0051             DO  9  I = 1,KK                                                0602019B
0052             J =  KKK + I - 1                                               0602019B
0053           9 A(J)  =  -A(J)/ ALPHA                                          0602019B
           C                                                                    0602019B
           C        COMPUTE :  1/ALPHA ; STORE IN  A                            0602019B
           C                                                                    0602019B
0054             JJ = KKK + KK                                                  0602019B
0055             A(JJ)  =  1.0 / ALPHA                                          0602019B
           C                                                                    0602019B
0056          22 CONTINUE                                                       0602019B
0057             GO TO 11                                                       0602019B
0058          75 WRITE (3,10)  ALPHA                                            0602019B
0059          10 FORMAT ('1NO SOLUTION BECAUSE OF SINGULAR MATRIX',10X,F10.8)   0602019B
0060             STOP                                                           0602019B
0061          11 RETURN                                                         0602019B
0062             END                                                            0602019B
```

```
*OPTIONS IN EFFECT*  NOID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
*OPTIONS IN EFFECT*  NAME = BORDER   , LINECNT =         50
*STATISTICS*    SOURCE STATEMENTS =          62,PROGRAM SIZE =       1594
*STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
      SUBROUTINE PLOT(ICODE,XMAX,XMIN,YMAX,YMIN)
      CCMMON  A(400),U(400),N, L, DET, X3(400),Y3(400),B(10)
      COMMON XXY(400,4)   ,YY(400)
      DIMENSION X(400),Y(400),IT(400)
      XMIN=XMIN
      XMAX=XMAX
      YMIN=YMIN
      YMAX=YMAX
      DC 40 I=1,N
      X(I)=X3(I)
   40 Y(I)=Y3(I)
      DO 25 I=1,N
   25 IT(I)=11
C
      GC TO  (41,42,43),ICODE
C
C        FIND  XMIN, XMAX ,YMIN,AND  YMAX
   41 CONTINUE
      XMIN = 100000.
      XMAX =-100000.
C
      DC 10 I=1,N
      IF(X(I).GE.XMIN)  GO TO 13
      XMIN = X(I)
   42 CCNTINUE
   13 IF(XMAX.GE.X(I))  GO TO 10
      XMAX = X(I)
   14 CONTINUE
   10 CCNTINUE
      YMIN = 100000.
      YMAX =-100000.
C
      DC 17  I = 1, N
C
      IF(Y(I).GE.YMIN)  GO TO 15
   12 YMIN = Y(I)
   15 IF(YMAX.GE.Y(I))  GO TO 17
   16 YMAX = Y(I)
C
   17 CCNTINUE
   43 CONTINUE
C
      CALL PLOTLF( N, X,Y, IT, XMIN, XMAX, YMIN, YMAX )
      RETURN
      END
```

```
*OPTIONS IN EFFECT*  NOID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
*OPTIONS IN EFFECT*  NAME = PLOT    , LINECNT =        50
*STATISTICS*    SOURCE STATEMENTS =        37,PROGRAM SIZE =      5786
*STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
0001        SUBROUTINE PLOTLF(N,X,Y,IT,XMIN,XMAX,YMIN,YMAX)
      C  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      C
      C THIS VERSION OMITS ANY POINTS OUTSIDE THE SPECIFIED MAX AND MIN,
      C BUT TELLS YOU HOW MANY WERE OMITTED.
      C
      C BY C. MESSINA,S.PEAVY,AND B.JOINER      NATIONAL BUREAU OF STANDARDS
      C LAST UPDATED 1/30/67
      C
      C     ORIGINAL DATA IS PRESERVED  (THIS ROUTINE SEARCHES
      C     INSTEAD OF SORTING)
      C     DOES NOT CALL NEW PAGE
      C
      C  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      C
0002        DIMENSION X(1),Y(1),IT(1),PRINT(101),XP(6)
0003        DIMENSION ITOTAL(101),TABLE(40)
0004        INTEGER PRINT,TABLE
0005        DATA   TABLE/ '1','2','3','4','5','6','7','8','9','0','*','+',
           1   '.',' ','A','B','C','D','E','F','G','H','I','J','K','L','M',
           2   'N','O','P','Q','R','S','T','U','V','W','X','Y','Z'/
      C
      C     IO = NUMBER OF PRINT TAPE
0006        IO=3
      C
0007        WRITE (IO,100)
0008    100 FORMAT(14X,101H+---------+---------+---------+---------+---------+
           1---------+---------+---------+---------+---------+ )
0009        YDELTA=(YMAX-YMIN)/50.
0010        XDELTA=(XMAX-XMIN)/100.
0011        YL=YMAX-YDELTA/2.
0012        YT=YMAX+YDELTA/2.
0013        YLOW=YMIN-YDELTA/2.
0014        XL=XMIN-XDELTA/2.
0015        XHIGH=XMAX+XDELTA/2.
0016        IOUT=0
0017        DO 110 I=1,N
0018        IF (Y(I)-YT) 101,101,109
0019    101 IF (Y(I)-YLOW) 109,109,102
0020    102 IF (X(I)-XL) 109,103,103
0021    103 IF (X(I)-XHIGH) 110,109,109
0022    109 IOUT=IOUT+1
0023    110 CONTINUE
0024        DO 350 I=1,6
0025        L=1
0026        DO 350 J=1,10
0027        DO 200 K=1,101
0028        ITOTAL(K)=1
```

```
  200 PRINT(K)=TABLE(14)
      IFLAG=0
      DO 260 K=1,N
      IF (Y(K)-YT) 205,205,260
  205 IF (Y(K)-YL) 260,260,210
  210 XL=XMIN-XDELTA/2.
      XT=XMIN+XDELTA/2.
      DO 255 KA=1,101
      IF (X(K)-XL) 250,215,215
  215 IF (X(K)-XT) 220,250,250
  220 IF (PRINT(KA)-TABLE(14)) 240,230,240
  230 ITA=IT(K)
      PRINT(KA)=TABLE(ITA)
      GC TO 260
 240  ITOTAL(KA)=ITOTAL(KA)+1
      IFLAG=1
      GC TO 260
  250 XL=XT
  255 XT=XT+XDELTA
  260 CONTINUE
      YT=YL
      YL=YL-YDELTA
      IF (IFLAG) 265,278,265
 265  DO 275 LA=1,101
      IF (ITOTAL(LA)-1) 268,275,268
 268  KK=ITOTAL(LA)
      IF (KK-9) 272,272,270
 270  KK=9
 272  PRINT(LA)=TABLE(KK)
 275  CCNTINUE
 278  CONTINUE
      GO TO (280,300),L
  280 IF (I-5) 285,285,400
  285 L=2
      YP=YT+YDELTA/2.
      WRITE (IO,290) YP,(PRINT(IXZ),IXZ=1,101)
  290 FCRMAT(1X,E12.4,1H+,101A1,1H+)
      GC TO 350
  300 WRITE (IO,310) (PRINT(IXZ),IXZ=1,101)
  310 FCRMAT (13X,1H-,101A1,1H-)
  350 CONTINUE
  400 WRITE (IO,290) YMIN,(PRINT(IXZ),IXZ=1,101)
      WRITE (IO,100)
      XP(1)=XMIN
      XP(6)=XMAX
      XR=20.*XDELTA
      DO 410 I=2,5
  410 XP(I)=XP(I-1)+XR
```

```
0077           WRITE (IO,420) (XP(IXZ),IXZ=1,6)
0078       420 FORMAT(6(7X,E13.5))
0079           IF (IOUT)600,600,500
0080       500 WRITE (IO,550) IOUT
0081       550 FORMAT (/20X,9H**NOTE.  I4,60H POINTS FELL OUTSIDE THE SPECIFIED L
              1IMITS AND WERE OMITTED.  )
0082       600 CONTINUE
0083           RETURN
0084           END
```