USERS GUIDE TO GROWBOOT

By: Charles Burkey Jr.

Regional Information Report[1] No. 5J88-06

Alaska Department of Fish and Game
Division of Commercial Fisheries
P.O. Box 3-2000
Juneau, Alaska 99802

August 1988

---

[1]The Regional Information Report Series was established in 1987 to provide an information access system for all unpublished divisional reports. These reports frequently serve diverse ad hoc informational purposes or archive basic uninterpreted data. To accomodate needs needs for up-to-date information, reports in this series may contain preliminary data.

AUTHOR

Charles E. Burkey Jr. is a Fisheries Biologist I for the Alaska Department of Fish and Game, Division of Commercial Fisheries, P. O. Box 3-2000, Juneau, AK 99802-2000.

ACKNOWLEDGMENTS

Fritz Funk (ADF&G) and Terry Quinn (University of Alaska, Southeast) made suggestions and provided guidance for the construction of this program. Thanks to Bob Wilbur, Phil Rigby, and June Grant for their editorial support and review.

TABLE OF CONTENTS

LIST OF APPENDICES

INTRODUCTION

GROWBOOT is a FORTRAN program which estimates parameters and their confidence intervals for ten common growth models using nonlinear least squares regression. Statistical confidence intervals for the growth curve parameter estimates are derived using a resampling technique called bootstrapping (Efron 1982). GROWBOOT was compiled using the Microsoft(TM) FORTRAN Optimizing Compiler, Version 4.1. It runs on a personal computer under MS-DOS with at least 410K bytes memory free (executable file is approx. 307K). It uses IMSL[2] subroutines and therefore must be linked to IMSL subroutine libraries STATCORE.LIB, STAT1.LIB, and STAT2.LIB.

METHODS

Description

The program uses age-length-weight (AWL) measurements from fish to fit growth curves using either weighted or unweighted nonlinear least squares (NLLS) regression (see IMSL documentation for RNLIN). Statistical confidence intervals for the growth curve parameter estimates are derived using the bootstrap resampling technique (Efron 1982). To begin the bootstrap process, nonlinear least squares procedures are used to fit the specified growth model to the input data set of N observations, producing N expected lengths or weights at age, and N residual deviations of the original lengths or weights at age from the expected lengths or weights at age. For each bootstrap iteration, a bootstrap sample of N residuals is drawn at random with replacement from the initial set of residuals, and added to the initial set of expected lengths or weights at age to form a bootstrap sample of lengths or weights at age. Nonlinear least squares procedures are then used to estimate parameters for the specified growth model on the bootstrap sample of lengths or weights at age. This process is repeated for a user specified number of iterations (1,000 are recommended by Efron, 1982), forming a sample of estimates for each parameter. The frequency distribution of the parameter estimates is then used to construct 95% and 90% confidence intervals by determining the values of the parameter between which 95% and 90% of the parameters estimates fall. Confidence intervals constructed in this manner may be slightly biased (Efron and Tibshirani 1986, Efron 1987), but the method does not depend on assumptions regarding underlying distributions of the parameters.

There are three data file formats available in the program: (1) free format with two variables, age and length or weight, (2) user specified format with four variables, age, length or weight, regression weight, and frequency, and (3) free format with four variables, age, length or weight, regression weight, and frequency. Variable age must be an integer value, while length or weight, regression weight and frequency are real data types.

---

[2]IMSL Stat/PC-Library Edition 1.0 from IMSL, 2500 ParkWest Tower One, 2500 CityWest Boulevard, Houston, Texas 77042-3020 USA.

The user specifies whether a length or weight based model is to be fit. The program requires a minimum of two values per record: age and length or weight. Units of measurement for length and weight are determined by the user. In cases where fish age is not an integer value, ages can be transformed (coded) into integer values (e.g. 0.5 years = 1, 1.5 years = 2, etc.) If coded ages are input into a model that estimates theoretical age at which length or weight equals zero (paremeter t0), the value for t0 must be back transformed using the reciprocal of the age transformation value.

Data file formats 2 and 3 allow the user to input a weighting value and frequency for each record. A weighted regression is appropriate in cases where the variance of the error term in the model being fitted is not constant. A frequency value (e.g. number of fish in an age class) could be used when only mean length or weight at age data is available. Regression weitght and frequency are set to 1.0 by default if no values are input for these variables. Data file format 2 (user specified format) requires values for all four variables to be input. If there are no data for regression weight and frequency the user must designate two blank fields for these variables in the format statement.

Models

There are 10 growth models in the program:

Model 1: 3-parameter von Bertalanffy growth model attempts to predict weight or length from the equation:

$$Yt = Y\infty(1-e^{-k(t-t0)})$$

where: Yt = length (or weight) at time t
$Y\infty$ = maximum length (or weight)
k = "growth" coefficient
$t0$ = theoretical age at which L (or W) = 0

The parameters to be estimated by NLLS regression in this model are $Y\infty$, k, and t0.

Model 2: 4-parameter von Bertalanffy uses a slightly different predictive equation:

$$Yt = Y\infty(1-e^{-k(t-t0)})^{b}$$

where: b = allometric growth constant

The parameters to be estimated are $Y\infty$, k, t0, and b.

Model 3: Logistic growth model uses the equation:

$$Yt = Y\infty(1+e^{-k(t-t0)})^{-1}$$

Model 4: Richards growth model uses the equation:

$$Yt = Y\infty(1+(e^{-k(t-t0)}/b))^{-b}$$

Model 5: Gompertz growth model uses the equation:

$$Yt = Y\infty(e^{-e^{-k(t-t0)}})$$

Parameters to be estimated in models 3, 4, and 5 are $Y\infty$, k, t0, and b in model 4.

Model 6: Kappenman's growth model uses the equation:

$$Yt = Y\infty(1-e^{-k(t*c)})$$

The parameters to be estimated are $Y\infty$, k, and c.

Choosing number 7 on the growth models menu produces another menu listing Schnute's four generalized growth models.

Case 1: 4-parameter model using the equation:

$$Yt = [Y1^b + (Y2^b - Y1^b)* \frac{(1-e^{-a(t-tau1)})}{(1-e^{-a(tau2-tau1)})}]^{1/b}$$

where: tau1 = youngest age in sample
tau2 = oldest age in sample

Parameters to be estimated are Y1, Y2, a, and b.

Case 2: 3-parameter model using the equation:

$$Yt = Y1(e[\log(Y2/Y1) * \frac{(1-e^{-a(t-tau1)})}{(1-e^{-a(tau2-tau1)})}])$$

Parameters to be estimated are Y1, Y2, and a.

Case 3: 3-parameter model using the equation:

$$Yt = [Y1^b + (Y2^b - Y1^b) * \frac{(t-tau1)}{(tau2-tau1)}]^{1/b}$$

Parameters to be estimated are Y1, Y2, and b.

Case 4: 2-parameter model using the equation:

$$Yt = Y1(e[\log(Y2/Y1) * \frac{(t-tau1)}{(tau2-tau1)}])$$

Parameters to be estimated are Y1, and Y2.

All the above growth models except number 6, from Kappenman (1980), were taken from Schnute (1981).

Output to Screen

After typing GROWBOOT at the DOS prompt, a descriptive banner is printed and the user is asked for the names of the input (AWL data), and output files. A second output file with the default name GBGRAPH.PRN is also created. If any of the output files already exist, the user can either write over the old files or rename the output files. The program then asks if a length or weight model is to be fit, what format the AWL data file is in, and if the user wants to see the data as it is being read in. A summary table showing mean length or weight at age of fish from the data file, and the number of fish with a valid length or weight measurement (depending on whether a length or weight model was chosen) is displayed. A title for the run is requested, and a menu showing the different growth models is printed. The user is asked for initial estimates of the growth model parameters, and if the default convergence parameters for the NLLS regression are acceptable. After the input data is regressed, the parameter estimates and summary statistics of the inital regression are output. The user is now asked for the number of bootstrap iterations to perform. The parameter estimates for each bootstrap iteration are printed to the screen after each regression. Finally, the user is asked if the residuals from the regression of the input data are to be printed to the output file. An example of screen output from GROWBOOT is shown in Appendix 1.

Output to File

Two output files are produced. One file contains:

1. summary table of the AWL data
2. growth curve parameter estimates and regression statistics for the raw AWL data
3. listing of the growth curve parameter estimates from all bootstraps
4. mean value of parameter estimates from all bootstraps and their associated 95% and 90% confidence intervals
5. summary statistics of growth parameter estimates for all bootstraps (see documentation for IMSL subroutine UVSTA for details)
6. variance-covariance matrix of growth curve parameters
7. mean values, for all bootstraps, of the mean length or weight at age, its standard deviation, and range

Residuals from the NLLS regression of the input data are output if requested. An example of the above output file is found in Appendix 2.

A second output file with the default name "GBGRAPH.PRN" contains the data needed for importation into a spreadsheet. Data consists of:

1. mean, standard deviation, range of length or weight at age of the input data and number of fish in age class.
2. parameter estimates of input data
3. means of the bootstrap parameter estimates and their associated 95% and 90% confidence intervals

An example of output file GBGRAPH.PRN is found in Appendix 3. From these data expected length or weight and "mean" residuals can be calculated and a graph showing observed and predicted growth, confidence intervals, and range of the data can be plotted.

Program Structure

A description of the structure of GROWBOOT.FOR is given in Appendix 4. Appendix 5 contains a flowchart detailing the organization of GROWBOOT and descriptions of the subroutines used in GROWBOOT are found in Appendix 6.

The program is easily modified to accept other growth models, including most common models. To use other models, the only subroutines that should need modifying are PICKMOD and MODELS. To change the default values for number of fish, oldest age class, and bootstrap replications, assign new values to the parameters MAXFISH, MAXBOOT, and MAXAGE in LB.INC and recompile. A MAKE file to automate compilation of GROWBOOT.EXE is found in Appendix 7. It is highly recommended that the program be run on a computer equipped with a math coprocessor. Execution time for a 403 record file, fit to a 3 parameter von Bertalanffy model (unweighted regression), using an IBM-XT with a 16-Mhz 80386 CPU and 80387 math coprocessor is approximately 15 seconds per bootstrap replication (average 3.15 iterations per bootstrap). Information describing the file header for GROWBOOT.EXE is shown in Appendix 8. Appendix 9 is a source code listing of GROWBOOT.EXE.

LITERATURE CITED

Efron, B. 1982. The jackknife, the bootstrap and other resampling plans. Soc. CBMS-NSF Regional Conference Series in Applied Mathematics #38.

Efron, B. 1987. Better bootstrap confidence intervals. Journal of the American Statistical Association **82**:171-200.

Efron, B. and R. Tibshirani. 1986. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. Statistical Science **1**:54-77.

Kappenman, K.F. 1981. A method for growth curve comparisons. Fishery Bulletin **79**(1):95-101

Schnute, J. 1981. A versatile growth model with statistically stable parameters. Canadian Journal of Fisheries and Aquatic Sciences **38**:1128-1140.

APPENDICES

Appendix 1. Output printed to the screen by GROWBOOT.EXE.
(user input and comments are in bold type)

```
C:\FORTRAN>GROWBOOT
*****************************************************
*   GROWBOOT - Growth curve estimation program with    *
*              estimation of parameter standard error  *
*              by bootstrap resampling.                *
*    Copyright (C) Charles Burkey, ADF&G 6/88          *
*****************************************************

ENTER INPUT FILE NAME: genf.r81

ENTER OUTPUT FILE NAME: genf.out

File exists; overwrite (Y/N)?: y

Graphics output File exists; overwrite (Y/N)?: y

Select LENGTH or WEIGHT model
           1 = LENGTH
           2 = WEIGHT    ==> 1

 Indicate format of data file: genf.r81

   1. TWO VARIABLES: age and length or weight only
      - FREE format in order: age, length or weight
   2. USER specified format
   NOTE: Program reads in variables in the order:
          age, length or weight, regression weight,
          frequency
   3. FREE format (age, length or weight, regression
                   weight, frequency)
      - 1 fish/line w/4 values separated by
        spaces (use 0 for missing values)

 Pick 1,2, or 3
1

If you want to see the data as it is read in
type Y, if not hit <ENTER>: y

... last 7 records from file GEN.R81

 Fish#:  397  Age:  2  DAT:  165.00  WT:     1.00  FREQ:     1.00
 Fish#:  398  Age:  1  DAT:  102.00  WT:     1.00  FREQ:     1.00
 Fish#:  399  Age:  2  DAT:  137.00  WT:     1.00  FREQ:     1.00
 Fish#:  400  Age:  1  DAT:  102.00  WT:     1.00  FREQ:     1.00
 Fish#:  401  Age:  2  DAT:  166.00  WT:     1.00  FREQ:     1.00
 Fish#:  402  Age:  1  DAT:  111.00  WT:     1.00  FREQ:     1.00
 Fish#:  403  Age:  2  DAT:  159.00  WT:     1.00  FREQ:     1.00
```

-Continued-

```
Summary of data in file: genf.r81
        Mean
Age    LENGTH    Number
 1     111.47      47.
 2     163.87      70.
 3     197.16     107.
 4     221.19      69.
 5     233.50     105.
 6     236.75       4.
 7     237.00       1.

No. fish with LENGTH measurment is    403
 Working, Please standby...


Enter title for this run (max 80 characters)
Kodiak, AK herring from General District, 1981 - seine caught

Pick a growth model
 1. 3 parameter von Bertanlanffy
    Y(t) = Y∞*EXP(-k(t-t0))
 2. 4 parameter von Bertanalaffy
    Y(t) = Y∞*EXP(-k(t-t0))**b
 3. Logistic
    Y(t) = Y∞*+EXP(-k(t-t0)))
 4. Richards
    Y(t) = Y∞*(EXP(-k(t-t0))/b))**-b
 5. Gompertz
    Y(t) = Y∞*(-EXP(-k(t-t0)))
 6. Kappenman
    Y(t) = Y∞*EXP(-k(EXP(t*c))))
 7. Schnute's Generalized Growth Models

  Y(t) = LENGTH at age t
   Y∞  = ultimate LENGTH (L∞ or W∞)
     k = "growth" coefficient
    t0 = theoretical age at which LENGTH=0
     b = allometric growth coefficient
1

Enter initial estimate of L∞
258

Enter initial estimate of k
.44

Enter initial estimate of t0
-.26
```

-Continued-

```
The default convergence parameters for the
Nonlinear least squares regression are:
   (1) Maximum number of iterations:        100
   (2) Maximum number of SSE evaluations:   400
   (3) Number of good digits in residuals:   24
   (4) Scaled gradient tolerance:        .3453E-03
   (5) Scaled step tolerance:            .2423E-04
   (6) Relative function tolerance:      .2423E-04
   (7) Absolute function tolerance:      .1422E-13
   (8) False convergence tolerance:      .1192E-04
   (9) Maximum allowable step size:  -999.0
  (10) Parameter scaling option: 1

Are these criteria OK (Y/N)? y

********* NLLS regression of input data *********
Iteration number:    1
Iteration number:    2
Iteration number:    3

Parameter estimates from Nonlinear least squares regression of input data
  Parameter      Estimate
    L∞            258.5
    k              .4439
    t0           -.2676
Rank of R matrix =    3
Sums of squares for error    =  75409.5900
Mean Square Error =    188.5240
Degrees of freedom for error =  400.000


Number of bootstrap replications to perform: 2

Bootstrap number:      1

Iteration number:    1
Iteration number:    2

 Parameter      Estimate
    L∞            260.9
    k              .4419
    t0           -.2205
```

-Continued-

```
Bootstrap number:     2

Iteration number:    1
Iteration number:    2
Iteration number:    3

 Parameter      Estimate
    L∞            256.0
    k              .4559
    t0            -.2123


Do you want to print the residuals (Y/N)?: y
```

End of output to screen from GROWBOOT.EXE

Appendix 2. Output file from GROWBOOT.EXE.

```
*****************************************************
*  GROWBOOT - Growth curve estimation program using  *
*           non-linear least squares regression     *
*           with estimation of parameter standard   *
*           error by bootstrap resampling.          *
*   Charles Burkey, ADF&G 8/88                      *
*****************************************************
```

Summary table of data in file: genf.r81

Age	Mean LENGTH	St Dev	Range		Number
1	111.5	23.6	197.0	84.0	47.
2	163.9	16.0	191.0	103.0	70.
3	197.2	9.6	228.0	160.0	107.
4	221.2	14.1	247.0	165.0	69.
5	233.5	9.0	251.0	205.0	105.
6	236.8	5.4	244.0	231.0	4.
7	237.0	5.4	237.0	237.0	1.

Growth model: three parameter von Bertanlanffy
of LENGTH at age

Parameter estimates from Nonlinear least squares regression of input data

Parameter	Estimate
L∞	258.5
k	.4439
t0	-.2674

Number of iterations = 3
Rank of R matrix = 3
Sums of squares for error = 75409.5700
Mean Square Error = 188.5239
Degrees of freedom for error = 400.000

GROWBOOT Growth curve bootstrap program.
by Charles Burkey, ADF&G - Juneau, Ak. 1988

Kodiak, AK herring from General District, 1981 - seine caught

Estimates of Growth Parameters for All BOOTSTRAP Replications

		Parameters		
Boot	MSE	L∞	k	t0
1	181.957	260.9	.4419	-.2205
2	192.365	256.0	.4559	-.2123

-Continued-

Mean Values and confidence intervals of Growth Parameters for All BOOTSTRAP Replications

	Mean	95% Confidence Interval Upper	95% Confidence Interval Lower	90% Confidence Interval Upper	90% Confidence Interval Lower
L∞	258.4	260.9	256.0	260.9	256.0
k	.4489	.4559	.4419	.4559	.4419
t0	-.2164	-.2123	-.2205	-.2123	-.2205

Summary Statistics of the Growth Parameter Estimates for All BOOTSTRAP Replications

	Variance	Skewness	Kurtosis	Min	Max	C.V.	95% C.I. of Mean Value Lower	95% C.I. of Mean Value Upper
L∞	11.7431	.0000	-2.0000	256.0	260.9	.0133	227.7	289.2
k	.0001	.0000	-2.0000	.4419	.4559	.0220	.3603	.5375
t0	.0000	.0000	-2.0000	-.2205	-.2123	-.0267	-.2683	-.1644

Variance-Covariance Matrix of Parameter Estimates

	L∞	k	t0
L∞	11.74		
k	-.3380E-01	.9726E-04	
t0	-.1981E-01	.5703E-04	.3343E-04

Mean LENGTH at Age of All BOOTSTRAP Replications.

Age	Mean	Range		Std. Dev.	Range	
1	108.5	111.5	108.5	13.0	23.6	12.0
2	163.4	163.9	161.2	13.6	16.0	12.8
3	197.1	198.8	197.1	12.8	13.0	9.6
4	219.6	221.9	219.6	15.2	15.2	14.1
5	233.6	234.0	233.5	13.9	14.2	9.0
6	242.2	243.2	236.8	7.5	9.4	5.4
7	253.1	259.0	237.0	7.5	9.4	5.4

Age	Observed LENGTH	Expected LENGTH	Residual
1	103.	111.	-8.2380
1	94.	111.	-17.2380
1	102.	111.	-9.2380
1	90.	111.	-21.2380
.			
.			

listing of residuals continues

Age	Observed LENGTH	Expected LENGTH	Residual
.			
.			
6	235.	242.	-7.4686
6	237.	242.	-5.4686
6	231.	242.	-11.4686
7	237.	248.	-11.2032

Appendix 3. Graphics output file GBGRAPH.PRN from program GROWBOOT.

Summary table of data in file: genf.r81

Age	Mean LENGTH	St Dev	Range		Number
1	111.5	23.6	197.0	84.0	47.
2	163.9	16.0	191.0	103.0	70.
3	197.2	9.6	228.0	160.0	107.
4	221.2	14.1	247.0	165.0	69.
5	233.5	9.0	251.0	205.0	105.
6	236.8	5.4	244.0	231.0	4.
7	237.0	5.4	237.0	237.0	1.

Kodiak, AK herring from General District, 1981 - seine caught

Growth model: three parameter von Bertanlanffy
of LENGTH at age

Parameter estimates from NLLS regression of input data

258.5
.4439
-.2674

Mean Values and confidence intervals of Growth Parameters for
All BOOTSTRAP Replications

	Ninety-five percent CI		Ninety percent CI	
Mean	Upper	Lower	Upper	Lower
258.4	260.9	256.0	260.9	256.0
.4489	.4559	.4419	.4559	.4419
-.2164	-.2123	-.2205	-.2123	-.2205

Appendix 4. Description of FORTRAN program GROWBOOT.FOR.

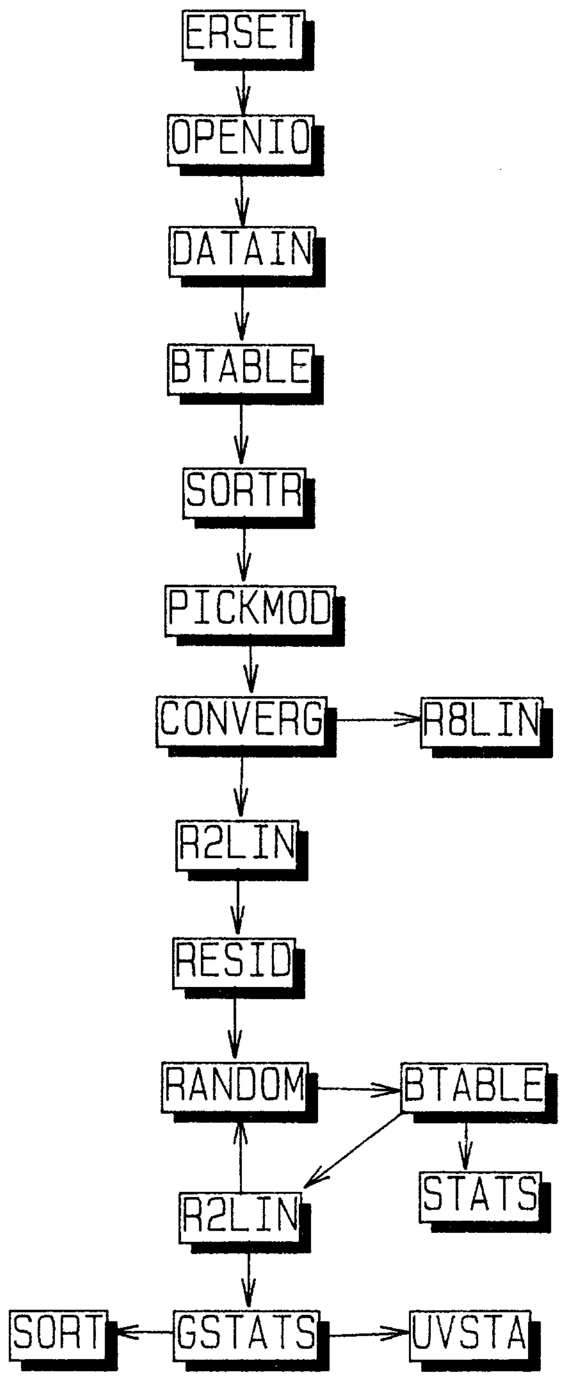
The program consists of 4 modules (files), containing 16 subroutines, 1 function, and an INCLUDE file.

MODULE	**SUBROUTINE**
GROWBOOT.FOR - main	ERSET - IMSL R2LIN - IMSL SORTR
DATAIN.FOR	OPENIO DATAIN BTABLE STATS
MODS.FOR	PICKMOD MODELS CONVERG R8LIN - IMSL
RANDOM.FOR	RANDOM RAND - function GSTATS SORT RESID

LB.INC is an include file containing COMMON statements and type statements for COMMON variables used in most of the SUBROUTINES.

Parameters: MAXFISH - maximum number of fish in sample
MAXBOOT - maximum number of bootstraps
MAXAGE - maximum age of a fish

Appendix 5. Flowchart of program GROWBOOT.



Appendix 6. Description of subroutines used in GROWBOOT.EXE:

SUBROUTINE ERSET - IMSL subroutine to control output from errors and warnings generated by IMSL subroutines

ERSET(3,0,0) -- turns off printing of warning messages

SUBROUTINE OPENIO - asks for input and output file names. Input file (UNIT=1, FILE=FILEIN or variable INPUT) contains AWL data (one line per fish). File opening errors are trapped and user is given the option of overwriting an existing file or renaming output files.

SUBROUTINE DATAIN - asks user what format the input AWL data file is in (variable CHOIC1) and reads it in accordingly. Only accepts fish between the age of 1 and MAXAGE. Puts data into 4 vectors; LAGE, LDAT, LWTH, and LFREQH (age, length or weight, regression weight, and frequency). User has the option of printing AWL input data to screen. Checks for number of fish older than MAXAGE, no fish in file, and no fish less than MAXAGE years old. Regression weight and frequency are set to 1.0E0 if input file does not include values for these variables.

SUBROUTINE BTABLE - Data are put into array TABLE. Finds minimum and maximum values of age (LOAGE, HIAGE), and length or weight at age.

SUBROUTINE STATS - called by BTABLE to compute mean length or weight and their standard deviations at age and store them in arrays (MEANS & SDS). Determines range of mean and standard deviation of length or weight at age for all bootstrap replications. Outputs mean, standard deviation, range of length or weight at age, and numbers in age class to GRAPH and OUTPUT files.

SUBROUTINE SORTR - sorts LAGE, LDAT, LWTH, AND LFREQH into ascending order based on value of LAGE. Necessary to put data in order by age so that residuals (calculated in SUBROUTINE MODELS) will match raw data values if user asks for them to be printed.

SUBROUTINE PICKMOD - describes growth models available, asks user to chose one (MODEL), sets the number of parameters to be estimated (NPARM), their names (PARMNAME), and a name for the model (MODNAME).

SUBROUTINE CONVERG - list default convergence parameters for the NLLS regression program (IPARAM, RPARAM) and allows user to change them if necessary. See attached documentation of IMSL subroutines R8LIN and R2LIN (a.k.a. RNLIN) for details.

SUBROUTINE RESID - calculates the expected value of length at age (if length based model) or weight at age (array EXPECT) from NLLS parameter estimates of raw data.

-Continued-

SUBROUTINE RANDOM - picks a residual at random from the array RES. Puts residual plus expected length or weight at that age into array LDAT. Puts the corresponding age into LAGE, regression weight into LWT, and frequency into LFREQ. Number of residuals chosen for each age class is the number of fish of that age class from the input sample with a length (or weight) measurement.

FUNCTION RAND - generates uniformly distributed random numbers from 0 to 1 (any number from 0 to 1 has an equal probability of being generated) which are used by RANDOM to pick bootstrap sample. Original seed values are calculated (in RANDOM) from values returned by the system clock using the intrinsic function GETTIME. Subsequent seeds are generated by RAND. See documentation of FUNCTION RAND in file RAND.FOR for origin of this routine.

SUBROUTINE R2LIN - see documentation for IMSL subroutine RNLIN

SUBROUTINE MODEL - called by R2LIN to compute the residual (E) at the given parameter vector THETA for a single observation. YDATA is assigned the observed value of length or weight of a fish, from vector LDAT, depending on whether a length or weight model is chosen. Prints the value of ITER to the screen after each iteration of R2LIN. A series of IF(MODEL .EQ. #) E=.... statements control which growth model is to be evaluated. Regression weight (WT) and frequency (FREQ) are assigned appropriate values from vectors LWT and LFREQ. See documentation of RNLIN for more details.

SUBROUTINE GSTATS - calculates covariance, and the 95% and 90% confidence intervals of the bootstrap parameters, and writes summary data to output file. The confidence intervals of parameter estimates are calculated by: (1) sorting the parameter array, PARMS, in ascending order using subroutine SORT (each column sorted individually since a column corresponds to all bootstrap estimates of a single parameter), and (2) picking the array elements most closely corresponding to the 2.5th and 97.5th percentile for 95% confidence intervals and the 5th and 95th percentile for 90% confidence intervals (eg. if 100 bootstraps were performed the 5th and 95th array elements would correspond exactly to the lower and upper 90% confidence bounds, respectively).

SUBROUTINE UVSTA - IMSL subroutine called by GSTATS to calculate summary statistics for the NLLS regression parameter estimates from all bootstraps. See IMSL documentation for details.

Appendix 7. MAKE file to automate compilation of GROWBOOT.EXE using the Microsoft(TM) utility MAKE.EXE

```
LB.EXE:     LB.INC
   FL /c /Fs LB.FOR DATAIN.FOR RAND.FOR MODS.FOR

LB.OBJ:     LB.FOR
   FL /c /Fs LB.FOR

DATAIN.OBJ:      DATAIN.FOR
   FL /c /Fs DATAIN.FOR

RAND.OBJ:  RAND.FOR
   FL /c /Fs RAND.FOR

MODS.OBJ:  MODS.FOR
   FL /c /Fs MODS.FOR

LB.EXE:     LB.OBJ DATAIN.OBJ RAND.OBJ MODS.OBJ
   LINK /SE:300 /E  LB+DATAIN+RAND+MODS,,,LLIBFOR7+STAT1+STAT2+SC;
```

Notes: 1. .OBJ files are LINKed with the Microsoft(TM) floating point library LLIBFOR7.LIB and the IMSL libraries STATCORE.LIB (abbreviated SC), STAT1.LIB, and STAT2.LIB
2. If the program is to be run on computers without a math coprocessor subsitiute LLIBFORE.LIB for LLIBFOR7.LIB in the LINK step
3. LINK option /SE:300 needed since program has more than the 128 segments allowed by default (due to IMSL subroutines)
4. LINK option /E produces a smaller (packed) .EXE file using the Microsoft(TM) utility EXEPACK.EXE

Appendix 8. Information in GROWBOOT.EXE header.

Microsoft (R) EXE File Header Utility Version 4.02

GROWBOOT.EXE	(hex)	(dec)
.EXE size (bytes)	4A77D	305021
Minimum load size (bytes)	6361D	407069
Overlay number	0	0
Initial CS:IP	0968:25B2	
Initial SS:SP	61BD:0800	2048
Minimum allocation (para)	190A	6410
Maximum allocation (para)	FFFF	65535
Header size (para)	20	32
Relocation table offset	1E	30
Relocation entries	0	0

Appendix 9. Source code for GROWBOOT.FOR.

```
************************************************************************
* GROWBOOT.FOR - Bootstrap estimation of growth equation parameters
*                using non-linear least squares regression.
*
*  SUBROUTINE ERSET - an IMSL routine to suppress printing of warnings
*                     generated by IMSL subprograms
*                     that do not require corrective action
************************************************************************

      INTEGER I,J,K,CHOIC2,BOOTS,INT,NPARM,IDERIV,IRANK,LDR,
     $        IWK(4),IPARAM(6),ITER,N
      REAL THETA(4),DFE,MODELS,R(4,4),SSE,THETA1(4),
     $     SCALE(4),WK(48),RPARAM(7)
      CHARACTER TITLE*80,MODNAME*50

$INCLUDE:'LB.INC'

      COMMON /NLLS/ITER
      EXTERNAL ERSET,DATAIN,RANDOM,BTABLE,RAND,OPENIO,STATS,GSTATS,
     $         MODELS,R2LIN,CONVERG,PICKMOD,RESID,SORTB
      DATA NPARM/0/,CHOIC2/0/,IDERIV/0/,THETA/4*0.0/,LDR/4/

      PRINT *,'*******************************************************'
      PRINT *,'*   GROWBOOT - Growth curve estimation program with    *'
      PRINT *,'*              estimation of parameter standard error  *'
      PRINT *,'*              by bootstrap resampling.                *'
      PRINT *,'*   Copyright (C) Charles Burkey, ADF&G 6/88           *'
      PRINT *,'*******************************************************'
      PRINT *,' '
      CALL ERSET(3,0,0)
      TIMES=1
      MODEL=0

C  Open input and output files

      CALL OPENIO

3     PRINT '(//A)',' Select LENGTH or WEIGHT model'
      PRINT *,'           1 = LENGTH'
      PRINT '(A\)','           2 = WEIGHT    ==> '
      READ(*,'(I1)',ERR=100) CHOIC2
100   IF(CHOIC2 .LT. 1 .OR. CHOIC2 .GT. 2) THEN
           PRINT *,' ERROR: Invalid choice - Pick 1 or 2'
           GO TO 3
      ENDIF
```

```
      IF(CHOIC2 .EQ. 1) THEN
           TYPE='LENGTH'
           PARMNAME(1)='L∞'
      ELSE
           TYPE='WEIGHT'
           PARMNAME(1)='W∞'
      ENDIF

C  Read in data from input file

      CALL DATAIN
      CALL BTABLE

C  Print out summary table of input data

      PRINT '(/A,A)',' Summary of data in file: ',FILEIN
      PRINT *,'        Mean'
      PRINT '(A,A6,A)',' Age   ',TYPE,'   Number'
      DO 2 I=LOAGE,HIAGE
           PRINT '(1X,I2,F10.2,F8.0)', I,MEANS(I),TABLE(I,1)
2     CONTINUE
      TAU(1)=LOAGE
      TAU(2)=HIAGE
      PRINT '(/A,A6,A,I6)',' No. fish with ',TYPE,' measurment is ',
     $                       NFISH
      PRINT *,' Working, Please standby...'
      CALL SORTR
      PRINT '(//A)',' Enter title for this run (max 80 characters)'
      READ(*,'(A80)') TITLE
      WRITE(GRAPH,'(//A80/)') TITLE

C  Go pick a model

      CALL PICKMOD(MODNAME,NPARM,MODEL,PARMNAME,TYPE,TAU)

C  Initial estimates of models parameters

      DO 10 I=1,NPARM
101        PRINT '(/A,A4)',' Enter initial estimate of ', PARMNAME(I)
           READ(*,*,ERR=101) THETA(I)
10    CONTINUE

C  Check convergence parameters for NLLS and change if necessary

      CALL CONVERG(RPARAM,IPARAM,NPARM,SCALE,PARMNAME)

C  Do the NLLS estimate of growth parameters using the input data

      PRINT '(//A)',' ********* NLLS regression of input data *********'
      CALL R2LIN(MODELS,NPARM,IDERIV,THETA,R,LDR,IRANK,DFE,SSE,
     $            IPARAM,RPARAM,SCALE,IWK,WK)
```

```
C  Output statistics from NLLS regression of input data
C  First to the screen

      WRITE(*,11)
11    FORMAT(1X,/' Parameter estimates from Nonlinear least squares regr
     $ession of input data',/1X,'  Parameter     Estimate')
      DO 12 J=1,NPARM
           WRITE(*,'(5X,A4,3X,G15.4)') PARMNAME(J),THETA(J)
           THETA1(J)=THETA(J)
12    CONTINUE
      WRITE(*,15) IRANK,SSE,SSE/DFE,DFE
15    FORMAT(1X,'Rank of R matrix = ',I3,/' Sums of squares for error
     $=',F12.4,/' Mean Square Error = ',F10.4,/' Degrees of freedom for
     $error = ',F8.3/)

C  To the output file

      WRITE(OUTPUT,14) MODNAME,TYPE
      WRITE(GRAPH,14) MODNAME,TYPE
14    FORMAT(1X,/' Growth model: ',A,/'                      of ',A6,' at age'
     $/)
      WRITE(OUTPUT,11)
      WRITE(GRAPH,'(/A/)')' Parameter estimates from NLLS regression of
     $input data'
      DO 13 I=1,NPARM
           WRITE(OUTPUT,'(5X,A4,3X,G15.4)') PARMNAME(I),THETA(I)
           WRITE(GRAPH,'(G15.4)') THETA(I)
13    CONTINUE
      WRITE(OUTPUT,'(/A,I5)') ' Number of iterations = ',ITER/3+1
      WRITE(OUTPUT,15) IRANK,SSE,SSE/DFE,DFE
      CALL RESID(THETA)

C  T H I S    I S    T H E    B O O T S T R A P    P A R T

6     PRINT '(1X,//A\)',' Number of bootstrap replications to perform: '
      READ(*,*,ERR=6) BOOTS
      IF(BOOTS .GT. MAXBOOT) THEN
           PRINT '(1X,A,I6,A)','ERROR: Program dimensioned for a maximum
     $ of ',MAXBOOT,' replications.'
           GO TO 6
      ENDIF
      IF(BOOTS .LT. 1) THEN
           CLOSE(OUTPUT)
           CLOSE(GRAPH)
           STOP 'Program terminated normally'
      ENDIF
      DO 7 I=1,BOOTS
           PRINT '(//A,I5/)',' Bootstrap number: ',I
           CALL RANDOM
           CALL BTABLE
           ITER=0
```

```
            CALL R2LIN(MODELS,NPARM,IDERIV,THETA,R,LDR,IRANK,DFE,SSE,
     $                  IPARAM,RPARAM,SCALE,IWK,WK)
            WRITE(*,'(1X,/A)') '  Parameter     Estimate'
            DO 18 J=1,NPARM
                 WRITE(*,'(5X,A4,3X,G15.4)') PARMNAME(J),THETA(J)
                 PARMS(I,J)=THETA(J)
                 THETA(J)=THETA1(J)
                 MSES(I)=SSE/DFE
18          CONTINUE
7       CONTINUE

C  B O O T S T R A P   O V E R
C  Compute and print out summary statistics of BOOTSTRAP parameter
C  estimates

        CALL GSTATS(BOOTS,NPARM,TITLE)
        END

        SUBROUTINE SORTR

*************************************************************************
*  Bubble sort routine to put arrays LAGE and LDAT into assending order
*  based on value in LAGE.  Linked sort that keeps corresponding elements
*  of all arrays in the same order.  This allows the residual array RES
*  (calculated in SUBROUTINE MODELS) to be in same order as its other
*  corresponding array elements if user requests printing of residuals.
*************************************************************************

        INTEGER I,J,IBND,IXCH,ITEMP
        REAL TEMP

$INCLUDE:'LB.INC'

        IBND=NFISH
30      IXCH=0
        DO 100 I=1,IBND-1
            IF (LAGE(I) .GT. LAGE(I+1)) THEN
                 TEMP=LDAT(I)
                 LDAT(I)=LDAT(I+1)
                 LDAT(I+1)=TEMP
                 TEMP=LWTH(I)
                 LWTH(I)=LWTH(I+1)
                 LWTH(I+1)=TEMP
                 TEMP=LFREQH(I)
                 LFREQH(I)=LFREQH(I+1)
                 LFREQH(I+1)=TEMP
                 ITEMP=LAGE(I)
                 LAGE(I)=LAGE(I+1)
                 LAGE(I+1)=ITEMP
                 IXCH=I
            ENDIF
```

```
100   CONTINUE
      IF (IXCH.EQ.0) GOTO 101
      IBND=IXCH
      GOTO 30
101   CONTINUE
      END

      BLOCK DATA
      INTEGER OUTPUT,INPUT,ITER,GRAPH
      CHARACTER*30 FILEIN,FILEOT,GRAFILE
      COMMON /INOUT/INPUT,OUTPUT,GRAPH,FILEIN,FILEOT,GRAFILE
      COMMON /NLLS/ITER
      DATA INPUT/1/,OUTPUT/2/,ITER/0/,GRAPH/3/
      END

      SUBROUTINE DATAIN

***********************************************************************
* SUBROUTINE FOR LVBBOOT.FOR
* Reads in data and puts it into 4 vectors after checking for a valid
* age (1-17).
*
* VECTORS: IAGE - age of fish
*          IDAT - length (or weight) of fish
*          IWTH - weight of observation in regression
*        LFREQH - frequency of observation
*
* User specifies format of data file
*                   -- free format
*                   -- user specified format
***********************************************************************

      LOGICAL EOF
      INTEGER CHOIC1,ISEL,IAGE,STAT,I,J,NOLD
      REAL DAT,WT,FREQ
      CHARACTER FMT*80,ANS*1

$INCLUDE:'LB.INC'

      DATA CHOIC1/0/,ISEL/0/,STAT/0/,ANS/' '/,NOLD/0/

      NFISH=0
      PRINT *,'                                                     '
      PRINT *,'                                                     '
      PRINT *,' Indicate format of data file: ', FILEIN
      PRINT *,'                                                     '
      PRINT *,'   1. TWO VARIABLES: age and length or weight only   '
      PRINT *,'      - FREE format in order: age, length or weight  '
      PRINT *,'   2. USER specified format                          '
      PRINT *,'   NOTE: Program reads in variables in the order:    '
      PRINT *,'          age, length or weight, regression weight,  '
```

```
      PRINT *,'           frequency                                   '
      PRINT *,'    3. FREE format (age, length or weight, regression   '
      PRINT *,'                  weight, frequency)                   '
      PRINT *,'       - 1 fish/line w/4 values separated by            '
      PRINT *,'         spaces (use 0 for missing values)             '
      PRINT *,'                                                       '
      PRINT *,' Pick 1,2, or 3                                        '
101   READ(*,'(I1)',ERR=102) CHOIC1
102   IF(CHOIC1 .LT. 1 .OR. CHOIC1 .GT. 3) THEN
           PRINT *,' ERROR: Invalid choice: Pick 1,2, or 3'
           GO TO 101
      ENDIF
      IF(CHOIC1 .EQ. 2) THEN
          PRINT *,' Enter data format in parenthesis (80 column limit).'
          PRINT *,' Use the T format if your data is not in            '
          PRINT *,' age, length or weight, regression weight,          '
          PRINT *,' frequency order.  Include 1 integer and 3 real     '
          PRINT *,' variables (in that order) in format descriptor.    '
          READ(*,'(A80)') FMT
      ENDIF
104   PRINT '(/A,/A\)',' If you want to see the data as it is read in',
     $                 ' type Y, if not hit <ENTER>: '
      READ(*,'(A1)',ERR=104) ANS
950   IF(.NOT. EOF(INPUT)) THEN
           IF(CHOIC1 .EQ. 1) READ(INPUT,*,END=1001,ERR=1000,IOSTAT=STAT)
     $                       IAGE,DAT
           IF(CHOIC1 .EQ. 2) READ(INPUT,FMT,END=1001,ERR=1000,
     $                       IOSTAT=STAT) IAGE,DAT,WT,FREQ
           IF(CHOIC1 .EQ. 3) READ(INPUT,*,END=1001,ERR=1000,IOSTAT=STAT)
     $                       IAGE,DAT,WT,FREQ
           IF(IAGE .GT. MAXAGE) NOLD=NOLD+1
           IF(IAGE .GT. 0 .AND. IAGE .LE. MAXAGE .AND. DAT .GT. 0.0)
     $     THEN
                NFISH=NFISH+1
                IF(NFISH .GT. MAXFISH) THEN
                     PRINT '(/A,/A,I6,A)',' ERROR: Maximum number of fis
     $h exceeded.',' Program dimensioned for a maximum of ',MAXFISH,' fi
     $sh.'
                     STOP
                ENDIF
                LAGE(NFISH)=IAGE
                LDAT(NFISH)=DAT
                LWTH(NFISH)=WT
                LFREQH(NFISH)=FREQ
                IF(WT .LE. 0.0) LWTH(NFISH)=1.0E0
                IF(FREQ .LE. 0.0) LFREQH(NFISH)=1.0E0
           ENDIF
```

```
            IF(ANS .EQ. 'y' .OR. ANS .EQ. 'Y') THEN
                 PRINT '(1X,A,I5,A,I3,3(A,F8.2))',' Fish#:',NFISH,
     $              '  Age:',IAGE,'  DAT:',DAT,'  WT:',LWTH(NFISH),
     $'  FREQ:',LFREQH(NFISH)
            ENDIF
            GO TO 950
      ENDIF

C  Error checking of input data

1000  IF(STAT .NE. 0) THEN
            PRINT *,' ERROR in input file: ', FILEIN
            PRINT *,' reading record No.: ', NFISH
            PRINT *,' STAT = ',STAT
            STOP
      ENDIF
1001  IF(NOLD .GT. 0) THEN
            PRINT '(/A,I5,A,I5,A,/A)',' WARNING: ',NOLD,' fish more than
     $',MAXAGE,' years old in input file.',
     $' Program will ignore these fish.'
      ENDIF
      IF(NFISH .EQ. 0) THEN
            PRINT '(A,I8)',' ERROR: No fish less than age ', MAXAGE
            PRINT *,' in file: ', FILEIN
            STOP
      ENDIF
      CLOSE(INPUT)
      END

**********************************************************************
      SUBROUTINE BTABLE
**********************************************************************

      INTEGER I,J
      REAL MAX,MIN

$INCLUDE:'LB.INC'

C  Zero out values in TABLE to start with clean slate

      DO 1 I=1,MAXAGE
      DO 1 J=1,3
1     TABLE(I,J)=0.0
```

```
C  To correctly determine the minimum length or weight in each
C  age class and the range of ages the appropriate variables must
C  be initalized with a value greater than the highest possible value

      IF(TIMES .EQ. 1) THEN
           DO 2 I=1,MAXAGE
                MINDAT(I)=1000.
2          CONTINUE
           LOAGE=1000
           HIAGE=0
      ENDIF

C  Accumulate age, and length or weight data into TABLE one fish
C  at a time.  NFISH = No. fish (from SUBROUTINE DATAIN)

      DO 100 I=1,NFISH
           IF(TIMES .EQ. 1) THEN
                HIAGE=MAX(HIAGE,LAGE(I))
                LOAGE=MIN(LOAGE,LAGE(I))

C  W E I G H T - L E N G T H   T A B L E

C  Calculates maximum and minimum lenght or weight in each age class

                MAXDAT(LAGE(I))=MAX(MAXDAT(LAGE(I)),LDAT(I))
                MINDAT(LAGE(I))=MIN(MINDAT(LAGE(I)),LDAT(I))
           ENDIF
           TABLE(LAGE(I),1)=TABLE(LAGE(I),1)+1
           TABLE(LAGE(I),2)=TABLE(LAGE(I),2)+LDAT(I)
           TABLE(LAGE(I),3)=TABLE(LAGE(I),3)+LDAT(I)*LDAT(I)
100   CONTINUE

C  END OF FILE, WE'RE DONE.
C  Compute mean length or weight at age and their
C  standard error using SUBROUTINE STATS

      IF(TIMES .EQ. 1) THEN

C  If there were no fish in an ageclass the minimum length or weight is
C  reset to 0

           DO 3 I=LOAGE,HIAGE
                IF(MINDAT(I) .EQ. 1000.) MINDAT(I)=0.
                IF(MINDAT(I) .EQ. 1000.) MINDAT(I)=0.
3          CONTINUE
      ENDIF
      CALL STATS
      TIMES=TIMES+1
      END
```

```
      SUBROUTINE OPENIO

************************************************************************
*ASKS FOR INPUT AND OUTPUT FILENAMES
*CONTAINS ERROR TRAPPING FOR FILE OPENING ERRORS
*STOP THE PROGRAM BY TYPING 'END'
************************************************************************

      INTEGER INPUT,OUTPUT,GRAPH,STAT
      CHARACTER*30 FILEIN,FILEOT,GRAFILE,IYES*1
      LOGICAL IANS
      COMMON /INOUT/INPUT,OUTPUT,GRAPH,FILEIN,FILEOT,GRAFILE

1     FORMAT(A)
2     FORMAT(1X,/A\)
3     FORMAT(1X,// ' ERROR NUMBER', I5, ' OPENING FILE',1X,A,1X,/
     $' PLEASE TRY AGAIN.'//)

*INPUT FILE

100   WRITE (*,2) ' ENTER INPUT FILE NAME: '
      READ(*,1) FILEIN
      IF (FILEIN.EQ.'END') STOP
      OPEN (INPUT, FILE= FILEIN, STATUS= 'OLD', IOSTAT= STAT)
      IF (STAT.NE.0) THEN
           WRITE (*,3) STAT, FILEIN
           GO TO 100
      ENDIF

*OUTPUT FILE

200   WRITE (*,2) ' ENTER OUTPUT FILE NAME: '
      READ (*,1) FILEOT
      IF (FILEOT.EQ.'END') THEN
           CLOSE (UNIT= INPUT)
           STOP
      ENDIF

C  Does the output file already exist?
C  If so, do you want to write over it?

      INQUIRE(file=FILEOT, EXIST=ians)
      if(ians) then
           print '(/A\)', ' File exists; overwrite (Y/N)?: '
           read(*,'(a)') iyes
           if(iyes.ne.'y'.and. iyes.ne.'Y') go to 200
      endif
      OPEN (OUTPUT, FILE= FILEOT, IOSTAT = STAT)
```

```
      IF (STAT.NE.0) THEN
           WRITE (*,3) STAT, FILEOT
           GO TO 200
      ENDIF

C  GRAPHICS OUTPUT FILE

      GRAFILE='GBGRAPH.PRN'

C  Does the output file already exist?
C  If so, do you want to write over it?

500   INQUIRE(file=GRAFILE, EXIST=ians)
      if(ians) then
           print '(/A\)', ' Graphics output File exists; overwrite (Y/N)
     $?: '
           read(*,'(a1)') iyes
           if(iyes.NE.'y'.AND. iyes.NE.'Y') THEN
                GO TO 300
           ELSE
                GO TO 400
           ENDIF
      endif
      GO TO 400
300   WRITE (*,2) ' ENTER GRAPHICS OUTPUT FILE NAME: '
      READ (*,1) GRAFILE
      IF (GRAFILE.EQ.'END') THEN
           CLOSE (UNIT= INPUT)
           CLOSE (UNIT= OUTPUT)
           STOP
      ENDIF
      GO TO 500
400   OPEN (GRAPH, FILE= GRAFILE, IOSTAT = STAT)
      IF (STAT.NE.0) THEN
           WRITE (*,3) STAT, FILEOT
           GO TO 300
      ENDIF
      END
```

```
      SUBROUTINE STATS

************************************************************************
*  SUBROUTINE STATS
*
*  Calculates mean and std. error of LENGTH or WEIGHT at age (1-HIAGE)
*  depending on type of model chosen.
*  Puts means and std. errors in vectors MEANS and SD
*  Largest and smallest mean and std. error at age in vectors
*  MXMEAN, MIMEAN, MXSD, and MISD
************************************************************************

      INTEGER M,I
      REAL STD2,SQRT,ABS,SDS,MAX,MIN

$INCLUDE:'LB.INC'

      IF(TIMES .EQ. 1) THEN
           DO 4 M=1,HIAGE
                MIMEAN(M)=1000.
                MISD(M)=1000.
4          CONTINUE
      ENDIF

      DO 1 M=1,HIAGE
           IF(TABLE(M,1) .LT. 2) GO TO 2
           STD2=TABLE(M,3)-TABLE(M,2)*TABLE(M,2)/TABLE(M,1)
           SDS=SQRT(ABS(STD2/(TABLE(M,1)-1.0)))
2          IF(TABLE(M,1) .EQ. 0) GO TO 6
           IF(TIMES .LE. 2) THEN
                MEANS(M)=TABLE(M,2)/TABLE(M,1)
                SD(M)=SDS
           ELSE
                MEANS(M)=((TIMES-2)*MEANS(M)+TABLE(M,2)/TABLE(M,1))
     $                          /(TIMES-1)
                SD(M)=((TIMES-2)*SD(M)+SDS)/(TIMES-1)
           ENDIF
6          MIMEAN(M)=MIN(MIMEAN(M),MEANS(M))
           MXMEAN(M)=MAX(MXMEAN(M),MEANS(M))
           MISD(M)=MIN(MISD(M),SD(M))
           MXSD(M)=MAX(MXSD(M),SD(M))
1     CONTINUE
      IF(TIMES .EQ. 1) THEN
           DO 5 M=1,HIAGE
                IF(MIMEAN(M) .EQ. 1000.) MIMEAN(M)=0.
                IF(MISD(M) .EQ. 1000.) MISD(M)=0.
5          CONTINUE
```

```
      WRITE(2,*)'*****************************************************'
      WRITE(2,*)'*  GROWBOOT - Growth curve estimation program using  *'
      WRITE(2,*)'*             non-linear least squares regression    *'
      WRITE(2,*)'*             with estimation of parameter standard  *'
      WRITE(2,*)'*             error by bootstrap resampling.         *'
      WRITE(2,*)'*  Charles Burkey, ADF&G 8/88                        *'
      WRITE(2,*)'*****************************************************'
      WRITE(2,*)' '

C  Write out mean length and weight at age and their minimum and maximum

           WRITE(OUTPUT,2041) FILEIN,TYPE
           WRITE(GRAPH,2041) FILEIN,TYPE
2041       FORMAT(1X,' Summary table of data in file: ',A,
     $//'         Mean',/
     $     ' Age     ',A6,'  St Dev      Range        Number')
           DO 2042 I=LOAGE,HIAGE
               WRITE(OUTPUT,'(I3,3X,4F8.1,F8.0)') I,MEANS(I),SD(I),
     $             MAXDAT(I),MINDAT(I),TABLE(I,1)
               WRITE(GRAPH,'(I3,3X,4F8.1,F8.0)') I,MEANS(I),SD(I),
     $             MAXDAT(I),MINDAT(I),TABLE(I,1)
2042       CONTINUE
      ENDIF
      END

      SUBROUTINE PICKMOD(MODNAME,NPARM,MODEL,PARMNAME,TYPE,TAU)

***********************************************************************
*   Menu for choosing a growth model.
*   Sets number and name of parameters to be estimated
*   (NPARM & PARMNAME), the ID number and name for the chosen model
*   (MODEL & MODNAME)
***********************************************************************

      INTEGER MODEL,NPARM
      REAL TAU(2)
      CHARACTER MODNAME*50,PARMNAME(4)*4,TYPE*6

      PARMNAME(4)='b '
1     PRINT '(/A)',' Pick a growth model'
      PRINT *,' 1. 3 parameter von Bertanlanffy'
      PRINT *,'    Y(t) = YEXP(-k(t-t0))'
      PRINT *,' 2. 4 parameter von Bertanalaffy'
      PRINT *,'    Y(t) = YEXP(-k(t-t0))**b'
      PRINT *,' 3. Logistic'
      PRINT *,'    Y(t) = Y+EXP(-k(t-t0)))'
      PRINT *,' 4. Richards'
      PRINT *,'    Y(t) = Y(EXP(-k(t-t0))/b))**-b'
      PRINT *,' 5. Gompertz'
      PRINT *,'    Y(t) = Y(-EXP(-k(t-t0)))'
```

```
      PRINT *,' 6. Kappenman'
      PRINT *,'     Y(t) = YEXP(-k(EXP(t*c))))'
      PRINT *,' 7. Schnute''s Generalized Growth Models'
      PRINT '(/A,A6,A)',' Y(t) = ',TYPE,' at age t'
      PRINT '(A,A6,A)','     Yultimate ',TYPE,' (L W
      PRINT *,'      k = "growth" coefficient'
      PRINT '(A,A6,A)','     t0 = theoretical age at which ',TYPE,'=0'
      PRINT *,'      b = allometric growth coefficient'
      READ(*,'(I2)',ERR=101) MODEL
101   IF(MODEL .LT. 1 .OR. MODEL .GT. 7) THEN
           PRINT '(/A)',' ERROR: Invalid choice'
           PRINT *,'       Pick a number between 1 and 7'
           GO TO 1
      ELSEIF(MODEL .EQ. 7) THEN
2          PRINT '(//A)',' Schnute''s Generalized Growth Models'
           PRINT *,' 8. Case 1: a <> 0, b <> 0'
           PRINT *,'     Y(t) = [Y1**b+(Y2**b-Y1**b)((1-EXP(-a(t-tau1)))'
           PRINT *,'            /(1-EXP(-a(tau2-tau1))))]**(1/b)'
           PRINT *,' 9. Case 2: a <> 0, b = 0'
           PRINT *,'     Y(t) = Y1*EXP[LOG(Y2/Y1)((1-EXP(-a(t-tau1)))'
           PRINT *,'            /(1-EXP(-a(tau2-tau1))))]'
           PRINT *,' 10. Case 3: a = 0, b <> 0'
           PRINT *,'     Y(t) = [Y1**b+(Y2**b-Y1**b)((t-tau1)'
           PRINT *,'            /(tau2-tau1))]**(1/b)'
           PRINT *,' 11. Case 4: a = 0, b = 0'
           PRINT *,'     Y(t) = Y1*EXP[LOG(Y2/Y1)((t-tau1)'
           PRINT '(A)','            /(tau2-tau1))]'
           PRINT '(/A,A6,A)',' Y(t) = ',TYPE,' at age t'
           PRINT '(A,A6,A,F4.0)','   Y1 = ',TYPE,' at age ',TAU(1)
           PRINT '(A,A6,A,F4.0)','   Y2 = ',TYPE,' at age ',TAU(2)
           PRINT *,'   a = constant relative rate of relative growth rat
     $e'
           PRINT *,'   b = incremental relative rate of relative growth
     $rate'
           PRINT '(A,F4.0)',' tau1 = age of youngest fish in sample =
     $', TAU(1)
           PRINT '(A,F4.0)',' tau2 = age of oldest fish in sample = ',
     $TAU(2)
           PRINT '(/A\)',' Choose 8,9,10 or 11: '
           READ(*,'(I2)',ERR=102) MODEL
102        IF(MODEL .LT. 8 .OR. MODEL .GT. 11) THEN
                PRINT *,' ERROR: Invalid choice'
                PRINT *,'       Pick number between 7 and 11'
                GO TO 2
           ENDIF
           PARMNAME(1)='Y1'
           PARMNAME(2)='Y2'
           PARMNAME(3)='a'
```

```
          IF(MODEL .EQ. 8) THEN
               NPARM=4
               MODNAME='Schnute''s Case 1: a <> 0, b <> 0'
          ELSEIF(MODEL .EQ. 11) THEN
               NPARM=2
               MODNAME='Schnute''s Case 4: a = 0, b = 0'
          ELSE
               NPARM=3
               MODNAME='Schnute''s Case 2: a <> 0, b = 0'
          ENDIF
          IF(MODEL .EQ. 10) THEN
               PARMNAME(3)='b'
               MODNAME='Schnute''s Case 3: a = 0, b <> 0'
          ENDIF
     ELSE
          PARMNAME(2)='k '
          PARMNAME(3)='t0'
          IF(MODEL .NE. 2 .AND. MODEL .NE. 4) THEN
             NPARM=3
             IF(MODEL .EQ. 1)MODNAME='three parameter von Bertanlanffy'
             IF(MODEL .EQ. 3)MODNAME='Logistic'
             IF(MODEL .EQ. 5)MODNAME='Gompertz'
             IF(MODEL .EQ. 6) THEN
                  MODNAME='Kappenman'
                  PARMNAME(3)='c'
             ENDIF
          ELSE
              NPARM=4
              IF(MODEL .EQ. 2)MODNAME='four parameter von Bertanlanffy'
              IF(MODEL .EQ. 4)MODNAME='Richards'
          ENDIF
     ENDIF
     END

     SUBROUTINE CONVERG(RPARAM,IPARAM,NPARM,SCALE,PARMNAME)

************************************************************************
*   Routine to change convergence parameters for Nonlinear least squares
*   regression.
*   IMSL SUBROUTINE R8LIN returns default convergence parameters and
*   sets the initialization flag to accept any user input convergence
*   settings.
*   Answering Y to the prompt " Are these criteria OK (Y/N)?" returns
*   control to the main program with the default settings intact.
************************************************************************

     INTEGER CHOICE,IPARAM(6),NPARM,I
     REAL RPARAM(7),SCALE(4)
     CHARACTER ANS*1,PARMNAME(4)*4
     EXTERNAL R8LIN
     DATA ANS/' '/
```

```
      CALL R8LIN(IPARAM,RPARAM)
1     PRINT '(//A,/A)',' The default convergence parameters for the',
     $' Nonlinear least squares regression are:'
      PRINT '(A,I4)','    (1) Maximum number of iterations:        ',
     $                      IPARAM(3)
      PRINT '(A,I4)','    (2) Maximum number of SSE evaluations:  ',
     $                      IPARAM(4)
      PRINT '(A,I2)','    (3) Number of good digits in residuals:    ',
     $                      IPARAM(2)
      PRINT '(A,G15.4)','    (4) Scaled gradient tolerance:  ',
     $                          RPARAM(1)
      PRINT '(A,G15.4)','    (5) Scaled step tolerance:       ',
     $                          RPARAM(2)
      PRINT '(A,G15.4)','    (6) Relative function tolerance:',
     $                          RPARAM(3)
      PRINT '(A,G15.4)','    (7) Absolute function tolerance:',
     $                          RPARAM(4)
      PRINT '(A,G15.4)','    (8) False convergence tolerance:',
     $                          RPARAM(5)
      PRINT '(A,G12.4)','    (9) Maximum allowable step size:',
     $                          RPARAM(6)
      PRINT '(A,I1)','   (10) Parameter scaling option: ',IPARAM(6)
      IF(ANS .EQ. ' ') THEN
17         PRINT '(/A\)',' Are these criteria OK (Y/N)? '
           READ(*,'(A1)') ANS
           IF(ANS .EQ. 'y' .OR. ANS .EQ. 'Y') THEN
                RETURN
           ELSEIF(ANS .EQ. 'n' .OR. ANS .EQ. 'N') THEN
                GO TO 1
           ELSE
                PRINT '(A/A)',' ERROR: Invalid choice','        Pick Y o
     $r N'
                GO TO 17
           ENDIF
      ENDIF

C  The rest of the routine is executed only if user wants to change
C  the default convergence paremeters

      PRINT *,'  (11) STOP changing parameters'
3     PRINT '(/A\)',' Pick a convergence parameter to change: '
      READ(*,'(I2)',ERR=103) CHOICE
103   IF(CHOICE .LT. 1 .OR. CHOICE .GT. 11) THEN
           PRINT *,' ERROR: Invalid choice'
           PRINT *,'        Pick number between 1 and 11'
           GO TO 3
      ENDIF
      IF(CHOICE .GE. 4 .AND. CHOICE .LT. 11) PRINT *,' Be sure to punch
     $ the decimal point'
```

```
      IF(CHOICE .EQ. 1) THEN
104        PRINT '(/A\)',' Enter new value for max No. of iterations: '
           READ(*,'(I4)',ERR=104) IPARAM(3)
      ELSEIF(CHOICE .EQ. 2) THEN
105        PRINT '(/A\)',' Enter new value for max No. SSE evaluations '
           READ(*,'(I4)',ERR=105) IPARAM(4)
      ELSEIF(CHOICE .EQ. 3) THEN
106        PRINT '(/A\)',' Enter new value for No. good digits in residu
     $als: '
           READ(*,'(I2)',ERR=106) IPARAM(2)
      ELSEIF(CHOICE .EQ. 4) THEN
107        PRINT '(/A\)',' Enter new value for scaled gradient tolerance
     $: '
           READ(*,'(G15.4)',ERR=107) RPARAM(1)
      ELSEIF(CHOICE .EQ. 5) THEN
108        PRINT '(/A\)',' Enter new value for scaled step tolerance: '
           READ(*,'(G15.4)',ERR=108) RPARAM(2)
      ELSEIF(CHOICE .EQ. 6) THEN
109       PRINT '(/A\)',' Enter new value for relative function toleran
     $ce: '
           READ(*,'(G15.4)',ERR=109) RPARAM(3)
      ELSEIF(CHOICE .EQ. 7) THEN
110        PRINT '(/A\)',' Enter new value for absolute function toleran
     $ce: '
           READ(*,'(G15.4)',ERR=110) RPARAM(4)
      ELSEIF(CHOICE .EQ. 8) THEN
111        PRINT '(/A\)',' Enter new value for false convergence toleran
     $ce: '
           READ(*,'(G15.4)',ERR=111) RPARAM(5)
      ELSEIF(CHOICE .EQ. 9) THEN
112        PRINT '(/A\)',' Enter new value for max step size: '
           READ(*,'(G12.4)',ERR=112) RPARAM(6)
      ELSEIF(CHOICE .EQ. 10) THEN
           IPARAM(6)=0
           DO 2 I=1,NPARM
113             PRINT '(/A,A4,A,F8.3)',' Present scaling value for ',
     $                                PARMNAME(I),' is: ',SCALE(I)
                PRINT '(A,A4,A\)',' Enter new scaling value for ',
     $                                PARMNAME(I),': '
                READ(*,'(F8.3)',ERR=113) SCALE(I)
2          CONTINUE
      ELSE
           RETURN
      ENDIF
      GO TO 1
      END
```

```
      SUBROUTINE MODELS(NPARM,THETA,IOPT,IOBS,FREQ,WT,E,DE,IEND)

***********************************************************************
*    Routine CALLed by IMSL SUBROUTINE RNLIN to calculate the residual
*    at the given parameter vector THETA for a single observation.
*    MODELS contains the different growth equations in the series of
*    lines:  IF(MODEL .EQ. #) E=(dependent variable) - (growth equation)
*      MODEL = chosen growth model number
*      E = residual
*      dependent variable = length or weight of fish (YDATA)
*      growth equation = equation to predict length or weight as a
*                        function of age
***********************************************************************

      INTEGER NPARM,IOPT,IOBS,IEND,ITER,MOD
      REAL THETA(4),FREQ,WT,E,DE(1),EXP,YDATA,DEN,Y1B,Y2B,LOG
      INTRINSIC EXP,LOG

$INCLUDE:'LB.INC'

      COMMON /NLLS/ITER

      IF(IOBS .LE. NFISH) THEN
           IF(TIMES .EQ. 2) THEN
                WT=LWTH(IOBS)
                FREQ=LFREQH(IOBS)
           ELSE
                WT=LWT(IOBS)
                FREQ=LFREQ(IOBS)
           ENDIF
           IEND=0

C  If length model chosen put length into YDATA, or put weight into
C  YDATA if a weight model is chosen

           YDATA=LDAT(IOBS)
           IF(MODEL .EQ. 1) E=YDATA-THETA(1)*(1.0-EXP(-THETA(2)
     $                        *(LAGE(IOBS)-THETA(3))))
           IF(MODEL .EQ. 2) E=YDATA-THETA(1)*(1.0-EXP(-THETA(2)
     $                        *(LAGE(IOBS)-THETA(3))))**THETA(4)
           IF(MODEL .EQ. 3) E=YDATA-THETA(1)/(1+EXP(-THETA(2)
     $                        *(LAGE(IOBS)-THETA(3))))
           IF(MODEL .EQ. 4) E=YDATA-THETA(1)*(1+(EXP(-THETA(2)
     $                        *(LAGE(IOBS)-THETA(3))))/THETA(4))
     $                        **(-THETA(4))
           IF(MODEL .EQ. 5) E=YDATA-THETA(1)*EXP(-EXP(-THETA(2)
     $                        *(LAGE(IOBS)-THETA(3))))
           IF(MODEL .EQ. 6) E=YDATA-THETA(1)*(1-EXP(-THETA(2)
     $                        *EXP(LAGE(IOBS)*THETA(3))))
```

```
          IF(MODEL .EQ. 8) THEN
               DEN=1.0E0-EXP(-THETA(3)*(TAU(2)-TAU(1)))
               Y1B=THETA(1)**THETA(4)
               Y2B=THETA(2)**THETA(4)
               E=YDATA-(Y1B+(Y2B-Y1B)*(1.0E0-EXP(-THETA(3)*(LAGE(IOBS)
     $           -TAU(1))))/DEN)**(1.0E0/THETA(4))
          ENDIF
          IF(MODEL .EQ. 9) THEN
               DEN=1.0E0-EXP(-THETA(3)*(TAU(2)-TAU(1)))
               E=YDATA-THETA(1)*EXP(LOG(THETA(2)/THETA(1))*
     $           (1.0E0-EXP(-THETA(3)*(LAGE(IOBS)-TAU(1))))/DEN)
          ENDIF
          IF(MODEL .EQ. 10) THEN
               Y1B=THETA(1)**THETA(3)
               Y2B=THETA(2)**THETA(3)
               E=YDATA-(Y1B+(Y2B-Y1B)*((LAGE(IOBS)
     $           -TAU(1))/(TAU(2)-TAU(1))))**(1.0E0/THETA(3))
          ENDIF
          IF(MODEL .EQ. 11) E=YDATA-THETA(1)*EXP(LOG(THETA(2)/THETA(1))
     $                          *((LAGE(IOBS)-TAU(1))/(TAU(2)-TAU(1))))
          IF(TIMES .EQ. 2) RES(IOBS)=E
      ELSE
          IEND=1

C  Count number of iterations taken by NLLS regression (R2LIN) and
C  update screen when value changes

          ITER=ITER+1
          IF(MOD(ITER,3) .EQ. 1) PRINT '(A,I4)',' Iteration number: ',
     $                             ITER/3+1
      ENDIF
      END

************************************************************************
      SUBROUTINE RANDOM
************************************************************************

      INTEGER IHR,IMIN,ISEC,I100TH,RNFISH,I,J,INT
      INTEGER*4 SEEDC,SEEDF
      REAL*8 RND,RAND

$INCLUDE:'LB.INC'

      COMMON /SUPER/SEEDC,SEEDF

C  Initalize seed for random number generator (FUNCTION RAND)
C  using the system clock

      CALL GETTIME(IHR,IMIN,ISEC,I100TH)
      SEEDC=(IHR+1)*(IMIN+1)*(ISEC+1)*(I100TH+1)
```

```
1     IF(SEEDC .LT. 5000) THEN
           SEEDC=SEEDC*10
           GO TO 1
      ENDIF
      SEEDF=SEEDC/10

C  SEEDF must be less than 2048

2     IF(SEEDF .GT. 2047) THEN
           SEEDF=SEEDF/10
           GO TO 2
      ENDIF
      RNFISH=1

C  Pick bootstrap residuals from RES, add to expected values in EXPECT,
C  and put in LLT if length based model or LWT if weight based model
C  TABLE(I,1) = # fish with length or weight measurement in age class I
C  RND = uniform random number generated by FUNCTION RAND

      DO 3 I=1,HIAGE
          IF(TABLE(I,1) .GT. 0) THEN
               DO 4 J=1,TABLE(I,1)
                    RND=RAND()
                    LDAT(RNFISH)=RES(INT(NFISH*RND)+1)+EXPECT(I)
                    LWT(RNFISH)=LWTH(INT(NFISH*RND)+1)
                    LFREQ(RNFISH)=LFREQH(INT(NFISH*RND)+1)
                    LAGE(RNFISH)=I
                    RNFISH=RNFISH+1
4              CONTINUE
          ENDIF
3     CONTINUE
      END

      REAL*8 FUNCTION RAND

***********************************************************************
c "Super-Duper" Pseaudorandom Number Generator
c Debra K. Hofmeister and George Woodworth
c Science Software 3(2): 100-102, 1987
c Installed by T. Quinn, 9/3/87
c Sample usage shown below:
c     common/super/ seedc,seedf
c     integer*4 seedc, seedf
c     real*8 rnd, rand
c     print *,'Input 2 odd seeds, the second between 0 & 2047:'
c     read(*,*) seedc,seedf
c     rnd=rand()
***********************************************************************
```

```
      integer*4 i126,ixl0,ixh0,ixl1,ixh1,ix,seedc,seedf,ISHFT,IEOR
      common/super/ seedc,seedf
      equivalence(u,ix)
      i126=1056964608
c  Congruential part
      ixl0=ishft(seedc,16)
      ixl0=ishft(ixl0,-16)
      ixh0=ishft(seedc,-16)
      ixl1=ixl0*3533
      ixh1=ixl0+ixh0*3533+ishft(ixl1,-16)
        ixh1=ishft(ixh1,16)
        ixh1=ishft(ixh1,-16)
        ixl1=ishft(ixl1,16)
        ixl1=ishft(ixl1,-16)
      ixl0=ixl1
      ixh0=ixh1
      seedc=ieor(ishft(ixh0,16),ixl0)
c  Feedback Shift Part
      seedf=ieor(ishft(seedf,-15),seedf)
      seedf=ieor(ishft(seedf, 17),seedf)
      ix=ieor(seedc,seedf)
      ix=ishft(ix,-9)
      ix=ieor(ix,i126)
      rand=2*u-1
      return
      end

      SUBROUTINE GSTATS(NROW,NPARM,TITLE)

***********************************************************************
*   Computes for all bootstraps:
*       (1) grand mean of growth parameter estimates
*       (2) basic univariate statistics of parameter estimates
*       (3) covariance matrix of parameter estimates
*
*   CALLs IMSL SUBROUTINE UVSTA to compute univariate statistics of
*   the parameter estimates
***********************************************************************

      INTEGER LDSTAT,LDX,NVAR,NPARM,K,L
      PARAMETER (LDSTAT=15)
      INTEGER I,J,IDO,IFRQ,IWT,MOPT,NRMISS,NROW,IPRINT
      REAL CONPRM,CONPRV,STAT(LDSTAT,4),
     $      COVAR,COV(6),TEMP
      CHARACTER TITLE*80,IYES*1

$INCLUDE:'LB.INC'

      DATA IDO/0/,IFRQ/0/,IWT/0/,MOPT/0/,IPRINT/0/,CONPRM/95.0/,
     $      CONPRV/95.0/,LDX/MAXBOOT/,COVAR/0.0/,COV/6*0.0/
```

```
      NVAR=NPARM

C  Calculate summary statistics of NLLS regression
C  parameter estimates for all bootstrap replocations

      CALL UVSTA(IDO,NROW,NVAR,X,LDX,IFRQ,IWT,MOPT,CONPRM,CONPRV,
     $            IPRINT,STAT,LDSTAT,NRMISS)
      WRITE(OUTPUT,19) TITLE
19    FORMAT(1X,///////' GROWBOOT Growth curve bootstrap program.'/
     $'  by Charles Burkey, ADF&G - Juneau, Ak. 1988'//,A80/)

C  Write NLLS parameter estimates for all bootstrap replications

      WRITE(OUTPUT,8) (PARMNAME(I),I=1,NPARM)
8     FORMAT(1X,///' Estimates of Growth Parameters for All BOOTSTRAP Re
     $plications',//26X,' Parameters',/' Boot    MSE',4X,4(5X,A4,5X))
      DO 9 I=1,NROW
           WRITE(OUTPUT,'(I4,F10.3,4G15.4)') I,MSES(I),
     $                                       (PARMS(I,J),J=1,NPARM)
9     CONTINUE

      IF(NROW .GT. 1) THEN

C  Calculate covariance of all parameters

      L=1
      DO 25 I=1,NPARM-1
           DO 26 J=I+1,NPARM
                DO 27 K=1,NROW
                     COVAR=COVAR+((PARMS(K,I)-STAT(1,I))*(PARMS(K,J)-
     $                      STAT(1,J)))
27              CONTINUE
                COV(L)=COVAR/(NROW-1.0)
                COVAR=0
                L=L+1
26         CONTINUE
25    CONTINUE
      IF(NPARM .EQ. 4) THEN
           TEMP=COV(3)
           COV(3)=COV(4)
           COV(4)=TEMP
      ENDIF

C  Sort the parameter estimates (individually) in assending order

      CALL SORT(NROW,NPARM)
```

```
C  Write mean of parameter estimates for all bootstrap replications
C  and their corresponding 95% and 90% percentile ('confidence') values

      WRITE(OUTPUT,21)
      WRITE(GRAPH,22)
21    FORMAT(1X,/' Mean Values and confidence intervals of Growth Parame
     $ters for All BOOTSTRAP Replications',/25X,'95% Confidence Interval
     $',7X,'90% Confidence Interval',/11X,'Mean',11X,'Upper',10X,'Lower'
     $,10X,'Upper',10X,'Lower')
22    FORMAT(1X,/' Mean Values and confidence intervals of Growth Parame
     $ters for All BOOTSTRAP Replications',/20X,'Ninety-five percent CI
     $',7X,'  Ninety percent CI',/6X,'Mean',11X,'Upper',10X,'Lower'
     $,10X,'Upper',10X,'Lower')
      DO 20 I=1,NPARM
           IF(NROW .GT. 39) THEN
           WRITE(OUTPUT,'(1X,A4,5G15.4)') PARMNAME(I),STAT(1,I),
     $              PARMS((NROW*.975),I),PARMS((NROW*.025),I),
     $              PARMS((NROW*.95),I),PARMS((NROW*.05),I)
           WRITE(GRAPH,'(5G15.4)') STAT(1,I),
     $              PARMS((NROW*.975),I),PARMS((NROW*.025),I),
     $              PARMS((NROW*.95),I),PARMS((NROW*.05),I)
           ELSE
           WRITE(OUTPUT,'(1X,A4,5G15.4)') PARMNAME(I),STAT(1,I),
     $            PARMS((NROW*.975)+1,I),PARMS((NROW*.025)+1,I),
     $            PARMS((NROW*.95)+1,I),PARMS((NROW*.05)+1,I)
           WRITE(GRAPH,'(5G15.4)') STAT(1,I),
     $            PARMS((NROW*.975)+1,I),PARMS((NROW*.025)+1,I),
     $            PARMS((NROW*.95)+1,I),PARMS((NROW*.05)+1,I)
           ENDIF
20    CONTINUE
      WRITE(OUTPUT,'(/A/)') ' Summary Statistics of the Growth Parameter
     $ Estimates for All BOOTSTRAP Replications'
      WRITE(OUTPUT,30)
30    FORMAT(1X,64X,'95% C.I. of Mean Value',/4X,' Variance',2X,'Skewnes
     $s',2X,'Kurtosis',4X,'Min',9X,'Max',7X,'C.V.',4X,'Lower',7X,'Upper'
     $)
      DO 10 I=1,NPARM
          WRITE(OUTPUT,'(A4,F9.4,2F10.4,2G12.4,F7.4,2G12.4)')
     $     PARMNAME(I),STAT(2,I),(STAT(J,I),J=4,7),
     $     STAT(9,I),(STAT(J,I),J=11,12)
10    CONTINUE

C  Print variance-covariance matrix

      WRITE(OUTPUT,'(/A,/4X,4(6X,A4,6X))') ' Variance-Covariance Matrix
     $of Parameter Estimates', (PARMNAME(I),I=1,NPARM)
      WRITE(OUTPUT,'(A4,G15.4)') PARMNAME(1),STAT(2,1)
      J=1
```

```
      DO 28 I=1,NPARM-1
           WRITE(OUTPUT,'(A4,4G15.4)') PARMNAME(I+1),(COV(J+K-1),K=1,I),
     $                                 STAT(2,I+1)
           J=J+I
28    CONTINUE
      ENDIF

      WRITE(OUTPUT,'(/A,A6,A//,A,A)') ' Mean ',TYPE,' at Age of All BOOT
     $STRAP Replications.',' Age','   Mean       Range      Std. Dev.
     $ Range'
      DO 31 I=LOAGE,HIAGE
           WRITE(OUTPUT,'(I3,6F8.1)') I,MEANS(I),MXMEAN(I),MIMEAN(I),
     $                                SD(I),MXSD(I),MISD(I)
31    CONTINUE

      PRINT '(//A\)',' Do you want to print the residuals (Y/N)?: '
      READ(*,'(A1)') iyes
      IF(iyes .EQ. 'y' .OR. iyes .EQ. 'Y') THEN
           WRITE(OUTPUT,23) TYPE,TYPE
23         FORMAT(1X,//'       Observed  Expected',/1X,' Age',3X,A6,4X,
     $A6,'  Residual'/)
           DO 24 I=1,NFISH
               WRITE(OUTPUT,29) LAGE(I),EXPECT(LAGE(I))+RES(I),
     $                          EXPECT(LAGE(I)),RES(I)
24         CONTINUE
29         FORMAT(1X,I3,F10.0,1X,F9.0,F11.4)
      ENDIF
      END

      SUBROUTINE RESID(THETA)

***********************************************************************
*  Calculates the expected value of LENGTH or WEIGHT at age from the
*  parameters (THETA) estimated by NLLS regression of the input data
*  CALLed when user requests printing of residuals
***********************************************************************

      INTEGER I
      REAL LOG,EXP,DEN,Y1B,Y2B,THETA(4)
      INTRINSIC EXP,LOG

$INCLUDE:'LB.INC'

      DO 1 I=1,HIAGE
           IF(MODEL .EQ. 1) EXPECT(I)=THETA(1)*(1.0-EXP(-THETA(2)
     $                          *(I-THETA(3))))
           IF(MODEL .EQ. 2) EXPECT(I)=THETA(1)*(1.0-EXP(-THETA(2)
     $                          *(I-THETA(3))))**THETA(4)
           IF(MODEL .EQ. 3) EXPECT(I)=THETA(1)/(1+EXP(-THETA(2)
     $                          *(I-THETA(3))))
```

```
          IF(MODEL .EQ. 4) EXPECT(I)=THETA(1)*(1+(EXP(-THETA(2)
     $                      *(I-THETA(3))))/THETA(4))
     $                      **(-THETA(4))
          IF(MODEL .EQ. 5) EXPECT(I)=THETA(1)*EXP(-EXP(-THETA(2)
     $                      *(I-THETA(3))))
          IF(MODEL .EQ. 6) EXPECT(I)=THETA(1)*(1-EXP(-THETA(2)
     $                      *EXP(I*THETA(3))))
          IF(MODEL .EQ. 8) THEN
               DEN=1.0E0-EXP(-THETA(3)*(TAU(2)-TAU(1)))
               Y1B=THETA(1)**THETA(4)
               Y2B=THETA(2)**THETA(4)
               EXPECT(I)=(Y1B+(Y2B-Y1B)*(1.0E0-EXP(-THETA(3)*(I
     $           -TAU(1))))/DEN)**(1.0E0/THETA(4))
          ENDIF
          IF(MODEL .EQ. 9) THEN
               DEN=1.0E0-EXP(-THETA(3)*(TAU(2)-TAU(1)))
               EXPECT(I)=THETA(1)*EXP(LOG(THETA(2)/THETA(1))*
     $           (1.0E0-EXP(-THETA(3)*(I-TAU(1))))/DEN)
          ENDIF
          IF(MODEL .EQ. 10) THEN
               Y1B=THETA(1)**THETA(3)
               Y2B=THETA(2)**THETA(3)
               EXPECT(I)=(Y1B+(Y2B-Y1B)*((I
     $           -TAU(1))/(TAU(2)-TAU(1))))**(1.0E0/THETA(3))
          ENDIF
          IF(MODEL .EQ. 11) EXPECT(I)=THETA(1)*EXP(LOG(THETA(2)
     $                      /THETA(1))*((I-TAU(1))/(TAU(2)-TAU(1))))
1       CONTINUE
        END
```

```
      SUBROUTINE SORT(NROW,NPARM)

**********************************************************************
*   Sorts parameters from growth curve bootstrap replications into
*   assending order using the "bubble sort"
**********************************************************************

      REAL TEMP
      INTEGER IBND,IXCH,I,J,NROW,NPARM

$INCLUDE:'LB.INC'

      IBND=NROW
      DO 101 J=1,NPARM
30          IXCH=0
            DO 100 I=1,IBND-1
                  IF (PARMS(I,J).GT.PARMS(I+1,J)) THEN
                        TEMP=PARMS(I,J)
                        PARMS(I,J)=PARMS(I+1,J)
                        PARMS(I+1,J)=TEMP
                        IXCH=I
                  ENDIF
100         CONTINUE
            IF (IXCH.EQ.0) THEN
                  IBND=NROW
                  GOTO 101
            ENDIF
            IBND=IXCH
            GOTO 30
101   CONTINUE
      END
```

```
* INCLUDE file LB.INC

      INTEGER MAXFISH,NFISH,INPUT,OUTPUT,TIMES,MAXBOOT,
     $        MODEL,GRAPH,MAXAGE,HIAGE,LOAGE
      PARAMETER (MAXFISH=2000, MAXBOOT=1000, MAXAGE=100)
      INTEGER*2 LAGE(MAXFISH)
      REAL LDAT[HUGE](MAXFISH),LWT[HUGE](MAXFISH),MSES[HUGE](MAXBOOT),
     $     LWTH[HUGE](MAXFISH),LFREQ[HUGE](MAXFISH),TABLE(MAXAGE,3),
     $     X[HUGE](MAXBOOT,4),TAU(2),EXPECT(MAXAGE),SD(MAXAGE),
     $     PARMS[HUGE](MAXBOOT,4),MEANS(MAXAGE),LFREQH[HUGE](MAXFISH),
     $     RES[HUGE](MAXFISH),MIMEAN(MAXAGE),MXMEAN(MAXAGE),
     $     MISD(MAXAGE),MXSD(MAXAGE),MAXDAT(MAXAGE),MINDAT(MAXAGE)
      CHARACTER*30 FILEIN,FILEOT,GRAFILE,TYPE*6,PARMNAME(4)*4
      COMMON /INOUT/ INPUT,OUTPUT,GRAPH,FILEIN,FILEOT,GRAFILE
      COMMON /DATAI/ LDAT,LWT,LAGE,RES,HIAGE,LOAGE,LWTH,LFREQ,LFREQH
      COMMON /STAT/ SD,NFISH,TIMES,MEANS,TABLE,MAXDAT,MINDAT
      COMMON /GSTAT/ X,PARMNAME,MSES,MIMEAN,MXMEAN,MISD,MXSD
      COMMON /MOD/ MODEL,TYPE,TAU,EXPECT
      EQUIVALENCE (X(1,1),PARMS(1,1))
```