FRI-UW-8806
May 1988

FISHERIES RESEARCH INSTITUTE
School of Fisheries
College of Ocean and Fishery Sciences
University of Washington
Seattle, Washington 98195

STATISTICAL METHODOLOGY FOR ESTIMATING COMPOSITION OF HIGH SEAS SALMONID MIXTURES USING SCALE ANALYSIS

by

Russell B. Millar

Submitted to

International North Pacific Fisheries Commission

by the

United States National Section

May 1988

THIS PAPER MAY BE CITED IN THE FOLLOWING MANNER:

Millar, R.B. 1988. Statistical methodology for estimating composition of high seas salmonid mixtures using scale analysis. (Document submitted to the International North Pacific Fisheries Commission.) 57 pp. FRI-UW-8806, Fisheries Research Institute, University of Washington, Seattle.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SIMULATIONS

ABSTRACT

Statistical estimation of mixed origin fishery composition using scale analysis has traditionally employed methods based on classification of the mixed fishery sample. Estimation studies using genetic data typically employ a different method, maximum likelihood, which does not classify the mixed sample. Until now, there has not been a direct comparison of the two methods because it was assumed that they could not be used with the same type of data. However, this difference in procedure is due primarily to computational convenience and it has recently been established that both methods can be used with both types of data. This study used only scale data and compared the performance of four classification estimators and the maximum likelihood estimator. For this comparison, a multipurpose simulation/bootstrap/analysis program and a "random composition" simulation program were developed using FORTRAN 77.

STATISTICAL METHODOLOGY FOR ESTIMATING COMPOSITION OF HIGH SEAS SALMONID MIXTURES USING SCALE ANALYSIS

INTRODUCTION

The 1986 International North Pacific Fisheries Commission (INPFC) Memorandum of Understanding (MOU) on research mandates an intensified research program that includes the use of scale pattern analysis to determine continental origins of salmonids in Convention waters. The MOU stipulates that Canadian, Japanese and U.S. researchers must base these studies, as much as possible, "on methodologies jointly agreed upon *a priori*." In recent INPFC-related scale pattern studies, Japanese and U.S. researchers have used different statistical methodologies for estimating composition of high seas salmonid mixtures (Myers et al. 1984 and Ito et al. 1986). This has resulted in considerable discussion among members and advisers of the INPFC Sub-committee on Salmon about the best procedures to use for composition estimation. The purpose of this paper is to compare several different estimation procedures to assess their relative performance with real scale data. The comparison is reinforced by known theoretical properties of the estimation procedures.

Composition estimation methods can be partitioned into two essential types depending on whether or not the mixed sample is classified fish by fish. Intuitively, it may seem that classification of the mixed sample is unavoidable, but this is not so. The method of maximum likelihood can be used to directly determine the composition most likely to give rise to the mixed sample.

Five estimation procedures were considered. With the exception of the maximum likelihood procedure mentioned above, they all classify the mixed sample. (In composition problems using genetic data, other non-classification methods have been used, e.g., linear regression. However, they have since been shown to be inappropriate on both theoretical and practical grounds (Mulligan et al., in press).) The five estimators are:

(1) Raw classification proportions
(2) Cook and Lord corrected classification estimates (Cook and Lord 1978)
(3) Cook's geometrically constrained classification estimates (Cook 1983)
(4) Millar's maximum likelihood constrained classification estimates (Millar 1987)
(5) Maximum likelihood estimates (Millar 1987)

Ito et al. (1986) used estimator (1), and Myers et al. (1984) used a constrained classification estimator that is similar to estimator (4) (see next section). To the best of my knowledge, the maximum likelihood estimator (5) has not previously been used in the application of composition estimation using scale analysis.

CLASSIFICATION ESTIMATORS

To calculate the classification-based estimates, (1) through (4), the first step is to determine a classification rule. This is typically done by submitting the standards to a statistical software package (BMDP, SAS, SPSS, etc.) after choosing the variables that are to be used for classification (see Davis 1987).

The form and complexity of the classification rule can vary; however, the simplest and hence most common form is the procedure of linear discriminant analysis, LDA. Simplicity is attained by making assumptions about the variables, i.e., that the variables are normally distributed and the covariance matrix of the variables is the same for each of the standards. Quadratic discriminant analysis dispenses with the assumption of common covariance matrices at the expense of additional computation. The normality assumption can also be dropped by resorting to non-parametric discriminant analysis, as done by Cook (1982), again at considerable computational expense. In this study, the simplest classification procedure, LDA, was used. In statistical terminology, the simplicity of the method gives more degrees of freedom because less parameters are being estimated, e.g., a common covariance matrix can be calculated much more accurately than individual covariance matrices for each standard (p. 69, Lachenbruch 1975). LDA is also relatively robust to departures from the assumption of normality (p. 504, Johnson and Wickern 1982).

The LDA provides a classification rule that is used to classify fish. In general, a "classification rule" is a set of functions that calculates for a given fish the likelihood of observing those scale measurements if that fish were from group i, for i=1, # groups. The fish is classified to the group for which this likelihood is highest. The proportion of mixed sample fish classified to each group gives the type (1) estimates. However, by using simple arguments, these raw proportions can be dismissed as viable estimators of composition because the probability of a mixed-sample fish being misclassified is not taken into account. The probability of misclassification depends on classification rule accuracy (this can be estimated) and the true composition of the mixed fishery (unknown).

Classification rule accuracy is summarized by its classification matrix. In the case of four groups, the classification matrix is a 4x4 matrix with the [i,j]'th element being the probability that a fish from group j is classified into group i (where i could equal j). The matrix has column sums of 1.0, but is not necessarily symmetrical. The classification matrix can be estimated by classifying the known origin fish from the standards. The [i,j]'th element is then simply the proportion of fish from the group j standard that classified to group i. An alternative, the "leave-one-out" procedure (Lachenbruch 1975), is preferable in practice, but for standards of reasonable size there will be little difference in the estimated classification matrices. Estimators (2)-(4) are given by correcting the raw classification proportions using the estimated classification matrix to produce valid estimators of the composition. Actually, the three corrected estimators differ only in one respect, and much of the time the three will give *exactly* the same composition estimates.

An estimate of composition is a vector of numbers, the elements being the estimated contribution of each group to the mixed fishery. For the vector to make sense, the estimated contributions must be non-negative and total 1.0. Estimator (2) is such that the estimates always total 1.0, but there is no assurance that they will be non-negative. Estimators (3) and (4) correct the raw proportions in the same way as (2) but with the proviso that they also constrain them to be non-negative. If the composition estimate given by (2) contains no negative values, then it will be the same as that given by (3) and (4). In the event of estimator (2) giving a negative estimate, some practitioners would repeat the composition analysis with that group removed (e.g., Myers et al. 1984). I considered this approach too difficult to automate and so did not include it in this study. Furthermore, the maximum likelihood constrained estimator (4) is based on achieving the same result, since it uses the classification matrix to assess the effect on the remaining composition estimates of setting to zero the negative estimate given by (2).

Since the Cook and Lord estimator (2) allows negative contribution estimates, it cannot truly be considered an estimator of composition and can be eliminated from consideration. The stock identification literature refers to (2) as an estimator and so, for want of a better word, I reluctantly continue to do the same. "Estimators" (1) and (2) are included in the output of the simulation programs because they are intermediate steps in the computation of the constrained classification estimators.

MAXIMUM LIKELIHOOD ESTIMATOR

The classification rule mentioned above provides the likelihood of a given mixed sample fish coming from each of the groups. In comparison, the method of maximum likelihood is based on calculating the likelihood for the entire mixed sample. With the same likelihood values used by the classification rule, it is possible to write down a likelihood function that, for any mixed fishery composition, gives the likelihood of observing the mixed sample. The maximum likelihood estimator of composition is found by maximizing the likelihood function over all possible compositions, i.e., all vectors of non-negative numbers that total 1.0.

Computing the maximum likelihood estimates is a non-linear maximization problem and it must be done numerically. A variety of algorithms available in optimization software packages, e.g., the NAG library, may be used (Pella and Milner 1987). Alternatively, as explained in Millar (1987), the EM (expectation-maximization) algorithm can be coded in a dozen lines of FORTRAN. The EM algorithm is extremely simple and the iterations are very fast in comparison to other algorithms, but it typically requires many more iterations to achieve convergence. It was the algorithm used in this study. Although Fournier et al. (1984) use a quasi-Newton maximization algorithm and Pella and Milner (1987) propose a iteratively reweighted least squares approach, it should be stressed that the maximum likelihood estimates will be the same in every case.

ESTIMATOR COMPARISON

The relative performance of the estimators can be deduced (at least in part) on theoretical grounds. Some theoretical considerations are summarized below, and more detail can be found in Millar (1987).

I. The maximum likelihood estimator (5) makes better use of the information in the mixed sample than the classification estimators. This is because classification of a fish simply indicates which group had the highest likelihood of that fish. If a fish has a relatively high likelihood for one group and relatively low likelihood for all other groups then its classification will probably be correct. On the other hand, if a fish has likelihoods of nearly equal magnitude for several groups, then its classification is of dubious value. The former fish is therefore more informative than the latter. This knowledge would be lost by the classification methods, since the classification matrix gives only "average" (over all fish in the group) misclassification probabilities and provides no indication as to how well any particular fish has been classified. In contrast, the maximum likelihood method uses all of the likelihood values directly.

II. Maximum likelihood estimators often exhibit some bias, which typically decreases very quickly with increasing sample size. The classification methods are slightly biased because the correction procedure requires the inverse of the estimated classification matrix. Even though the classification matrix can be estimated without bias, it does not follow that its inverse is unbiased.

III. The classification methods may be more robust to anomalies in the standards. Possible anomalies include the presence of outliers or gross violations in the assumptions of normally distributed variables with a covariance matrix common to all groups.

IV. With regard to the two constrained estimators, the Cook (1983) estimator (3) is purely a geometric constraining of estimator (2), and the constraining makes no reference to the actual composition estimation problem at hand. The maximum likelihood constrained estimator (4) of Millar (1987) further utilizes the estimated classification matrix to determine the effect on the composition estimates of correcting for any negative estimates yielded by estimator (2). Hence, the latter method should be better.

The two classes of estimators (classification and maximum likelihood) have their respective advantages and disadvantages; therefore, comparisons with real data are required to assess their relative performance in practice. For this purpose, two simulation programs were written in FORTRAN 77 for use on the University of Washington CDC Cyber 180-855 computer. The simulation programs seek to emulate the sampling randomness involved in practical application of the composition estimation problem. A set of standards is required as input to the programs. To simulate new standards and mixed samples, the given standards are sampled with replacement. In this way, the given standards are effectively defining the group populations (of infinite size). For example, to simulate creation of a size 200 standard from group i, the group i standard is sampled 200 times with replacement. When simulating creation of a mixed sample, the number of fish from each group is a random quantity with distribution depending on the size of the mixed sample and the mixed fishery composition. The programs randomly generate these numbers and sample from the given standards accordingly. Note that the programs do not attempt to simulate factors such as non-representative sampling.

In my terminology, a simulation means the process of estimating a composition using simulated standards and mixed sample as explained above. The programs perform a number of simulations, as specified in the control file, and I call this a simulation run. One hundred is a standard number of simulations for a run. The control file also includes the sizes of the simulated standards and mixed sample. This allows for investigation of the effect of standard and mixed sample sizes on both the bias (accuracy) and variability (precision) of the composition estimates. The user may optionally specify that instead of simulating new standards for each simulation, the input standards should be used. In this case, the programs treat the input standards as group populations, thus simulating the hypothetical case where the group populations are "known", i.e., the distribution of the measured variables is known exactly. This can be interpreted as corresponding to simulated standards of infinite size.

Relative performance of the estimators is a function of many things, including variables used, standard sizes, mixed sample size and the actual composition. To see that the actual composition plays a role, suppose that each of the groups gives a sizeable contribution to the mixed fishery population. Then it would be unlikely that estimator (2) would yield any negative contribution estimates, in which case estimators (2), (3) and (4) would all be the same. However, if one or more groups were of low contribution, then (2) is quite likely to yield negative estimates, and the three estimates will be different.

The first simulation program requires the user, in the control file, to specify the composition of the mixed fishery population (Appendix 1). For performance comparison purposes, the bias and variance of the simulated estimates are combined to produce a table of mean squared errors. The second simulation program differs from the first only in that the program randomly generates the actual composition to use for each simulation. That is, for a 100 simulation run, the program uses 100 different mixed fishery compositions. Thus, it is no longer possible to separate variance and bias, since they depend on the particular composition and the program only does one simulation for each composition. Instead, it records the absolute difference between actual and estimated composition. The measure of estimator performance used is the average (over the simulations) of the squared differences. Using the squared difference is in keeping with common statistical practice (e.g., "least squares") where the measure of error is the square of the difference or residual.

SIMULATION STUDY AND RESULTS

All of the simulations were with mixed fisheries comprised of four regional groups (see Myers et al. 1984). The regions were Asia, Western Alaska, Central Alaska and Southeastern Alaska/British Columbia (ASIA, WEST, CENT AND SEBC, respectively). Data for age 1.2 chinook from the 1971 and 1977 brood years were used. The 1971 standards are of size 161, 200, 122 and 195 respectively and the 1977 standard sizes are 200, 199, 199, and 198. Myers et al. used nine variables in their 1971 brood year analysis. The same nine variables were considered for use in the simulations, but one of the nine variables was of mixed type (both discrete and continuous components) and so was eliminated. (Such highly non-normal variables are not recommended for use in composition analyses; see Davis 1987.) The 1977 brood year analysis of Myers et al. used 16 variables, but four of these were eliminated for the same reason. No other checks were performed on any variables, the presumption being that these variables are of a "quality" typical of variables to be used in future analyses.

Simulation 1 shows the outcome of a 1971 brood year mixed fishery simulation with true composition of ASIA, WEST, CENT AND SEBC being 0.25, 0.65, 0.1 and 0.0, respectively. One hundred mixed samples, each of size 100, were simulated, and a graphical display of the composition estimates is given in Figure 1. The standards were not simulated in this run, so for each simulation the given standards were used and only the mixed sample was

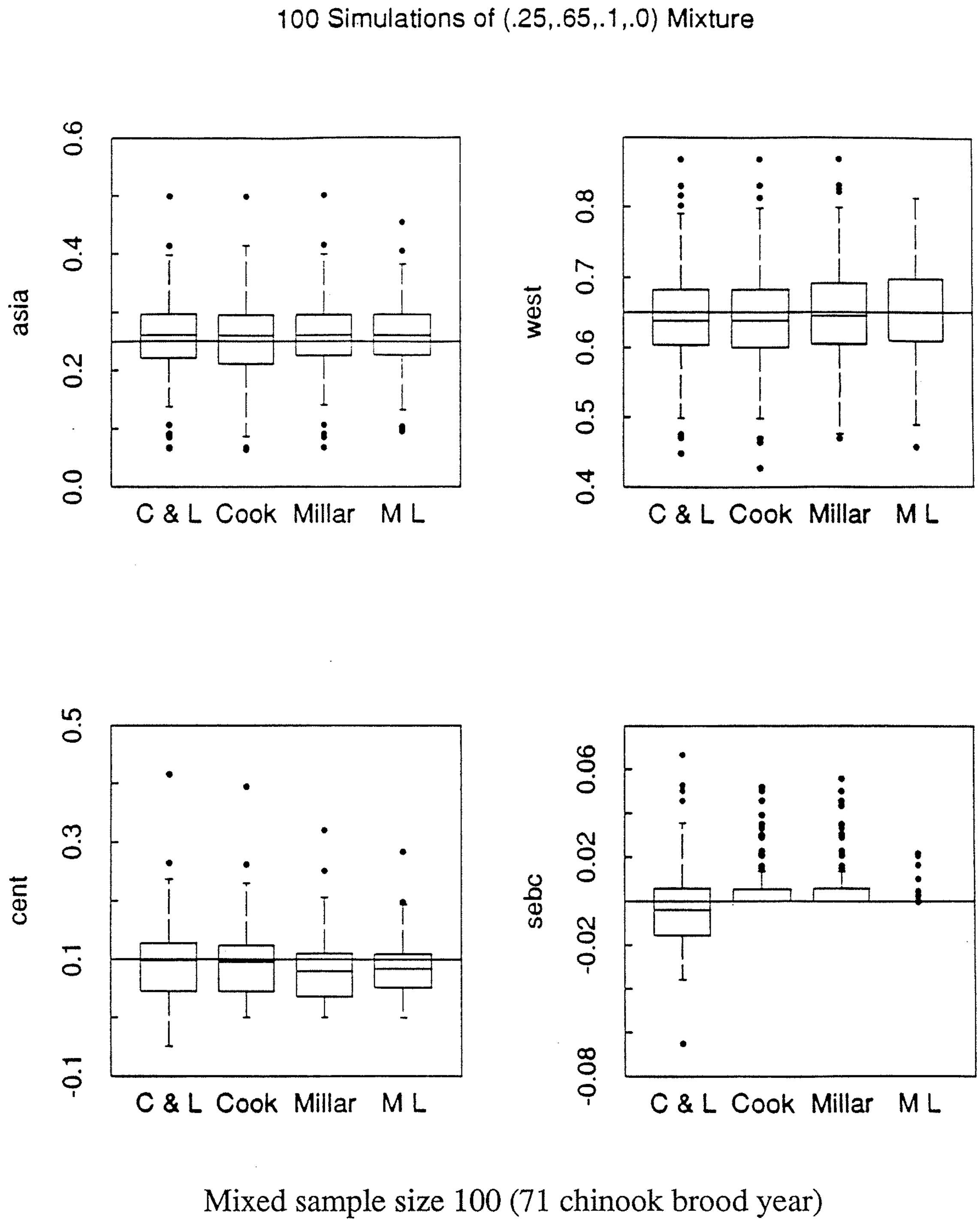

Figure 1. Boxplots of the estimated compositions from simulation 1. The middle line in each box is the median value and the top and bottom of the box are the upper quartile and lower quartile, respectively. The height of the box is the interquartile range. The "whiskers" that extend from the top and bottom of each box reach to the furthest value that is within $1\frac{1}{2}$ times the interquartile range. Any values beyond that are displayed as dots.

generated. Comparison of estimators (3)-(5) shows that the maximum likelihood estimator has lower mean square error for all four components (ASIA, WEST, CENT AND SEBC) of the composition vector. This is not always the case, but over a dozen further simulation runs with different true compositions it never failed to do better in at least two components. One may note that the raw classification proportions "estimate" did best for estimating the contribution of the Asian group—this is entirely fortuitous. Further simulation runs showed that for some mixture compositions the raw proportions were the worst estimator of the Asian contribution.

Four simulation runs (Simulations 2-5) were performed with the random composition program, two for each of the brood years. Within each brood year the two simulation runs used mixed sample sizes of 100 and 400, respectively. The standards were simulated in these runs, so in addition to creating a mixed sample for each simulation, the program was also creating new standards from the given (input) standards. Consequently, the program had to perform the linear discriminant analysis and calculate the classification matrix each time, thereby considerably increasing execution time from 17s for the Simulation 1 run to 55s for the run producing Simulation 2. The simulated standards were specified to be the same size as the given standards, thus mimicking the amount of sampling randomness present in constructing the given standards.

The maximum likelihood estimator (5) performs best in all four runs and the reduction in average squared error is typically 20 to 30 percent. Only in Simulation 5 did maximum likelihood not perform best at estimating all four components of the composition vector. Instead, the classification estimators were better at estimating the contribution from ASIA and WEST with respective decreases in error of 18% and 10%. However, the CENT and SEBC contributions were estimated better by maximum likelihood with improvements of 53% and 25%, respectively. The two constrained classification estimators had almost identical performance and I feel that additional comparison would be required to refer one over the other. As mentioned previously, (1) and (2) are not true estimators of composition, hence their poor performance.

It must be remembered that these simulation programs are not burdened by the many difficulties encountered in practical composition estimation, e.g., non-representative sampling, non-normal data. Different specifications of sample sizes, number of variables used and number of stock groups will affect the relative performance of the estimators also. However, my feeling is that the maximum likelihood estimator will perform best except when there are very extreme violations to the model assumptions, in which case it is unlikely that any estimator will give useful results.

DISCUSSION

The raw classification proportions were dismissed from consideration because they are inappropriate for composition estimation. The bias (a function of the classification matrix and the composition) of this "estimator" can be considerable. For a 1971 brood year composition consisting entirely of fish from the CENT region, the bias in estimating the contribution from CENT is 0.39 because the raw proportion estimate has expected value equal to the probability of correctly classifying fish from CENT, which is 0.61. The bias does not decrease with increased sample sizes since it is a consequence of the inherent deficiencies of this estimator.

The constrained classification procedures (Cook 1983 and Millar 1987) and maximum likelihood (Millar 1987) give legitimate estimates of composition. The simulations indicate that the two constrained classification estimators have nearly identical performance. In composition analyses with a greater number of stock groups, there is more chance of contribution estimates being subject to the non-negativity constraint, and the relative performance of the estimators may become more pronounced. The last table in Simulations 2-5 gives the average absolute difference between Cook's constrained classification estimator and the maximum likelihood estimator. The largest value, 0.04, is from Simulation 2 while in Simulation 5 the average absolute difference is as small as 0.01. The mean squared error table of Simulation 1 indicates that the variability of the composition estimates exceeds the magnitude of these differences, suggesting that the estimates are influenced more by sampling randomness than by the choice of which of the three estimators is used.

As a result of this work and discussions among Canadian, Japanese and U.S. scientists at meetings of the INPFC scale pattern working group and scale pattern analysis workshop (INPFC 1987a,b), the 1987 Sub-committee on Salmon recommended using either maximum likelihood or classification (with error correction and non-negativity constraints) as statistical procedures for estimating stock composition of high seas salmonid mixtures. This recommendation was endorsed by the Standing Committee of Biology and Research at the 1987 annual meeting of the INPFC. A FORTRAN program for performing routine analyses with maximum likelihood and classification estimators is provided in Appendix 2.

REFERENCES

Cook, R.C. and G.E. Lord. 1978. Identification of stocks of Bristol Bay sockeye salmon, *Oncorhynchus nerka*, by evaluating scale patterns with a polynomial discriminant method. Fish. Bull. 76:415-423.

Cook, R.C. 1982. Stock identification of sockeye salmon (*Oncorhynchus nerka*) with scale pattern recognition. Can. J. Fish. Aquat. Sci. 39: 611-617.

Cook, R.C. 1983. Simulation and application of stock composition estimators. Can. J. Fish. Aquat. Sci. 40:2113-2118.

Davis, N.D. 1987. Variable selection and performance of variable subsets in scale pattern analysis. (Document submitted to annual meeting of the International North Pacific Fisheries Commission, 1987) 47 pp. FRI-UW-8713. Fisheries Research Institute, University of Washington, Seattle.

Fournier, D.A., T.D. Beacham, B.E. Riddell and C.A. Busack. 1984. Estimating stock composition in mixed stock fisheries using morphometric, meristic, and electrophoretic characteristics. Can. J. Fish. Aquat. Sci. 41: 400-408.

International North Pacific Fisheries Commission. 1987a. Report of the scale pattern working group, 27 February 1987. International North Pacific Fisheries Commission, Vancouver, Canada.

International North Pacific Fisheries Commission. 1987b. Report of scale pattern analysis workshop, 22-23 October 1987. International North Pacific Fisheries Commission, Vancouver, Canada.

Ito, J., Y. Ishida and S. Ito. 1986. Further analysis of stock identification of chinook salmon in offshore waters in 1974. (Document submitted to the International North Pacific Fisheries Commission). 9 pp. Fisheries Agency of Japan, Tokyo.

Johnson, R.A. and D.W. Wickern. 1982. Applied multivariate statistical analysis. Prentice-Hall.

Lachenbruch, P.A. 1975. Discriminant analysis. Hafner Press.

Millar, R.B. 1987. Maximum likelihood estimation of mixed stock fishery composition. Can. J. Fish. Aquat. Sci. 44:583-590.

Mulligan, T.J., S. McKinnell and C.C.Wood. Uncertainty in stock composition estimates of oceanic steelhead trout using electrophoretic characteristics: Comments on a recent study. Can. J. Fish. Aquat. Sci. In Press.

Myers, K.W., D.E. Rogers, C.K. Harris, C.M. Knudsen, R.V. Walker and N.D. Davis. 1984. Origins of chinook salmon in the area of the Japanese mothership and landbased driftnet salmon fisheries in 1975-1981. (Document submitted to annual meeting of the International North Pacific Fisheries Commission, Vancouver, Canada, November 1984). 208 pp. Fisheries Research Institute, University of Washington, Seattle.

Pella, J.J., and G.B. Milner. 1987. Use of genetic marks in stock composition analysis. In N. Ryman and F.M. Utter [ed.]. Population genetics and applications to fisheries management. University of Washington Press, Seattle, WA.

SIMULATIONS

Simulation 1. Simulation with chinook 1971 brood year data. Four regional groups in mixture with composition (0.25, 0.65, 0.10, 0.00). Includes table of means and standard deviations for the eight variables used.

```
PROGRAM  HISEA.......EXECUTION DATE:  87/09/09.      16.03.50.

71 B.Y. CHINOOK.      8 NON L VARIABLES USED


FUNCTION OF THIS RUN IS.........SIMULATION
#STOCKS IN THE MODEL............ 4
THE STOCKS ARE..................ASIA  WEST  CENT  SEBC
#VARIABLES USED................. 8

STANDARD BEING RESAMPLED?.......N
MIXTURE BEING SIMULATED?........Y

RANDOM NUMBER GENERATOR SEED....  .61221670D+06
NUMBER OF RUNS REQUESTED?.......  100
SIZE OF SIMULATED MIXTURE.......  100
ACTUAL COMPOSITION IS...........    .250    .650    .100    .000

======================================================================
```

MEAN AND STANDARD DEVIATION TABLE

VAR	ASIA	WEST	CENT	SEBC
1	147.54 (13.0)	160.73 (13.1)	150.63 (13.4)	137.82 (11.9)
2	1049.4 (145.)	1158.3 (180.)	1185.1 (171.)	1225.5 (188.)
3	28.677 (2.83)	30.355 (3.09)	29.689 (3.15)	32.595 (3.62)
4	107.82 (13.7)	118.43 (15.2)	121.57 (19.2)	107.27 (13.6)
5	.62136E-01 (.930)-02	.54921E-01 (.972)-02	.59603E-01 (.126)-01	.57053E-01 (.103)-01
6	1656.0 (216.)	1679.5 (230.)	1644.9 (188.)	1501.9 (180.)
7	.80880 (.255)-01	.81719 (.277)-01	.80797 (.312)-01	.78783 (.352)-01
8	5750.1 (724.)	6492.8 (692.)	5884.8 (684.)	6173.1 (679.)

```
======================================================================


THE SIZES OF THE STANDARDS ARE    161    200    122    195


TABLE OF COMPOSITION ESTIMATE MEANS.  NUMBER OF RUNS =  100
```

Simulation 1 - cont'd.

	RAW	COOK & LORD	COOK CONSTRAINED	MILLAR CONSTRAINED	MAXIMUM LIKELIHOOD
ASIA	.2804	.2596	.2562	.2604	.2607
WEST	.5680	.6469	.6435	.6523	.6532
CENT	.1361	.0960	.0945	.0812	.0853
SEBC	.0155	-.0026	.0058	.0060	.0008

TABLE OF COMPOSITION ESTIMATE STANDARD DEVIATIONS OVER THE 100 RUNS

	RAW	COOK & LORD	COOK CONSTRAINED	MILLAR CONSTRAINED	MAXIMUM LIKELIHOOD
ASIA	.0419	.0726	.0729	.0724	.0616
WEST	.0525	.0819	.0827	.0814	.0714
CENT	.0370	.0723	.0674	.0596	.0495
SEBC	.0114	.0202	.0116	.0118	.0036

TABLE OF MEAN SQUARED ERRORS

	RAW	COOK & LORD	COOK CONSTRAINED	MILLAR CONSTRAINED	MAXIMUM LIKELIHOOD
ASIA	.0518	.0733	.0732	.0731	.0625
WEST	.0974	.0819	.0830	.0814	.0714
CENT	.0517	.0724	.0676	.0625	.0517
SEBC	.0192	.0204	.0130	.0133	.0037

16.04.19. 17.081 CP SECONDS EXECUTION TIME.

Simulation 2. Simulation with chinook 1971 brood year data. Random compositions and mixed sample size 100. Includes table of means and standard deviations for the eight variables used.

```
OUTPUT FROM RANDOM COMPOSITION SIMULATION PROGRAM
      EXECUTION DATE  88/01/09.      15.36.16.

71 B.Y. CHINOOK.     8 NON L VARIABLES USED


FUNCTION OF THIS RUN IS.........SIMULATION
#STOCKS IN THE MODEL............ 4
THE STOCKS ARE..................ASIA  WEST  CENT  SEBC
#VARIABLES USED................. 8

STANDARD BEING RESAMPLED?.......Y
RESAMPLED STANDARD SIZES........   161   200   122   195
MIXTURE BEING SIMULATED?........Y

RANDOM NUMBER GENERATOR SEED....  .33333930D+06
NUMBER OF RUNS REQUESTED?.......  100
SIZE OF SIMULATED MIXTURE.......  100


======================================================================
```

MEAN AND STANDARD DEVIATION TABLE

VAR	ASIA	WEST	CENT	SEBC
1	147.54 (13.0)	160.73 (13.1)	150.63 (13.4)	137.82 (11.9)
2	1049.4 (145.)	1158.3 (180.)	1185.1 (171.)	1225.5 (188.)
3	28.677 (2.83)	30.355 (3.09)	29.689 (3.15)	32.595 (3.62)
4	107.82 (13.7)	118.43 (15.2)	121.57 (19.2)	107.27 (13.6)
5	.62136E-01 (.930)-02	.54921E-01 (.972)-02	.59603E-01 (.126)-01	.57053E-01 (.103)-01
6	1656.0 (216.)	1679.5 (230.)	1644.9 (188.)	1501.9 (180.)
7	.80880 (.255)-01	.81719 (.277)-01	.80797 (.312)-01	.78783 (.352)-01
8	5750.1 (724.)	6492.8 (692.)	5884.8 (684.)	6173.1 (679.)

```
======================================================================


THE SIZES OF THE STANDARDS ARE    161   200   122   195
```

Simulation 2 - cont'd.

AVERAGE OF THE SQUARED ERRORS. NUMBER OF RUNS = 100

	RAW	COOK & LORD	COOK CONSTRAINED	MILLAR CONSTRAINED	MAXIMUM LIKELIHOOD
ASIA	.0093	.0057	.0046	.0048	.0036
WEST	.0077	.0043	.0038	.0037	.0029
CENT	.0127	.0074	.0065	.0064	.0043
SEBC	.0055	.0037	.0035	.0034	.0025

THE AVERAGE ABSOLUTE DIFFERENCE BETWEEN COOK CONSTRAINED AND MAXIMUM LIKELIHOOD ESTIMATES

ASIA	.0325
WEST	.0322
CENT	.0411
SEBC	.0352

15.38.46. 55.273 CP SECONDS EXECUTION TIME.

Simulation 3. Simulation with chinook 1971 brood year data. Random compositions and mixed sample size 400.

```
OUTPUT FROM RANDOM COMPOSITION SIMULATION PROGRAM
      EXECUTION DATE  88/01/09.      15.45.03.

71 B.Y. CHINOOK.      8 NON L VARIABLES USED


FUNCTION OF THIS RUN IS.........SIMULATION
#STOCKS IN THE MODEL............ 4
THE STOCKS ARE..................ASIA   WEST   CENT   SEBC
#VARIABLES USED................. 8

STANDARD BEING RESAMPLED?.......Y
RESAMPLED STANDARD SIZES........  161    200    122    195
MIXTURE BEING SIMULATED?........Y

RANDOM NUMBER GENERATOR SEED....  .55555550D+06
NUMBER OF RUNS REQUESTED?.......  100
SIZE OF SIMULATED MIXTURE.......  400


THE SIZES OF THE STANDARDS ARE    161    200    122    195


AVERAGE OF THE SQUARED ERRORS.  NUMBER OF RUNS =  100
```

	RAW	COOK & LORD	COOK CONSTRAINED	MILLAR CONSTRAINED	MAXIMUM LIKELIHOOD
ASIA	.0057	.0023	.0022	.0022	.0016
WEST	.0046	.0019	.0016	.0016	.0013
CENT	.0123	.0035	.0034	.0034	.0021
SEBC	.0038	.0016	.0016	.0017	.0014

THE AVERAGE ABSOLUTE DIFFERENCE BETWEEN COOK CONSTRAINED AND MAXIMUM LIKELIHOOD ESTIMATES

ASIA	.0268
WEST	.0219
CENT	.0352
SEBC	.0271

```
15.50.29.     80.382 CP SECONDS EXECUTION TIME.
```

Simulation 4. Simulation with chinook 1977 brood year data. Random compositions and mixed sample size 100. Includes table of means and standard deviations for the twelve variables used.

```
OUTPUT FROM RANDOM COMPOSITION SIMULATION PROGRAM
      EXECUTION DATE  87/12/10.     19.06.16.

77 B.Y. CHINOOK.     12 VARIABLE STANDARD USED


FUNCTION OF THIS RUN IS.........SIMULATION
#STOCKS IN THE MODEL............ 4
THE STOCKS ARE..................ASIA  WEST  CENT  SEBC
#VARIABLES USED.................12

STANDARD BEING RESAMPLED?.......Y
RESAMPLED STANDARD SIZES........   200   199   199   198
MIXTURE BEING SIMULATED?........Y

RANDOM NUMBER GENERATOR SEED....  .77777770D+06
NUMBER OF RUNS REQUESTED?.......  100
SIZE OF SIMULATED MIXTURE.......  100


======================================================================
```

MEAN AND STANDARD DEVIATION TABLE

VAR	ASIA	WEST	CENT	SEBC
1	.11524 (.159)-01	.81654E-01 (.134)-01	.83650E-01 (.136)-01	.73850E-01 (.125)-01
2	151.17 (14.5)	158.46 (12.1)	149.88 (11.4)	136.04 (11.7)
3	1080.7 (194.)	1161.6 (165.)	1181.9 (166.)	1270.4 (195.)
4	114.73 (17.2)	115.82 (13.7)	125.90 (17.5)	101.60 (14.0)
5	26.795 (2.57)	34.864 (3.91)	32.834 (4.28)	35.732 (4.41)
6	1713.1 (241.)	1612.5 (219.)	1570.0 (165.)	1485.8 (165.)
7	.11368 (.177)-01	.83629E-01 (.121)-01	.81575E-01 (.129)-01	.72058E-01 (.132)-01
8	.65386E-01 (.115)-01	.56892E-01 (.891)-02	.62944E-01 (.113)-01	.56631E-01 (.102)-01
9	164.80 (18.3)	172.55 (14.5)	158.22 (12.4)	149.01 (13.2)
10	1359.3 (472.)	1791.5 (220.)	1434.2 (283.)	1320.1 (165.)
11	.94476E-01	.67457E-01	.76155E-01	.76372E-01

Simulation 4 - cont'd.

	(.172)-01	(.122)-01	(.149)-01	(.125)-01
12	.10639 (.154)-01	.74020E-01 (.146)-01	.83452E-01 (.142)-01	.76305E-01 (.117)-01

THE SIZES OF THE STANDARDS ARE 200 199 199 198

AVERAGE OF THE SQUARED ERRORS. NUMBER OF RUNS = 100

	RAW	COOK & LORD	COOK CONSTRAINED	MILLAR CONSTRAINED	MAXIMUM LIKELIHOOD
ASIA	.0029	.0030	.0029	.0029	.0027
WEST	.0053	.0037	.0035	.0034	.0026
CENT	.0061	.0039	.0036	.0035	.0022
SEBC	.0034	.0024	.0023	.0023	.0018

THE AVERAGE ABSOLUTE DIFFERENCE BETWEEN COOK CONSTRAINED AND MAXIMUM LIKELIHOOD ESTIMATES

ASIA	.0173
WEST	.0283
CENT	.0300
SEBC	.0181

19.08.15. 83.456 CP SECONDS EXECUTION TIME.

Simulation 5. Simulation with chinook 1977 brood year data. Random compositions and mixed sample size 400.

```
OUTPUT FROM RANDOM COMPOSITION SIMULATION PROGRAM
      EXECUTION DATE  87/12/10.      19.20.54.

77 B.Y. CHINOOK.      12 VARIABLE STANDARD USED


FUNCTION OF THIS RUN IS.........SIMULATION
#STOCKS IN THE MODEL............ 4
THE STOCKS ARE..................ASIA   WEST   CENT   SEBC
#VARIABLES USED.................12

STANDARD BEING RESAMPLED?.......Y
RESAMPLED STANDARD SIZES........   200    199    199    198
MIXTURE BEING SIMULATED?........Y

RANDOM NUMBER GENERATOR SEED....  .34345800D+05
NUMBER OF RUNS REQUESTED?.......  100
SIZE OF SIMULATED MIXTURE.......  400



THE SIZES OF THE STANDARDS ARE    200    199    199    198


AVERAGE OF THE SQUARED ERRORS.  NUMBER OF RUNS =  100
```

	RAW	COOK & LORD	COOK CONSTRAINED	MILLAR CONSTRAINED	MAXIMUM LIKELIHOOD
ASIA	.0020	.0009	.0009	.0009	.0011
WEST	.0035	.0009	.0009	.0009	.0010
CENT	.0048	.0015	.0015	.0015	.0007
SEBC	.0025	.0008	.0008	.0008	.0006

THE AVERAGE ABSOLUTE DIFFERENCE BETWEEN COOK CONSTRAINED AND MAXIMUM LIKELIHOOD ESTIMATES

ASIA	.0141
WEST	.0172
CENT	.0181
SEBC	.0125

19.23.28.   103.908 CP SECONDS EXECUTION TIME.

APPENDIX 1.

LISTING OF MULTIPURPOSE PROGRAM

```
      PROGRAM HISEA(TAPE7,TAPE8,TAPE9,OUTPUT,TAPE6=OUTPUT,TAPE2)
C
C COPYRIGHT: THIS PROGRAM WAS WRITTEN BY  RUSSELL MILLAR (DEPT OF STAT,
C 16 OCT 87  GN-22) FOR F.R.I. USE.  F.R.I. RESERVES THE RIGHT TO
C            DISTRIBUTE THIS PROGRAM AT THEIR DISCRETION.
C
C MODIFIED  10 DEC 87
C
C.....................................................................
C TAPE7 CONTAINS ALL OF THE STANDARDS APPENDED TOGETHER WITH END OF
C   FILE MARKERS SEPARATING THEM.
C TAPE8 IS THE CONTROL FILE - SPECIFIES PURPOSE OF RUN (ANALYSIS,
C   SIMULATION OR BOOTSTRAP) AND #POPNS, #VARIABLES   ETC ETC.
C TAPE9 CONTAINS THE MIXED SAMPLE FILE.  IT IS NOT REQUIRED BY A
C   SIMULATION RUN.
C.....................................................................
C SOME VARIABLES USED INCLUDE:
C   NP=NUMBER OF STOCK GROUPS (POPULATIONS) ON TAPE7
C   NV=NUMBER OF VARIABLES
C   NSTD=TOTAL NUMBER OF FISH OVER ALL THE STANDARDS
C   NMIX=SIZE OF MIXED SAMPLE ON TAPE9
C   N=SIZE OF SIMULATED MIXED SAMPLE
C*********************************************************************
C************PRESET LIMITS FOR THIS VERSION OF SOURCE CODE ARE**********
C************ NP<=8   NV<=16  NSTD<=1400   NMIX<=1400   N<=2000 ********
C*********************************************************************
C THE INDEX VARIABLES IN  DO  LOOPS ARE PREDOMINANTLY USED IN THE
C   FOLLOWING WAY:
C   J  INDEXS THE STOCK GROUP     J=1,...,NP
C   G  INDEXS FISH IN THE STDS OR MIXED SAMPLE    G=1,...,NSTD OR NMIX
C   L  INDEXS THE VARIABLES    L=1,...,NV
C SOME ARRAYS USED INCLUDE:
C   STSIZE(J)=SIZE OF STANDARD FOR STOCK GROUP J
C   GSIZE(J)=SIZE OF RESAMPLED STANDARD.  OFTEN THE SAME AS STSIZE(J).
C   FREQ(G)=FREQUENCY OF FISH  G  IN THE RESAMPLED DATA.
C   CLASS(G)=THE CLASS FISH  G  IS CLASSIFIED TO.
C   VARBLE(G,L)=THE L'TH MEASUREMENT ON THE G'TH FISH.
C   THETA(J,K)=THE CONTRIBUTION ESTIMATES
C             THETA(*,1)=RAW CLASSIFICATION ESTIMATES
C             THETA(*,2)=COOK AND LORD CORRECTED ESTIMATES
C             THETA(*,3)=COOK CONSTRAINED CORRECTED ESTIMATES
C             THETA(*,4)=MILLAR CONSTRAINED CORRECTED ESTIMATES
C              NB: THE LAST THREE TYPES OF ESTIMATES WILL BE IDENTICAL
C                  WHEN THE COOK AND LORD CORRECTED ESTIMATES ARE ALL
C                  NON NEGATIVE.
C             THETA(*,5)=DIRECT MAXIMUM LIKELIHOOD ESTIMATES
C   ACTUAL()=TRUE VECTOR OF CONTRIBUTIONS FOR SIMULATION PURPOSES
C*********************************************************************
C THE MAIN PROGRAM READS THE CONTROL FILE (TAPE8) AND CALLS THE
C APPROPRIATE SUBPROGRAMS DEPENDING ON THE PURPOSE OF THE RUN.  IT FIRST
C DISTINGUISHES CASES ACCORDING TO WHETHER THE STANDARDS ARE BEING
C RESAMPLED.
C*********************************************************************
C.....................................................................
      INTEGER NP,N,NV,NSAMPS,GSIZE(8)
      REAL ACTUAL(8),X(1400,8),STDVAR(1400,16),ORDERS(2001)
      REAL THSUM(8,5),THSSQ(8,5),COVAR(8,8)
      CHARACTER TYPE*10,OPT(2),POPNAM(8)*6
      REAL SEED
      COMMON THSUM,THSSQ,COVAR
```

```
      COMMON /UNIFORM/ ORDERS
      COMMON /OPTIONS/ TYPE,OPT
C     CALL CONNEC(5)
      DO 1 J=1,8
        DO 2 K=1,5
          THSUM(J,K)=0.0
          THSSQ(J,K)=0.0
    2   CONTINUE
        DO 3 K=1,8
          COVAR(J,K)=0.0
    3   CONTINUE
    1 CONTINUE
      CALL READ8(NP,POPNAM,NV,N,NSAMPS,ACTUAL,GSIZE,SEED)
C INITIALIZE THE  CYBER  RANDOM NUMBER GENERATOR WITH THE SEED VALUE
C GIVEN IN THE CONTROL FILE
      CALL RANSET(SEED)
      IF(OPT(1).EQ.'N') THEN
C........STANDARD NOT BEING RESAMPLED
        CALL BASEFIX(NP,NV,N,NSAMPS,ACTUAL,X,STDVAR,POPNAM)
      ELSE
        CALL NUBASES(NP,NV,N,NSAMPS,GSIZE,ACTUAL,X,STDVAR,POPNAM)
      END IF
      CALL RESULTS(NP,NSAMPS,ACTUAL,TYPE,POPNAM)
      STOP
      END
C**********************************************************************
      SUBROUTINE READ8(NP,POPNAM,NV,N,NSAMPS,ACTUAL,GSIZE,SEED)
C THIS SUBROUTINE READS THE CONTROL FILE (TAPE8)
      INTEGER NP,NV,N,NSAMPS,GSIZE(8)
      REAL ACTUAL(8),ACTSUM
      REAL SEED
      CHARACTER TYPE*10,STD*3,MIX*3,OPT(2),POPNAM(8)*6,TITLE*80
      CHARACTER *10 YMD,HOUR
      COMMON /OPTIONS/ TYPE,OPT

      CALL DATE(YMD)
      CALL TIME(HOUR)
      WRITE(6,777) YMD,HOUR

      READ(8,*) TITLE
      WRITE(6,720) TITLE
      READ(8,*) TYPE
      WRITE(6,721) TYPE
      IF(TYPE(1:1).NE.'A'.AND.TYPE(1:1).NE.'B'.AND.TYPE(1:1).NE.'S')
     . THEN
        PRINT *,'LINE 2 OF CONTROL FILE MUST SPECIFY  ANALYSIS ',
     .          'BOOTSTRAP OR SIMULATION'
        STOP
      END IF
      READ(8,*) NP,(POPNAM(J),J=1,NP)
      WRITE(6,722) NP
      WRITE(6,7225) (POPNAM(J),J=1,NP)
      IF(NP.LT.2.OR.NP.GT.8) THEN
        PRINT *,'NUMBER OF STOCKS IS OUT OF BOUNDS.  2 - 8'
        STOP
      END IF
      READ(8,*) NV
      WRITE(6,723) NV
      IF(NV.LT.1.OR.NV.GT.16) THEN
        PRINT *,'NUMBER OF VARIABLES IS OUT OF BOUNDS,  1 - 16'
```

```
        STOP
      END IF
      IF(TYPE(1:1).EQ.'A') THEN
        OPT(1)='N'
        OPT(2)='N'
        RETURN
      END IF
      READ(8,*) STD,OPT(1)
      WRITE(6,701) OPT(1)
      IF(STD.NE.'STD'.OR..NOT.(OPT(1).EQ.'Y'.OR.OPT(1).EQ.'N')) THEN
        PRINT *,'INVALID INPUT ON  STD LINE OF TAPE8 - CONSULT PROGRAM',
     .          ' DOCUMENTATION'
        STOP
      END IF
      READ(8,*) (GSIZE(J),J=1,NP)
      IF(OPT(1).EQ.'Y') THEN
        DO 8 J=1,NP
          IF(GSIZE(J).LT.1.0) THEN
            PRINT *,'SIZE OF RESAMPLED STANDARD FOR STOCK ',J,' IS NOT'
            PRINT *,'POSITIVE.   PROGRAM STOPPED'
            STOP
          END IF
8       CONTINUE
        WRITE(6,7014) (GSIZE(J),J=1,NP)
      END IF
      READ(8,*) MIX,OPT(2)
      IF(MIX.NE.'MIX'.OR..NOT.(OPT(2).EQ.'Y'.OR.OPT(2).EQ.'N')) THEN
        PRINT *,'INVALID INPUT ON  MIX LINE OF TAPE8 - CONSULT PROGRAM',
     .          ' DOCUMENTATION'
        STOP
      END IF
      IF(TYPE(1:1).EQ.'B') THEN
        WRITE(6,7015) OPT(2)
      ELSE
        WRITE(6,7016) OPT(2)
      END IF
      IF(TYPE(1:1).EQ.'S'.AND.OPT(2).EQ.'N') THEN
        WRITE(6,702)
        STOP
      END IF
      IF(OPT(1).EQ.'N'.AND.OPT(2).EQ.'N') THEN
        WRITE(6,706) TYPE
        STOP
      END IF
      READ(8,*) SEED
      IF(SEED.LT.2.0**7.OR.SEED.GT.2.0**25) THEN
        PRINT *,'SEED VALUE ',SEED,' OUT OF BOUNDS'
        SEED=1.234567E+05
        PRINT *,'SEED SET TO ',SEED
      END IF
      WRITE(6,705) SEED
      READ(8,*) NSAMPS
      WRITE(6,703) NSAMPS
      IF(NSAMPS.LT.1.OR.NSAMPS.GT.1000) THEN
        PRINT *,'NUMBER OF RESAMPLES REQUESTED IS OUT OF BOUNDS'
        STOP
      END IF
      IF(TYPE(1:1).EQ.'B') RETURN
      READ(8,*) N,(ACTUAL(J),J=1,NP)
      WRITE(6,704) N,(ACTUAL(J),J=1,NP)
```

```
      IF(N.LT.1) THEN
        PRINT *,'SIZE OF SIMULATED MIXTURE IS INVALID'
        STOP
      END IF
      ACTSUM=0.0
      DO 1 J=1,NP
        IF(ACTUAL(J).LT.0.0) THEN
          PRINT *,'NEGATIVE CONTRIBUTION GIVEN IN TAPE8????????'
          STOP
        END IF
        ACTSUM=ACTSUM+ACTUAL(J)
    1 CONTINUE
      IF(ACTSUM.LT.0.999.OR.ACTSUM.GT.1.001) THEN
        PRINT *,'TOTAL CONTRIBUTION IS ',ACTSUM,'<>1.000??????'
        STOP
      END IF
      RETURN
  777 FORMAT(' PROGRAM  HISEA.......EXECUTION DATE: ',A10,4X,A10)
  720 FORMAT(/1X,A80//)
  721 FORMAT(1X,'FUNCTION OF THIS RUN IS.........',A10)
  722 FORMAT(1X,'#STOCKS IN THE MODEL............',I2)
 7225 FORMAT(1X,'THE STOCKS ARE..................',8A6)
  723 FORMAT(1X,'#VARIABLES USED.................',I2/)
  701 FORMAT(1X,'STANDARD BEING RESAMPLED?.......',A1)
 7014 FORMAT(1X,'RESAMPLED STANDARD SIZES........',8I6)
 7015 FORMAT(1X,'MIXTURE BEING RESAMPLED?........',A1/)
 7016 FORMAT(1X,'MIXTURE BEING SIMULATED?........',A1/)
  702 FORMAT(1X,'MUST GENERATE MIXTURE WHEN SIMULATING - EXPECTED ',
     .          'TO SEE ''MIX''  ''Y''  ON LINE 7 OF TAPE8')
  703 FORMAT(1X,'NUMBER OF RUNS REQUESTED?.......',I5)
  704 FORMAT(1X,'SIZE OF SIMULATED MIXTURE.......',I5/,
     .       1X,'ACTUAL COMPOSITION IS...........',8F8.3)
  705 FORMAT(1X,'RANDOM NUMBER GENERATOR SEED....',D15.8)
  706 FORMAT(1X,A10,' IS IMPOSSIBLE SINCE NEITHER STANDARDS OR MIXTURE'/
     .       'ARE BEING RESAMPLED OR SIMULATED.')
      END
C**********************************************************************
      SUBROUTINE BASEFIX(NP,NV,N,NSAMPS,ACTUAL,X,STDVAR,PN)
C IF THIS ROUTINE CALLED THEN THE STANDARDS ARE NOT BEING RESAMPLED.
C NOTE: THE MATRIX OF MEASUREMENT LIKELIHOODS  X  CORRESPONDING TO THE
C STANDARD IS CALCULATED IN CLASFY (CALLED FROM STDUP).  IT WILL
C BE OVERWRITTEN IF DOING AN ANALYSIS OR BOOTSTRAP.
      INTEGER NP,NV,N,NMIX,NSAMPS,NSTD,STSIZE(8),CLASS(1400),G,SEVEN
      INTEGER FREQ(1400),GSIZE(8),NINE,ONE,DUMMY(8)
      REAL ACTUAL(NP),COEFS(8,17),PHI(8,8),X(1400,8),STDVAR(1400,16)
      CHARACTER TYPE*10,PN(NP)*6
      COMMON /OPTIONS/ TYPE
      SEVEN=7
      NINE=9
      ONE=1
      CALL READIN(NP,NV,STSIZE,NSTD,STDVAR,SEVEN,PN)
      WRITE(6,700) (STSIZE(I),I=1,NP)
      DO 800 G=1,NSTD
  800   FREQ(G)=1
      DO 801 J=1,NP
  801   GSIZE(J)=STSIZE(J)
      CALL STDUP(NP,NV,NSTD,STSIZE,GSIZE,FREQ,COEFS,CLASS,PHI,X,STDVAR)
      IF(TYPE(1:1).EQ.'S') THEN
        CALL SIMLATE(NP,N,NSAMPS,ACTUAL,NSTD,STSIZE,CLASS,PHI,X)
      ELSE
```

```
      CALL READIN(ONE,NV,DUMMY,NMIX,STDVAR,NINE,PN)
      WRITE(6,701) NMIX
      DO 802 G=1,NMIX
802     FREQ(G)=1
      CALL CLASFY(NP,NV,NMIX,FREQ,CLASS,COEFS,STDVAR,X)
      CALL REALDAT(NP,NSAMPS,NMIX,CLASS,PHI,X)
    END IF
    RETURN
700 FORMAT(//1X,'THE SIZES OF THE STANDARDS ARE ',8I6)
701 FORMAT(/1X,'MIXED SAMPLE HAS ',I5,' FISH')
    END
C**********************************************************************
    SUBROUTINE NUBASES(NP,NV,N,NSAMPS,GSIZE,ACTUAL,X,STDVAR,PN)
C IF THIS ROUTINE IS CALLED THEN THE STANDARDS ARE BEING RESAMPLED.
    INTEGER NP,NV,N,NSAMPS,NSTD,STSIZE(8),CLASS(1400),BFREQ(1400),G,
   .          NINE,SEVEN,ONE,LOOP,DUMMY(8),MFREQ(1400),GSIZE(NP)
    REAL ACTUAL(NP),COEFS(8,17),PHI(8,8),X(1400,8),STDVAR(1400,16),
   .      MIXVAR(1400,16)
    CHARACTER TYPE*10,PN(NP)*6
    COMMON /OPTIONS/ TYPE
    NINE=9
    SEVEN=7
    ONE=1
    CALL READIN(NP,NV,STSIZE,NSTD,STDVAR,SEVEN,PN)
    WRITE(6,700)  (STSIZE(J),J=1,NP)
    IF(TYPE(1:1).NE.'S') THEN
      CALL READIN(ONE,NV,DUMMY,NMIX,MIXVAR,NINE,PN)
      WRITE(6,709) NMIX
    END IF
    DO 1 LOOP=1,NSAMPS
      CALL RESAMP(NP,NSTD,GSIZE,STSIZE,BFREQ)
      CALL STDUP(NP,NV,NSTD,STSIZE,GSIZE,BFREQ,COEFS,CLASS,PHI,X,
   .             STDVAR)
      IF(TYPE(1:1).EQ.'S') THEN
        CALL SIMLATE(NP,N,ONE,ACTUAL,NSTD,STSIZE,CLASS,PHI,X)
      ELSE
        DO 800 G=1,NMIX
800       MFREQ(G)=1
        CALL CLASFY(NP,NV,NMIX,MFREQ,CLASS,COEFS,MIXVAR,X)
        CALL REALDAT(NP,ONE,NMIX,CLASS,PHI,X)
      END IF
  1 CONTINUE
    RETURN
700 FORMAT(//1X,'THE SIZES OF THE STANDARDS ARE ',8I6,)
709 FORMAT(/1X,'MIXED SAMPLE HAS ',I5,' FISH')
    END
C**********************************************************************
    SUBROUTINE SIMLATE(NP,N,NSAMPS,ACT,NSTD,STSIZE,CLASS,PHI,X)
C ARRAYS  X  AND  CLASS CONTAIN LIKELIHOOD AND CLASSIFICATION VALUES
C FOR EVERY FISH IN THE STANDARDS
    INTEGER NP,N,NSAMPS,STSIZE(NP),NSTD,FREQ(1400),CLASS(NSTD),EIGHT,
   .        EST
    REAL ACT(NP),THETA(8,5),PHI(8,8),X(1400,8)
    DOUBLE PRECISION PHIINV(8,8),DET
    EIGHT=8
    DO 801 J=1,NP
      DO 801 JJ=1,NP
801     PHIINV(J,JJ)=DBLE(PHI(J,JJ))
    CALL MXINT(PHIINV,NP,EIGHT,DET)
    IF(DET.EQ.0.0) THEN
```

```
          PRINT *,'THE CLASSIFICATION MATRIX IS SINGULAR!'
          PRINT *,'-NO UNIQUE COMPOSITION ESTIMATE IS POSSIBLE - ABORT'
          STOP
        END IF
        DO 10 LOOP=1,NSAMPS
          CALL SIMMIX(NP,N,NSTD,STSIZE,FREQ,ACT)
          CALL COOKLRD(NP,N,NSTD,FREQ,CLASS,THETA,PHIINV)
          CALL MILLARC(NP,N,THETA,PHI,TYPE)
          CALL ML(NP,N,NSTD,FREQ,THETA,X,TYPE)
          CALL ESTOUT(NP,THETA)
   10   CONTINUE
        RETURN
        END
C***********************************************************************
        SUBROUTINE REALDAT(NP,NSAMPS,NMIX,CLASS,PHI,X)
C ARRAYS  X  AND  CLASS CONTAIN LIKELIHOOD AND CLASSIFICATION VALUES
C FOR EVERY FISH IN THE MIXED SAMPLE
        INTEGER NP,N,NSAMPS,NMIX,G,CLASS(NMIX),FREQ(1400),DUMMY1(1),
       .        DUMMY2(1),EIGHT,ONE
        REAL PHI(8,8),X(1400,8),THETA(8,5)
        CHARACTER TYPE*10,OPT(2)
        DOUBLE PRECISION PHIINV(8,8),DET
        COMMON /OPTIONS/ TYPE,OPT
        EIGHT=8
        ONE=1
        DO 801 J=1,NP
          DO 801 JJ=1,NP
  801       PHIINV(J,JJ)=DBLE(PHI(J,JJ))
        CALL MXINT(PHIINV,NP,EIGHT,DET)
        IF(DET.EQ.0.0) THEN
          PRINT *,'THE CLASSIFICATION MATRIX IS SINGULAR!'
          PRINT *,'-NO UNIQUE COMPOSITION ESTIMATE IS POSSIBLE - ABORT'
          STOP
        END IF
        IF(OPT(2).EQ.'N') THEN
          DO 800 G=1,NMIX
  800       FREQ(G)=1
          N=NMIX
          CALL COOKLRD(NP,N,NMIX,FREQ,CLASS,THETA,PHIINV)
          CALL MILLARC(NP,N,THETA,PHI,TYPE)
          CALL ML(NP,N,NMIX,FREQ,THETA,X,TYPE)
          CALL ESTOUT(NP,THETA)
        ELSE
          DUMMY1(1)=NMIX
          DUMMY2(1)=NMIX
          N=NMIX
          DO 20 LOOP=1,NSAMPS
            CALL RESAMP(ONE,NMIX,DUMMY1,DUMMY2,FREQ)
            CALL COOKLRD(NP,N,NMIX,FREQ,CLASS,THETA,PHIINV)
            CALL MILLARC(NP,N,THETA,PHI,TYPE)
            CALL ML(NP,N,NMIX,FREQ,THETA,X,TYPE)
            CALL ESTOUT(NP,THETA)
   20     CONTINUE
        END IF
        RETURN
        END
C***********************************************************************
        SUBROUTINE STDUP(NP,NV,NSTD,STSIZE,GSIZE,FREQ,COEFS,CLASS,PHI,X,V)
        INTEGER NP,NV,NSTD,STSIZE(NP),GSIZE(NP),CLASS(1400),FREQ(1400),G
        REAL V(1400,16),COEFS(8,17),PHI(8,8),X(1400,8)
```

```
      CALL LDF(NP,NV,NSTD,STSIZE,GSIZE,FREQ,COEFS,V)
C     WRITE(6,701)
C     WRITE(6,702) ((COEFS(I,J),J=1,NV+1),I=1,NP)
      CALL CLASFY(NP,NV,NSTD,FREQ,CLASS,COEFS,V,X)
C     WRITE(6,705)
C     WRITE(6,706) (CLASS(G),G=1,NSTD)
      CALL GETPHI(NP,NSTD,FREQ,CLASS,STSIZE,GSIZE,PHI)
C     WRITE(6,707)
C     WRITE(6,708) ((PHI(I,J),J=1,NP),I=1,NP)
      RETURN
  701 FORMAT(//1X,'THE LDF COEFS ARE:'/)
  702 FORMAT(1X,10F12.3)
  705 FORMAT(//1X,'THE CLASSIFICATION VECTOR IS:'/)
  706 FORMAT(1X,20I4)
  707 FORMAT(//1X,'THE CLASSIFICATION MATRIX IS:'/)
  708 FORMAT(1X,8F12.4)
      END
C***********************************************************************
      SUBROUTINE READIN(NP,NV,STSIZE,NSTD,VARBLE,UNIT,POPNAM)
C NP  =NO. OF STOCK GROUPS SPECIFIED IN CTRL FILE (TAPE8)
C NPIN=NO. OF STOCK GROUPS FOUND ON TAPE7
C THIS SUBROUTINE CHECKS THAT NP AND NPIN ARE EQUAL.
C*****NB: THIS PROGRAM DOES NOT DO VARIABLE SELECTION.  THE
C*****VARIABLES BELOW WERE PREDETERMINED.
      INTEGER NPIN,NP,NV,STSIZE(NP),NSTD,COUNT,G,UNIT,SPLIT(0:8)
      REAL SUM,SIGMA,VARBLE(1400,16),MEANS(8),SDEVS(8)
      CHARACTER POPNAM(NP)*6
      NPIN=0
      G=1
  100 DO 1 I=1,1401
        COUNT=I
        READ(UNIT,*,END=2) (VARBLE(G,L),L=1,NV)
        G=G+1
        IF(G.GT.1401) THEN
          PRINT *,'MIXTURE SAMPLE IS TOO LARGE  > 1400 FOR STORAGE'
          STOP
        END IF
    1 CONTINUE
    2 IF(EOF(UNIT).NE.0) NOUGHT=0
      IF(COUNT-1.GT.0) THEN
        NPIN=NPIN+1
        IF(NPIN.GT.NP) THEN
          WRITE(6,701) NP,NPIN
          STOP
        END IF
        IF(COUNT-1.LT.2) THEN
          PRINT *,'ONLY ONE FISH IN ',POPNAM(NPIN),' STANDARD???'
          STOP
        END IF
        STSIZE(NPIN)=COUNT-1
        GO TO 100
      ELSE
        NSTD=G-1
        IF(NPIN.NE.NP) THEN
          WRITE(6,701) NP,NPIN
          STOP
        END IF
      END IF
C OUTPUT SUMMARY STATISTICS ON THE VARIABLES IN THE STANDARDS
      IF(UNIT.EQ.7) THEN
```

```
        WRITE(6,702)
        WRITE(6,703)  (POPNAM(J),J=1,NP)
        SPLIT(0)=0
        DO 800 J=1,NP
  800     SPLIT(J)=SPLIT(J-1)+STSIZE(J)
        DO 801 L=1,NV
          DO 802 J=1,NP
            SUM=0.0
            SIGMA=0.0
            DO 803 G=SPLIT(J-1)+1,SPLIT(J)
              SUM=SUM+VARBLE(G,L)
              SIGMA=SIGMA+VARBLE(G,L)*VARBLE(G,L)
  803       CONTINUE
            MEANS(J)=SUM/STSIZE(J)
            SDEVS(J)=SIGMA-STSIZE(J)*MEANS(J)*MEANS(J)
            SDEVS(J)=SQRT(SDEVS(J)/(STSIZE(J)-1))
  802     CONTINUE
        WRITE(6,704) L,(MEANS(J),J=1,NP)
        WRITE(6,705)  (SDEVS(J),J=1,NP)
  801 CONTINUE
      WRITE(6,702)
      END IF
      RETURN
  701 FORMAT(/1X,I2,' STOCK GROUPS SPECIFIED BUT ',I2,' FOUND???')
  702 FORMAT(/1X,79('=')//)
  703 FORMAT(25X,'MEAN AND STANDARD DEVIATION TABLE'//,
     .       1X,'VAR',8(9X,A6)/)
  704 FORMAT(/1X,I2,9X,8(G10.5,5X))
  705 FORMAT(12X,8(G9.3,TL9,'(',4X,')',9X))
      END
C**********************************************************************
      SUBROUTINE RESAMP(NP,NSTD,GSIZE,STSIZE,FREQ)
      INTEGER G,NP,NSTD,GSIZE(NP),STSIZE(NP),FREQ(NSTD),SPLIT(0:8),
     .        STARTJ
      REAL ORDERS(2001)
      COMMON /UNIFORM/ ORDERS
      SPLIT(0)=0
      DO 800 J=1,NP
  800   SPLIT(J)=SPLIT(J-1)+STSIZE(J)
      DO 801 G=1,NSTD
  801   FREQ(G)=0
      DO 1 J=1,NP
        CALL ORDVEC(GSIZE(J),FLOAT(STSIZE(J)),ORDERS)
        STARTJ=SPLIT(J-1)+1
        DO 2 G=1,GSIZE(J)
    2     FREQ(STARTJ+INT(ORDERS(G)))=FREQ(STARTJ+INT(ORDERS(G)))+1
    1 CONTINUE
      RETURN
      END
C**********************************************************************
      SUBROUTINE LDF(NP,NV,NSTD,STSIZE,GSIZE,FREQ,COEFS,VARBLE)
      INTEGER NP,NV,N,GSIZE(NP),FREQ(NSTD),G,SPLIT(0:8),STSIZE(NP),
     .        SIXTEN,GCOUNT
      REAL COEFS(8,17),MEANS(8,16),VARBLE(1400,16),WORK(8,16)
      DOUBLE PRECISION DET,SIGMA1,SIGMA2,SIGMA(16,16)
      SIXTEN=16
      N=0
      SPLIT(0)=0
      DO 800 J=1,NP
        SPLIT(J)=SPLIT(J-1)+STSIZE(J)
```

```
        GCOUNT=0
        DO 801 G=SPLIT(J-1)+1,SPLIT(J)
  801     GCOUNT=GCOUNT+FREQ(G)
C AS A CHECK ON THE PREVIOUS ROUTINES, TEST THAT THESE ARE THE SAME
        IF(GSIZE(J).NE.GCOUNT) THEN
          PRINT *,'CONFLICT OF STANDARD SIZE IN SUBROUTINE  LDF'
          STOP
        END IF
        N=N+GSIZE(J)
  800 CONTINUE
      DO 1 J=1,NP
        DO 1 L=1,NV
          SUM=0.0
          DO 2 G=SPLIT(J-1)+1,SPLIT(J)
    2       SUM=SUM+VARBLE(G,L)*FREQ(G)
          MEANS(J,L)=SUM/GSIZE(J)
    1 CONTINUE
C     WRITE(6,702)
C     WRITE(6,703) ((MEANS(J,L),L=1,NV),J=1,NP)
      DO 3 L=1,NV
        DO 3 LL=1,L
          SIGMA1=0.0
          SIGMA2=0.0
          DO 4 J=1,NP
    4       SIGMA2=SIGMA2+GSIZE(J)*MEANS(J,L)*MEANS(J,LL)
          DO 5 G=1,NSTD
    5       SIGMA1=SIGMA1+VARBLE(G,L)*VARBLE(G,LL)*FREQ(G)
          SIGMA(L,LL)=(SIGMA1-SIGMA2)/(N-NP)
          SIGMA(LL,L)=SIGMA(L,LL)
    3 CONTINUE
C     WRITE(6,700)
      CALL MXINT(SIGMA,NV,SIXTEN,DET)
      IF(DET.EQ.0.0) THEN
        PRINT *,'THE POOLED COVARIANCE MATRIX IS SINGULAR!'
        PRINT *,'POSSIBLY VARIABLES TOO HIGHLY CORRELATED - ABORT'
        STOP
      END IF
C     WRITE(6,701)
      CALL MXMULT(NP,NV,MEANS,SIGMA,WORK)
      DO 6 J=1,NP
        SUM=0.0
        DO 7 L=1,NV
          SUM=SUM+WORK(J,L)*MEANS(J,L)
          COEFS(J,L)=WORK(J,L)
    7   CONTINUE
        COEFS(J,NV+1)=-SUM/2.0
    6 CONTINUE
      RETURN
C 700 FORMAT(///1X,'POOLED COVARIANCE MATRIX'///)
C 701 FORMAT(///1X,'ITS INVERSE'///)
C 702 FORMAT(///1X,'THE RESAMPLED STANDARD MEANS ARE:'///)
C 703 FORMAT(1X,9F12.4)
      END
C***********************************************************************
      SUBROUTINE SIMMIX(NP,N,NSTD,STSIZE,FREQ,ACTUAL)
C THIS SUBROUTINE SIMULATES A MIXED SAMPLE OF SIZE N FROM A MIXED
C POPULATION HAVING PROPORTIONS  ACTUAL  .  SINCE THE MIXED SAMPLE
C FISH ARE ALL FROM THE STANDARD IT IS NECESSARY ONLY TO RETURN
C AN INTEGER VECTOR (FREQ) OF FREQUENCIES.
      INTEGER NP,N,NSTD,STSIZE(NP),FREQ(NSTD),SPLIT(0:8),INDEX,G
```

```
      REAL ACTUAL(NP),ASPLIT(0:8),ORDERS(2001),WORK,UPPER,AJ
      COMMON /UNIFORM/ ORDERS
      UPPER=1.0
      ASPLIT(0)=0.0
      SPLIT(0)=0
      DO 800 J=1,NP
        SPLIT(J)=SPLIT(J-1)+STSIZE(J)
        ASPLIT(J)=ASPLIT(J-1)+ACTUAL(J)
  800 CONTINUE
      ASPLIT(NP)=1+1.0E-12
      CALL ORDVEC(N,UPPER,ORDERS)
      DO 801 G=1,NSTD
  801   FREQ(G)=0
      J=1
      AJ=ACTUAL(J)
      DO 1 G=1,N
    2   IF(ORDERS(G).LT.ASPLIT(J)) THEN
C WORK IS ALWAYS BETWEEN ZERO AND ONE
C INDEX RANGES FROM SPLIT(J-1)+1 TO SPLIT(J)
          WORK=((ORDERS(G)-ASPLIT(J-1))/AJ)
          INDEX=INT(WORK*STSIZE(J))+1+SPLIT(J-1)
          FREQ(INDEX)=FREQ(INDEX)+1
          GOTO 1
        END IF
        J=J+1
        IF(J.GT.NP) THEN
          PRINT *,'ERROR IN SUBROUTINE SIMMIX.  PROGRAM STOPPED.'
          STOP
        END IF
        AJ=ACTUAL(J)
        GOTO 2
    1 CONTINUE
      RETURN
      END
C***********************************************************************
      SUBROUTINE CLASFY(NP,NV,NSTD,FREQ,CLASS,COEFS,VARBLE,X)
      INTEGER G,NP,NV,NSTD,FREQ(NSTD),CLASS(NSTD),MAX
      REAL COEFS(8,17),VARBLE(1400,16),LDFVAL(8),BIGEST,X(1400,8)
      CHARACTER TYPE*10
      COMMON /OPTIONS/ TYPE
      DO 1 G=1,NSTD
        IF(FREQ(G).GT.0.OR.TYPE(1:1).EQ.'S') THEN
          DO 2 J=1,NP
            LDFVAL(J)=0.0
            DO 3 L=1,NV
    3         LDFVAL(J)=LDFVAL(J)+COEFS(J,L)*VARBLE(G,L)
            LDFVAL(J)=LDFVAL(J)+COEFS(J,NV+1)
C IT IS ONLY THE RELATIVE SIZE OF THE LINEAR DISCRIMINANT FUNCTIONS
C THAT IS IMPORTANT SO WE CAN SET  LDFVAL(1) TO ZERO AND TRANSLATE
C THE OTHERS BY THE SAME AMOUNT.
C THE  X  MATRIX STORES THE 'LIKELIHOOD' VALUES THAT ARE REQUIRED BY
C THE DIRECT MAXIMUM LIKELIHOOD METHOD.
    2     CONTINUE
          DO 5 J=2,NP
            LDFVAL(J)=LDFVAL(J)-LDFVAL(1)
            X(G,J)=EXP(LDFVAL(J))
    5     CONTINUE
          LDFVAL(1)=0
          X(G,1)=1.0
          BIGEST=LDFVAL(1)
```

```
          MAX=1
          DO 4 J=2,NP
            IF(LDFVAL(J).GT.BIGEST) THEN
              MAX=J
              BIGEST=LDFVAL(J)
            END IF
    4     CONTINUE
        ELSE
          MAX=0
          DO 6 J=1,NP
    6       X(G,J)=0.0
        END IF
        CLASS(G)=MAX
    1 CONTINUE
      RETURN
      END
C*********************************************************************
      SUBROUTINE GETPHI(NP,NSTD,FREQ,CLASS,STSIZE,GSIZE,PHI)
      INTEGER G,NP,NSTD,FREQ(NSTD),CLASS(NSTD),STSIZE(NP),
     .        GSIZE(NP),SPLIT(0:8),CCOUNT(8)
      REAL CHECK,PHI(8,8)
      SPLIT(0)=0
      DO 800 J=1,NP
  800   SPLIT(J)=SPLIT(J-1)+STSIZE(J)
      DO 1 J=1,NP
        DO 801 JJ=1,NP
  801     CCOUNT(JJ)=0
        DO 2 G=SPLIT(J-1)+1,SPLIT(J)
    2     IF(FREQ(G).GT.0) CCOUNT(CLASS(G))=CCOUNT(CLASS(G))+FREQ(G)
        CHECK=0.0
        DO 3 JJ=1,NP
          PHI(JJ,J)=FLOAT(CCOUNT(JJ))/FLOAT(GSIZE(J))
          CHECK=CHECK+PHI(JJ,J)
    3   CONTINUE
        IF(CHECK.LE.0.9999.OR.CHECK.GE.1.0001) THEN
          PRINT *,'ERROR IN SUBROUTINE GETPHI.  PROGRAM STOPPED.'
          WRITE(6,700) CHECK
          STOP
        END IF
    1 CONTINUE
      RETURN
  700 FORMAT(///'OH NO - VALUE CHECK IS ',F8.4,' <>1.0'///)
      END
C*********************************************************************
      SUBROUTINE COOKLRD(NP,N,NSTD,FREQ,CLASS,THETA,PHIINV)
      INTEGER NP,N,NSTD,FREQ(NSTD),CLASS(NSTD),Y(8),G
      REAL THETA(8,5),A(8),SUM,ATOT,ATHTOT
      DOUBLE PRECISION PHIINV(8,8)
      DO 800 J=1,NP
  800   Y(J)=0
      DO 801 G=1,NSTD
  801   Y(CLASS(G))=Y(CLASS(G))+FREQ(G)
      DO 802 J=1,NP
  802   THETA(J,1)=FLOAT(Y(J))/FLOAT(N)
C
C NOW CALCULATE CORRECTED ESTIMATES
C
      DO 1 J=1,NP
        SUM=0.0
        DO 2 JJ=1,NP
```

```
    2       SUM=SUM+PHIINV(J,JJ)*THETA(JJ,1)
          THETA(J,2)=SUM
    1   CONTINUE
C
C NOW CALCULATE CONSTRAINED COOK ESTIMATES.  THIS METHOD CAN
C REQUIRE MORE THAN ONE ITERATION, BUT NEVER MORE THAN NP-1
        DO 888 J=1,NP
  888     THETA(J,3)=THETA(J,2)
        DO 10 ITERS=1,NP
          DO 889 J=1,NP
  889       IF(THETA(J,3).LT.0.0) GO TO 9
          RETURN
    9     ATOT=0.0
          ATHTOT=0.0
          DO 11 J=1,NP
            IF(THETA(J,3).GT.0.0) THEN
              A(J)=1.0
            ELSE
              A(J)=0.0
            END IF
            ATOT=ATOT+A(J)
            THETA(J,3)=A(J)*THETA(J,3)
            ATHTOT=ATHTOT+THETA(J,3)
   11     CONTINUE
          DO 12 J=1,NP
   12       THETA(J,3)=THETA(J,3)-A(J)*(ATHTOT-1.0)/ATOT
   10   CONTINUE
        PRINT *,'FAULT WITH CODE FOR COOK''S CONSTRAINED ESTIMATE'
        WRITE(6,700) ITERS,(A(J),J=1,NP)
        WRITE(6,701) ((THETA(J,K),K=1,3),J=1,NP)
  700   FORMAT(1X,'ITERS=',I2,'  A: ',8F4.1)
  701   FORMAT(1X,'THETA:'/8(/1X,3F12.8))
        STOP
        END
C***********************************************************************
        SUBROUTINE MILLARC(NP,N,THETA,X,TYPE)
C THIS SUBROUTINE OBTAINS THE MAXIMUM LIKELIHOOD CONSTRAINED CORRECTED
C ESTIMATES AS ADVOCATED BY MILLAR.  THE EM (EXPECTATION MAXIMIZATION)
C ALGORITHM IS USED.  SUBROUTINE  EM  BELOW IS THE SAME VERSION USED
C BY WDF TO DO DIRECT MAXIMUM LIKELIHOOD.  THE INPUT FOR  EM  IS
C CREATED BELOW - MUCH OF IT IS SETTING ALGORITHM SPECIFICATIONS.
        INTEGER NP,N,Y(8),IMAX,MAXG,NROWSX
        REAL THETA(8,5),THETA4(8),LAMBDA(8),X(8,8),TOL
        CHARACTER OPT1*3,OPT5*3,TYPE*10
        IMAX=100
C LAMBDA IS NOT BY THIS ROUTINE
C GIVE 'EVEN' STARTING VALUES FOR THE ESTIMATES
        DO 4 J=1,NP
    4     THETA4(J)=1.0/FLOAT(NP)
        TOL=1.0E-06
C OPT1 IS A PRINT OPTION.  IF YES THEN INTERMEDIATE RESULTS ARE PRINTED
C EVERY ITERATION.  IT WILL BE SET TO 'NO' BUT COULD BE ALLOWED AS A
C CONTROL FILE OPTION IF DESIRED.
C OPT5 CONTROLS USE OF THE EM ACCELERATION SUBROUTINE  ACCEL
        OPT1='NO'
        OPT5='YES'
        MAXG=NP
C CALCULATE CLASSIFICATION FREQUENCIES
        DO 1 J=1,NP
    1     Y(J)=NINT(THETA(J,1)*N)
```

```
C NROWSX  IS THE NUMBER OF ROWS IN  X  (THE CLASSIFICATION MATRIX)
      NROWSX=8
C
C HERE GOES.....
C
      CALL EM(NP,IMAX,LAMBDA,X,THETA4,TOL,OPT1,OPT5,MAXG,Y,N,NROWSX)
      DO 2 J=1,NP
    2   THETA(J,4)=THETA4(J)
      RETURN
      END
C*********************************************************************
      SUBROUTINE ML(NP,N,MAXG,Y,THETA,X,TYPE)
C THIS SUBROUTINE OBTAINS THE DIRECT MAXIMUM LIKELIHOOD ESTIMATES.
C THE EM (EXPECTATION MAXIMIZATION) ALGORITHM IS USED.
C SUBROUTINE  EM  BELOW IS THE SAME VERSION USED BY WDF.
C THE INPUT FOR  EM  IS CREATED BELOW -
C MUCH OF IT IS SETTING ALGORITHM SPECIFICATIONS.
      INTEGER NP,N,Y(MAXG),IMAX,MAXG,NROWSX
      REAL THETA(8,5),THETA5(8),LAMBDA(1400),X(1400,8),TOL
      CHARACTER OPT1*3,OPT5*3,TYPE*10
      IMAX=100
C LAMBDA IS NOT USED HERE
C GIVE 'EVEN' STARTING VALUES FOR THE ESTIMATES
      DO 4 J=1,NP
    4   THETA5(J)=1.0/FLOAT(NP)
      TOL=1.0E-06
C OPT1 IS A PRINT OPTION.  IF YES THEN INTERMEDIATE RESULTS ARE PRINTED
C EVERY ITERATION.
      OPT1='NO'
      OPT5='YES'
C
C HERE GOES.....
C
      NROWSX=1400
      CALL EM(NP,IMAX,LAMBDA,X,THETA5,TOL,OPT1,OPT5,MAXG,Y,N,NROWSX)
      DO 2 J=1,NP
    2   THETA(J,5)=THETA5(J)
      RETURN
      END
C*********************************************************************
      SUBROUTINE ORDVEC(NUM,UPPER,ORDERS)
      INTEGER NUM
      REAL UPPER,ORDERS(NUM+1)
      ORDERS(1)=-ALOG(RANF())
      DO 1 I=2,NUM+1
    1   ORDERS(I)=ORDERS(I-1)-ALOG(RANF())
      C=UPPER/ORDERS(NUM+1)
      DO 2 I=1,NUM
    2   ORDERS(I)=ORDERS(I)*C
      RETURN
      END
C*********************************************************************
      SUBROUTINE RESULTS(NP,NS,ACTUAL,TYPE,POPNAM)
      INTEGER NP,NS
      REAL THSUM(8,5),THSSQ(8,5),THMSE(8,5),ACTUAL(NP),COVAR(8,8)
      CHARACTER TYPE*10,POPNAM(NP)*6
      COMMON THSUM,THSSQ,COVAR
      IF(TYPE(1:1).EQ.'A') THEN
        WRITE(6,704)
        NS=0
```

```
      ELSE
        WRITE(6,700) NS
      END IF
      IF(NS.GT.1) THEN
        DO 1 K=1,5
          DO 1 J=1,NP
            THSUM(J,K)=THSUM(J,K)/NS
            THSSQ(J,K)=(THSSQ(J,K)-NS*THSUM(J,K)*THSUM(J,K))/(NS-1)
            THSSQ(J,K)=SQRT(THSSQ(J,K))
    1   CONTINUE
        DO 11 K=1,NP
          DO 11 J=1,NP
            COVAR(J,K)=COVAR(J,K)-NS*THSUM(J,5)*THSUM(K,5)
            COVAR(J,K)=COVAR(J,K)/(NS-1)
   11   CONTINUE
      END IF
      WRITE(6,701)
      DO 888 J=1,NP
  888 WRITE(6,702) POPNAM(J),(THSUM(J,K),K=1,5)
      IF(NS.GT.1) THEN
        WRITE(6,703) NS
        WRITE(6,701)
        DO 889 J=1,NP
  889   WRITE(6,702) POPNAM(J),(THSSQ(J,K),K=1,5)
        IF(TYPE(1:1).EQ.'S') THEN
          DO 2 K=1,5
            DO 2 J=1,NP
              WORK=ACTUAL(J)-THSUM(J,K)
              THMSE(J,K)=SQRT(WORK*WORK+THSSQ(J,K)*THSSQ(J,K))
    2     CONTINUE
          WRITE(6,705)
          WRITE(6,701)
          DO 890 J=1,NP
  890     WRITE(6,702) POPNAM(J),(THMSE(J,K),K=1,5)
        END IF
        WRITE(6,706) NS
        CALL MXOUT(NP,COVAR)
        WRITE(6,707)
        CALL RHO(NP,COVAR)
        CALL MXOUT(NP,COVAR)
      END IF
      RETURN
  700 FORMAT(//'1'/1X,'TABLE OF COMPOSITION ESTIMATE MEANS.   ',
     .'NUMBER OF RUNS =',I5//)
  701 FORMAT(11X,'RAW',6X,'COOK & LORD',
     .6X,'COOK',8X,'MILLAR',6X,'MAXIMUM'/,
     .34X,'CONSTRAINED',2X,'CONSTRAINED',2X,'LIKELIHOOD'//)
  702 FORMAT(1X,A6,F8.4,4(5X,F8.4))
  703 FORMAT(////,1X,'TABLE OF COMPOSITION ESTIMATE STANDARD ',
     .'DEVIATIONS OVER THE',I5,' RUNS'//)
  704 FORMAT(//'1',1X,'TABLE OF COMPOSITION ESTIMATES'//)
  705 FORMAT(////,1X,'TABLE OF MEAN SQUARED ERRORS'//)
  706 FORMAT('1THE COVARIANCE MATRIX OF THE',I5,' MAXIMUM LIKELIHOOD ',
     . 'COMPOSITION ESTIMATES IS:'///)
  707 FORMAT(////' THE CORRESPONDING CORRELATION MATRIX IS:')
      END
C*********************************************************************
      SUBROUTINE MXINT(A,NRR,NRA,DET)
      DIMENSION ISW(96), A(*)
      DOUBLE PRECISION A,DET
```

```
C THIS MATRIX INVERSION SUBROUTINE TAKES A  NRA*NRA  SQUARE MATRIX AND
C INVERTS THE UPPER  NRR*NRR  SUBMATRIX

      NA = NRA+1
      ND = NRR*NRA
      DO 1 I=1,NRR
    1 ISW(I) = I
      NDIAG = 1
      NUP = 1
      DET = 1.
      DO 9 I=1,NRR
         KA = I
         AM = 0.
         LA = NDIAG
         DO 2 J=I,NRR
            IF (DABS(A(LA)).LE.AM) GO TO 2
            KA = J
            AM = DABS(A(LA))
    2    LA = LA+1
         IF (AM.GT.1.0E-20) GO TO 3
         WRITE (6,14)
         DET = 0.
         RETURN
    3    IF (KA.EQ.I) GO TO 5
         MUP = I
         DO 4 MLOW=KA,ND,NRA
            S = A(MUP)
            A(MUP) = A(MLOW)
            A(MLOW) = S
    4    MUP = MUP+NRA
         IS = ISW(KA)
         ISW(KA) = ISW(I)
         ISW(I) = IS
    5    AA = A(NDIAG)
         DET = DET*AA
         DO 6 J=I,ND,NRA
    6    A(J) = A(J)/AA
         A(NDIAG) = 1./AA
         NEL = NUP
         DO 8 J=1,NRR
            IF (J.EQ.I) GO TO 8
            AA = A(NEL)
            A(NEL) = 0.
            KA = I
            DO 7 K=J,ND,NRA
               A(K) = A(K)-AA*A(KA)
    7       KA = KA+NRA
    8    NEL = NEL+1
         NUP = NUP+NRA
    9 NDIAG = NDIAG+NA
      I = 1
   10 IF (ISW(I).NE.I) GO TO 11
      I = I+1
      IF (I.GT.NRR) GO TO 13
   11 J = ISW(I)
      ISW(I) = ISW(J)
      ISW(J) = J
      MLOW = (I-1)*NRA+1
      MUP = (J-1)*NRA+1
      DO 12 K=1,NRR
```

```
         S = A(MLOW)
         A(MLOW) = A(MUP)
         A(MUP) = S
         MLOW = MLOW+1
   12 MUP = MUP+1
      GO TO 10

   13 RETURN
   14 FORMAT (1X,19HPIVOT ZERO IN MXINT)
      END
C**********************************************************************
      SUBROUTINE MXOUT(NP,V)
      DIMENSION V(8,8)
      REAL V
C THIS MATRIX PRINTING ROUTINE IS GENERAL ENOUGH TO PRINT ANY SIZE SQUARE
C MATRIX BY SPLITTING IT UP INTO PAGE SIZE PORTIONS.  THE MATRIX IS
C ASSUMED SYMMETRIC AND ONLY THE LOWER TRIANGULAR PART IS PRINTED.

      DO 39 I=1,NP,7
         K=I+6
         IF(K.GT.NP)K=NP
            WRITE(6,33)I,NP,I,K
            WRITE(6,35) (L,L=I,K)
         END IF
         DO 37 J=I,NP
            L=J
            IF(J.LE.K)GO TO 1
            L=K
    1       WRITE(6,36)J,(V(J,M),M=I,L)
   37    CONTINUE
   39    CONTINUE
   33 FORMAT(1H-,13H(POPULATIONS ,I2,1H-,I2,3H * ,I2,1H-,I2,1H))
   35 FORMAT(2H0 ,7I12)
   36 FORMAT(1H ,I3,7E12.6)
      RETURN
      END
C*******************************************************************
      SUBROUTINE RHO(NP,COVAR)
      REAL COVAR(8,8)

      DO 3 I=1,NP
      DO 3 J=NP,I,-1
        IF(COVAR(I,I)*COVAR(J,J).EQ.0.0) THEN
          COVAR(J,I)=0.
          IF(I.EQ.J) COVAR(I,I)=1.0
        ELSE
          COVAR(J,I)=COVAR(J,I)/SQRT(COVAR(I,I)*COVAR(J,J))
        END IF
    3 CONTINUE

      RETURN
      END
C*******************************************************************
      SUBROUTINE ESTOUT(NP,THETA)
      INTEGER NP,EST
      REAL THETA(8,5),THSUM(8,5),THSSQ(8,5),COVAR(8,8)
      COMMON THSUM,THSSQ,COVAR
      DO 1 EST=1,5
        DO 1 J=1,NP
          THSUM(J,EST)=THSUM(J,EST)+THETA(J,EST)
```

```
           THSSQ(J,EST)=THSSQ(J,EST)+THETA(J,EST)*THETA(J,EST)
    1 CONTINUE
      DO 2 J=1,NP
        DO 2 K=1,NP
    2      COVAR(J,K)=COVAR(J,K)+THETA(J,5)*THETA(K,5)
C     WRITE(2,700)  ((THETA(J,EST),J=1,NP),EST=2,5)
C 700 FORMAT(16(1X,F6.4))
      RETURN
      END
C**********************************************************************
      SUBROUTINE MXMULT(N1,N2,A,B,C)
C CALCULATES  C = AB.  A  IS (N1*N2) AND  B IS (N2*N2)
      DIMENSION A(8,16),B(16,16),C(8,16)
      DOUBLE PRECISION B
      DO 1 I=1,N1
         DO 1 J=1,N2
            S=0.0
            DO 2 K=1,N2
    2          S=S+A(I,K)*B(K,J)
            C(I,J)=S
    1    CONTINUE
      RETURN
      END
C**********************************************************************
      SUBROUTINE EM(NP,IMAX,LAMBDA,X,THETA,TOL,OPT1,OPT5,MAXG,Y,N,NROWS)
      REAL LAMBDA(MAXG),X(NROWS,8),THETA(NP),WORK(8),THETA1(8)
      REAL DIFF1(8),DIFF2(8)
      INTEGER G,Y(MAXG),POINTR(1400),NROWS,GCOUNT,MAXG,TOTG
      CHARACTER OPT1*3,OPT5*3,POPNAM(8)*6,FLAG
      DATA POPNAM/'POPN 1','POPN 2','POPN 3','POPN 4','POPN 5','POPN 6',
     .            'POPN 7','POPN 8'/
      LAG=1
      FLAG='N'
      DO 72 K=1,NP
   72    DIFF1(K)=0.0
      HOOD=0.0
      AA=1.0
C CONDENSE  X & Y
      GCOUNT=0
      DO 8 G=1,MAXG
        IF(Y(G).GT.0) THEN
          GCOUNT=GCOUNT+1
          POINTR(GCOUNT)=G
        END IF
    8 CONTINUE
      TOTG=GCOUNT

      DO 1 ITER=0,IMAX
C THE VARIABLES  FLAG AND LAG  CONTROL THE ACCELERATION PROCESS.
C IF FLAG.EQ.'Y' IS TRUE THEN IT INDICATES A SUCCESSFUL ACCELERATION
C IN WHICH CASE  LAMBDA & HOOD  HAVE ALREADY BEEN CALCULATED IN
C SUBROUTINE ACCEL.
         IF(FLAG.EQ.'Y')GO TO 333
         OHOOD=HOOD
         HOOD=0.0
         DO 2 GCOUNT=1,TOTG
            G=POINTR(GCOUNT)
            LAMBDA(G)=0.0
            DO 10 K=1,NP
               LAMBDA(G)=LAMBDA(G)+X(G,K)*THETA(K)
```

```
   10        CONTINUE
             IF(LAMBDA(G).LT.1.0E-12.OR.LAMBDA(G).GT.1.0E+12) THEN
               WRITE(6,98)ITER,G,LAMBDA(G)
             END IF
    2     CONTINUE
C TEST TO SEE WHETHER IT IS NECESSARY TO CALCULATE THE LOG LIKELIHOOD
C ON THIS ITERATION
          IF(AA.LT.TOL.OR.ITER.EQ.IMAX)GO TO 21
          IF(OPT5.EQ.'NO'.OR.LAG.GE.3.OR.ITER.LE.8)GO TO 334
   21     DO 22 GCOUNT =1,TOTG
             G=POINTR(GCOUNT)
             HOOD=HOOD+Y(G)*ALOG(LAMBDA(G))
   22     CONTINUE
  333     IF(AA.LT.TOL.OR.ITER.EQ.IMAX)GO TO 900
C TEST TO SEE IF PRINTOUT REQUIRED EACH ITERATION
  334     IF(OPT1.EQ.'NO') GO TO 14
          WRITE(6,37)ITER
          IF(HOOD.NE.0.0) WRITE(6,38)HOOD
          WRITE(6,48)
          DO 29 K=1,NP
             WRITE(6,39)K,POPNAM(K),THETA(K)
   29     CONTINUE
          WRITE(6,40)AA

   14     DO 11 K=1,NP
   11        WORK(K)=0.0
          DO 3 GCOUNT=1,TOTG
             G=POINTR(GCOUNT)
             QUOTI=Y(G)/LAMBDA(G)
             DO 3 K=1,NP
                WORK(K)=WORK(K)+QUOTI*X(G,K)
    3     CONTINUE
          DO 4 K=1,NP
             THETA1(K)=WORK(K)*THETA(K)/N
             DIFF2(K)=DIFF1(K)
             DIFF1(K)=THETA1(K)-THETA(K)
    4     CONTINUE
          AA=0.0
          DO 20 K=1,NP
             AA=AMAX1(AA,ABS(DIFF1(K)))
             THETA(K)=THETA1(K)
   20     CONTINUE
          IF(OPT5.EQ.'NO'.OR.ITER.LT.10)GO TO 1
          LAG=LAG-1
          FLAG='N'
          IF(LAG.NE.0)GO TO 1
          HOOD=HOOD+2*(HOOD-OHOOD)
          CALL ACCEL(NP,MAXG,DIFF2,DIFF1,THETA,HOOD,LAG,AA,X,LAMBDA,Y,
                    NROWS,POINTR,TOTG)
          IF(LAG.EQ.3) THEN
             IF(OPT1.EQ.'YES') WRITE(6,50) ITER
             FLAG='Y'
          END IF
          IF(LAG.EQ.10.AND.OPT1.EQ.'YES')WRITE(6,51)ITER
    1 CONTINUE
C......................END OF EM ALGORITHM............................
  900 IF(OPT1.NE.'YES') RETURN
      WRITE(6,37)ITER
      WRITE(6,38)HOOD
      WRITE(6,48)
```

```
      DO 30 K=1,NP
         WRITE(6,39) K,POPNAM(K),THETA(K)
   30 CONTINUE
      WRITE(6,40)AA
      IF(AA.LT.TOL) THEN
         WRITE(6,42)
      ELSE
         WRITE(6,41)
      END IF
      RETURN

   37 FORMAT(///1H ,8('==========='),//,1H ,'ESTIMATED COMPOSITION OF ',
     .'MIXED FISHERY AT ITERATION ',I4)
   38 FORMAT(//1H ,'THE LOGLIKELIHOOD IS ',F12.4)
   48 FORMAT(1H //,10X,10HPOPULATION,10X,12HCONTRIBUTION/)
   39 FORMAT(1H ,I6,5X,A6,13X,F9.6)
   40 FORMAT(//1H ,'CONVERGENCE ',F10.6)
   41 FORMAT(3(/1H ,'*****MAXIMUM NUMBER OF ITERATIONS REACHED*****'))
   42 FORMAT(3(/1H ,'*****CONVERGENCE TO SPECIFIED TOLERANCE******'))
   50 FORMAT(//1H ,'***********************************',
     .         /1H ,'*SUCCESSFUL ACCELERATION COMPLETED*  ITER=',I4,
     .         /1H ,'***********************************')
   51 FORMAT(//1H ,'*************************************',
     .         /1H ,'*ACCELERATION ATTEMPTED BUT DECLINED*  ITER=',I4,
     .         /1H ,'*************************************')
   98 FORMAT(1H ,'CAUTION: AT ITER ',I4,' LAMBDA(',I4,')=',E9.2)
      END
C***********************************************************************
      SUBROUTINE ACCEL(NP,MAXG,DIFF2,DIFF1,THETA,HOOD,LAG,AA,X,LAMBDA,Y,
     .                 NROWSX,POINTR,TOTG)
      REAL LAMBDA(MAXG)
      INTEGER G,GCOUNT,NROWSX,Y(MAXG),POINTR(TOTG),TOTG,MAXG
      DIMENSION DIFF2(8),DIFF1(8),THETA(8),X(NROWSX,8),RATIO(8),
     .HALT(8),THETA1(8)
      ALPHA=100.0
      SUM=0.0
      DO 1 K=1,NP
         IF(ABS(DIFF1(K)).GE.AA/20) GO TO 2
         HALT(K)=0.0
         GO TO 1
    2    IF(DIFF1(K).LT.0.0)HALT(K)=-THETA(K)/DIFF1(K)
         IF(DIFF1(K).GT.0.0)HALT(K)=(1-THETA(K))/DIFF1(K)
         HALT(K)=INT(HALT(K))
    1 CONTINUE
      DO 3 K=1,NP
         IF(ABS(DIFF1(K)).LT.AA/20)GO TO 3
         RATIO(K)=DIFF1(K)/DIFF2(K)
         ALPHA=AMIN1(ALPHA,RATIO(K))
    3 CONTINUE
      IF(ALPHA.GT.0.75)GO TO 8
      LAG=10
      RETURN
    8 IF(ALPHA.EQ.1.0)ALPHA=0.99
      ALPHA=AMIN1(10.0,(1/(1-ALPHA))-1)
      DO 5 K=1,NP
         THETA1(K)=THETA(K)+AMIN1(HALT(K),ALPHA)*DIFF1(K)
         SUM=SUM+THETA1(K)
    5 CONTINUE
      DO 7 K=1,NP
    7    THETA1(K)=THETA1(K)/SUM
```

```
C CHECK TO SEE WHETHER ACCELERATION GIVES IMPROVEMENT IN LIKELIHOOD
      HOOD1=0.0
      DO 30 GCOUNT=1,TOTG
          G=POINTR(GCOUNT)
          LAMBDA(G)=0.0
          DO 31 K=1,NP
   31         LAMBDA(G)=LAMBDA(G)+X(G,K)*THETA1(K)
          HOOD1=HOOD1+Y(G)*ALOG(LAMBDA(G))
   30 CONTINUE
      IF(HOOD1.GT.HOOD)GO TO 32
      LAG=10
      RETURN
   32 AA=0.0
      HOOD=HOOD1
      DO 33 K=1,NP
          AA=AMAX1(AA,ABS(THETA1(K)-THETA(K)))
          THETA(K)=THETA1(K)
   33 CONTINUE
      LAG=3
      RETURN
      END
```

APPENDIX 2.

LISTING OF ANALYSIS PROGRAM

```
      PROGRAM ANNA(TAPE7,TAPE8,TAPE9,OUTPUT,TAPE6,TAPE2)
C
C COPYRIGHT:  THIS PROGRAM WAS WRITTEN BY  RUSSELL MILLAR (DEPT OF STAT,
C 16 OCT 87   GN-22) FOR F.R.I. USE.    F.R.I. RESERVES THE RIGHT TO
C             DISTRIBUTE THIS PROGRAM AT THEIR DISCRETION.
C
C.......................................................................
C TAPE7 CONTAINS ALL OF THE STANDARDS APPENDED TOGETHER WITH END OF
C   FILE MARKERS SEPARATING THEM.
C TAPE8 IS THE CONTROL FILE - SPECIFIES THE TITLE OF THE RUN
C   AND #POPNS, POPN NAMES, #VARIABLES   ETC ETC.
C TAPE9 CONTAINS THE MIXED SAMPLE FILE.
C
C>>>>>>>>>TAPE7 AND TAPE9 ARE ASSUMED TO BE IN FREE FORMAT<<<<<<<<<<<<<<
C.......................................................................
C SOME VARIABLES USED INCLUDE:
C   NP=NUMBER OF STOCK GROUPS (POPULATIONS) ON TAPE7
C   NV=NUMBER OF VARIABLES
C   NSTD=TOTAL NUMBER OF FISH OVER ALL THE STANDARDS
C   NMIX(OR N)=SIZE OF MIXED SAMPLE ON TAPE9
C**********************************************************************
C************PRESET LIMITS FOR THIS VERSION OF SOURCE CODE ARE**********
C************ NP<=8    NV<=20    NSTD=1000       NMIX<=3000    ************
C**********************************************************************
C THE INDEX VARIABLES IN  DO  LOOPS ARE PREDOMINANTLY USED IN THE
C   FOLLOWING WAY:
C   J  INDEXS THE STOCK GROUP    J=1,...,NP
C   G  INDEXS FISH IN THE STDS OR MIXED SAMPLE    G=1,...,NSTD OR NMIX
C   L  INDEXS THE VARIABLES    L=1,...,NV
C SOME ARRAYS USED INCLUDE:
C   STSIZE(J)=SIZE OF STANDARD FOR STOCK GROUP J
C   GSIZE(J)=SIZE OF RESAMPLED STANDARD.  OFTEN THE SAME AS STSIZE(J).
C   FREQ(G)=FREQUENCY OF FISH  G  IN THE RESAMPLED DATA.
C   CLASS(G)=THE CLASS FISH  G  IS CLASSIFIED TO.
C   VARBLE(G,L)=THE L'TH MEASUREMENT ON THE G'TH FISH.
C   THETA(J,K)=THE CONTRIBUTION ESTIMATES
C             THETA(*,1)=RAW CLASSIFICATION ESTIMATES
C             THETA(*,2)=COOK AND LORD CORRECTED ESTIMATES
C             THETA(*,3)=COOK CONSTRAINED CORRECTED ESTIMATES
C             THETA(*,4)=MILLAR CONSTRAINED CORRECTED ESTIMATES
C              NB: THE LAST THREE TYPES OF ESTIMATES WILL BE IDENTICAL
C                  WHEN THE COOK AND LORD CORRECTED ESTIMATES ARE ALL
C                  NON NEGATIVE.
C             THETA(*,5)=DIRECT MAXIMUM LIKELIHOOD ESTIMATES
C**********************************************************************
C.......................................................................
      INTEGER NP,NV,NMIX,NSTD,STSIZE(8)
      REAL X(3000,8),THETA(8,5),PHI(8,8)
      CHARACTER POPNAM(8)*6
      CHARACTER *10 YMD,HOUR

      CALL DATE(YMD)
      CALL TIME(HOUR)
      WRITE(6,700) YMD,HOUR

C     CALL CONNEC(5)
      CALL READ8(NP,POPNAM,NV)
      CALL SETUP(NP,NV,NMIX,NSTD,STSIZE,PHI,X,POPNAM)
      CALL ANALYSE(NP,NMIX,PHI,X,THETA,POPNAM)
      CALL STDDEVS(NP,NMIX,PHI,X,THETA,POPNAM,STSIZE)
```

```
      STOP
  700 FORMAT(' OUTPUT FROM ANALYSIS PROGRAM:    EXECUTION DATE ',
     . A10,4X,A10//)
      END
C*******************************************************************
      SUBROUTINE READ8(NP,POPNAM,NV)
C THIS SUBROUTINE READS THE CONTROL FILE (TAPE8)
      INTEGER NP,NV
      CHARACTER POPNAM(8)*6,TITLE*80
      READ(8,*) TITLE
      WRITE(6,720) TITLE
      READ(8,*) NP,(POPNAM(J),J=1,NP)
      WRITE(6,722) NP
      WRITE(6,7225) (POPNAM(J),J=1,NP)
      IF(NP.LT.2.OR.NP.GT.8) THEN
        PRINT *,'NUMBER OF STOCKS IS OUT OF BOUNDS.  2 - 8'
        STOP
      END IF
      READ(8,*) NV
      WRITE(6,723) NV
      IF(NV.LT.1.OR.NV.GT.20) THEN
        PRINT *,'NUMBER OF VARIABLES IS OUT OF BOUNDS,  1 - 20'
        STOP
      END IF
      RETURN
  720 FORMAT(/1X,A80//)
  722 FORMAT(1X,'#STOCKS IN THE MODEL............',I2)
 7225 FORMAT(1X,'THE STOCKS ARE..................',8A6)
  723 FORMAT(1X,'#VARIABLES USED.................',I2/)
      END
C*******************************************************************
      SUBROUTINE SETUP(NP,NV,NMIX,NSTD,STSIZE,PHI,X,POPNAM)
      INTEGER G,NP,NV,NMIX,NSTD,STSIZE(NP)
      REAL X(3000,8),COEFS(8,21),STDVAR(1000,20),PHI(8,8)
      CHARACTER POPNAM(NP)*6
C...................................................................
C THE STANDARD VARIABLES (STDVAR) ARE STORED FOR TWO REASONS:
C -IT ENABLED CODE TO BE TAKEN FROM PROGRAM  HISEA
C -OTHERWISE THE STANDARDS HAVE TO BE READ TWICE, ONCE TO CONSTRUCT LDF
C  AND AGAIN TO DO THE CLASSIFICATION
C IF THE PROGRAM DIMENSIONS NEED TO BE INCREASED THEN CONSIDERABLE
C STORAGE COULD BE SAVED BY NOT STORING THE STANDARDS.
C...................................................................
      CALL READ7(NP,NV,NSTD,STSIZE,STDVAR)
      CALL LDF(NP,NV,NSTD,STSIZE,COEFS,STDVAR,POPNAM)
      CALL GETPHI(NP,NV,NSTD,STSIZE,COEFS,STDVAR,PHI)
      CALL READ9(NP,NV,NMIX,COEFS,X)
      RETURN
      END
C*******************************************************************
      SUBROUTINE READ7(NP,NV,NSTD,STSIZE,STDVAR)
C NP   =NO. OF STOCK GROUPS SPECIFIED IN CTRL FILE (TAPE8)
C NPIN=NO. OF STOCK GROUPS FOUND ON TAPE7
C THIS SUBROUTINE CHECKS THAT NP AND NPIN ARE EQUAL.
C*****NB: THIS PROGRAM DOES NOT DO VARIABLE SELECTION.  THE NV
C*****VARIABLES BELOW WERE PREDETERMINED.
      INTEGER NPIN,NP,NV,STSIZE(NP),NSTD,COUNT,G
      REAL STDVAR(1000,20)
      NPIN=0
      G=1
```

```
  100 DO 1 I=1,1001
        COUNT=I
        READ(7,*,END=2) (STDVAR(G,L),L=1,NV)
        IF(G.GT.1000) THEN
          PRINT *,'STORAGE FOR STANDARDS EXCEEDED IN SUBROUTINE  READ7'
          STOP
        END IF
        G=G+1
    1 CONTINUE
    2 IF(EOF(7).NE.0) NOUGHT=0
      IF(COUNT-1.GT.0) THEN
        NPIN=NPIN+1
        IF(NPIN.GT.NP) THEN
          WRITE(6,701) NP,NPIN
          STOP
        END IF
        STSIZE(NPIN)=COUNT-1
        GO TO 100
      ELSE
        WRITE(6,702) (STSIZE(J),J=1,NPIN)
        NSTD=G-1
        IF(NPIN.NE.NP) THEN
          WRITE(6,701) NP,NPIN
          STOP
        END IF
      END IF
      RETURN
  701 FORMAT(/1X,I2,' STOCK GROUPS SPECIFIED BUT ',I2,' FOUND???')
  702 FORMAT(/1X,'THE SIZES OF THE STANDARDS ARE ',8I6)
      END
C*********************************************************************
      SUBROUTINE LDF(NP,NV,NSTD,STSIZE,COEFS,STDVAR,POPNAM)
      INTEGER NP,NV,NSTD,G,SPLIT(0:8),STSIZE(NP),SIX,TWENTY
      REAL COEFS(8,21),STDVAR(1000,20),WORK(8,20)
      REAL SSQ,SUM,MEANS(8,20),SDEVS(8)
      DOUBLE PRECISION DET,SIGMA1,SIGMA2,SIGMA(20,20)
      CHARACTER POPNAM(NP)*6
      TWENTY=20
      SPLIT(0)=0
      DO 800 J=1,NP
  800   SPLIT(J)=SPLIT(J-1)+STSIZE(J)
C OUTPUT SUMMARY STATISTICS ON THE STANDARD VARIABLE
      WRITE(6,702)
      WRITE(6,703) (POPNAM(J),J=1,NP)
      DO 1 L=1,NV
        DO 2 J=1,NP
          SUM=0.0
          SSQ=0.0
          DO 801 G=SPLIT(J-1)+1,SPLIT(J)
            SUM=SUM+STDVAR(G,L)
            SSQ=SSQ+STDVAR(G,L)*STDVAR(G,L)
  801     CONTINUE
          MEANS(J,L)=SUM/STSIZE(J)
          SDEVS(J)=SSQ-STSIZE(J)*MEANS(J,L)*MEANS(J,L)
          IF(STSIZE(J).GT.1) THEN
            SDEVS(J)=SQRT(SDEVS(J)/(STSIZE(J)-1))
          ELSE
            SDEVS(J)=0.0
          END IF
    2   CONTINUE
```

```
            WRITE(6,704)  L,(MEANS(J,L),J=1,NP)
            WRITE(6,705)  (SDEVS(J),J=1,NP)
    1   CONTINUE
        WRITE(6,702)
        DO 3 L=1,NV
          DO 3 LL=1,L
            SIGMA1=0.0
            SIGMA2=0.0
            DO 4 J=1,NP
    4         SIGMA2=SIGMA2+STSIZE(J)*MEANS(J,L)*MEANS(J,LL)
            DO 5 G=1,NSTD
    5         SIGMA1=SIGMA1+STDVAR(G,L)*STDVAR(G,LL)
            SIGMA(L,LL)=(SIGMA1-SIGMA2)/(NSTD-NP)
            SIGMA(LL,L)=SIGMA(L,LL)
    3   CONTINUE
        SIX=6
        WRITE(6,700)
        CALL MXOUT(NV,SIGMA,SIX)
        WRITE(6,702)
        CALL MXINT(SIGMA,NV,TWENTY,DET)
        IF(DET.EQ.0.0) THEN
          PRINT *,'THE POOLED COVARIANCE MATRIX IS SINGULAR!'
          PRINT *,'POSSIBLY VARIABLES TOO HIGHLY CORRELATED - ABORT'
          STOP
        END IF
C       WRITE(6,701)
C       CALL MXOUT(NV,SIGMA,SIX)
        CALL MXMULT(NP,NV,MEANS,SIGMA,WORK)
        DO 6 J=1,NP
          SUM=0.0
          DO 7 L=1,NV
            SUM=SUM+WORK(J,L)*MEANS(J,L)
            COEFS(J,L)=WORK(J,L)
    7     CONTINUE
          COEFS(J,NV+1)=-SUM/2.0
    6   CONTINUE
        RETURN
  700 FORMAT('1'/1X,'POOLED COVARIANCE MATRIX'/)
C 701 FORMAT(///1X,'ITS INVERSE'///)
  702 FORMAT(/1X,79('=')//)
  703 FORMAT(25X,'MEAN AND STANDARD DEVIATION TABLE'//,
     .        1X,'VAR',8(9X,A6)/)
  704 FORMAT(/1X,I2,9X,8(G10.5,5X))
  705 FORMAT(12X,8(G9.3,TL9,'(',4X,')',9X))
        END
C***********************************************************************
        SUBROUTINE GETPHI(NP,NV,NSTD,STSIZE,COEFS,STDVAR,PHI)
        INTEGER G,NP,NV,NSTD,STSIZE(NP),SPLIT(0:8),CCOUNT(8)
        REAL CHECK,PHI(8,8),STDVAR(1000,20),COEFS(8,21),BIGEST,LDFVAL(8)
        SPLIT(0)=0
        DO 800 J=1,NP
  800     SPLIT(J)=SPLIT(J-1)+STSIZE(J)
C MAIN LOOP
        DO 1 J=1,NP
          DO 801 JJ=1,NP
  801       CCOUNT(JJ)=0
          DO 2 G=SPLIT(J-1)+1,SPLIT(J)
            DO 3 JJ=1,NP
              LDFVAL(JJ)=0.0
              DO 4 L=1,NV
```

```
4            LDFVAL(JJ)=LDFVAL(JJ)+COEFS(JJ,L)*STDVAR(G,L)
          LDFVAL(JJ)=LDFVAL(JJ)+COEFS(JJ,NV+1)
3       CONTINUE
        BIGEST=LDFVAL(1)
        MAX=1
        DO 5 JJ=2,NP
          IF(LDFVAL(JJ).GT.BIGEST) THEN
            MAX=JJ
            BIGEST=LDFVAL(JJ)
          END IF
5       CONTINUE
        CCOUNT(MAX)=CCOUNT(MAX)+1
2     CONTINUE
      CHECK=0.0
      DO 6 JJ=1,NP
        PHI(JJ,J)=FLOAT(CCOUNT(JJ))/FLOAT(STSIZE(J))
        CHECK=CHECK+PHI(JJ,J)
6     CONTINUE
      IF(CHECK.LE.0.9999.OR.CHECK.GE.1.0001) THEN
        PRINT *,'ERROR IN SUBROUTINE GETPHI.  PROGRAM STOPPED.'
        WRITE(6,700) CHECK
        STOP
      END IF
1   CONTINUE
    WRITE(6,707)
    DO 802 J=1,NP
802   WRITE(6,708) (PHI(J,JJ),JJ=1,NP)
    RETURN
707 FORMAT(////1X,'THE CLASSIFICATION MATRIX IS:'/)
708 FORMAT(1X,8F10.4)
700 FORMAT(///'OH NO - VALUE CHECK IS ',F8.4,' <>1.0'///)
    END
C*****************************************************************************
      SUBROUTINE READ9(NP,NV,NMIX,COEFS,X)
      INTEGER NP,NV,NMIX,COUNT,G
      REAL COEFS(8,21),X(3000,8),MIXVAR(20),LDFVAL(8),WORK
      DO 1 G=1,3001
        COUNT=G
        READ(9,*,END=2) (MIXVAR(L),L=1,NV)
        IF(G.GT.3000) THEN
          PRINT *,'STORAGE FOR MIXTURE EXCEEDED IN SUBROUTINE  READ9'
          STOP
        END IF
        DO 3 J=1,NP
          WORK=0.0
          DO 4 L=1,NV
4           WORK=WORK+COEFS(J,L)*MIXVAR(L)
          LDFVAL(J)=WORK+COEFS(J,NV+1)
3       CONTINUE
C IT IS ONLY THE RELATIVE SIZE OF THE LINEAR DISCRIMINANT FUNCTIONS
C THAT IS IMPORTANT SO WE CAN SET  LDFVAL(1) TO ZERO AND TRANSLATE
C THE OTHERS BY THE SAME AMOUNT.
C THE  X  MATRIX STORES THE 'LIKELIHOOD' VALUES THAT ARE REQUIRED BY
C THE DIRECT MAXIMUM LIKELIHOOD METHOD.
        DO 5 J=2,NP
          LDFVAL(J)=LDFVAL(J)-LDFVAL(1)
          X(G,J)=EXP(LDFVAL(J))
5       CONTINUE
        LDFVAL(1)=0.0
        X(G,1)=1.0
```

```
    1 CONTINUE
    2 IF(EOF(9).NE.0)  NOUGHT=0
      NMIX=COUNT-1
      WRITE(6,700)  NMIX
      RETURN
  700 FORMAT(//' THE MIXED SAMPLE IS OF SIZE ',I4)
      END
C*********************************************************************
      SUBROUTINE ANALYSE(NP,NMIX,PHI,X,THETA,POPNAM)
      INTEGER NP,NMIX
      REAL THETA(8,5),PHI(8,8),X(3000,8)
      CHARACTER POPNAM(NP)*6,TITLE*80
      DATA TITLE/'(1H1,31H TABLE OF COMPOSITION ESTIMATES)'/
      CALL COOKLRD(NP,NMIX,PHI,X,THETA)
      CALL MILLARC(NP,NMIX,THETA,PHI)
      CALL ML(NP,NMIX,THETA,X)
      CALL RESULTS(NP,THETA,POPNAM,TITLE)
      RETURN
      END
C*********************************************************************
      SUBROUTINE COOKLRD(NP,NMIX,PHI,X,THETA)
      INTEGER NP,NMIX,CCOUNT(8),G,MAX,EIGHT
      REAL PHI(8,8),THETA(8,5),A(8),SUM,BIGEST,ATOT,ATHTOT
      REAL X(3000,8)
      DOUBLE PRECISION PHIINV(8,8),DET
      DO 800 J=1,NP
  800   CCOUNT(J)=0.0
      DO 20 G=1,NMIX
        MAX=1
        BIGEST=X(G,1)
        DO 21 J=2,NP
          IF(X(G,J).GT.BIGEST)  THEN
            MAX=J
            BIGEST=X(G,J)
          END IF
   21   CONTINUE
        CCOUNT(MAX)=CCOUNT(MAX)+1
   20 CONTINUE
      DO 22 J=1,NP
   22   THETA(J,1)=FLOAT(CCOUNT(J))/FLOAT(NMIX)
C
C NOW DO COOK AND LORD CORRECTION
C
      EIGHT=8
      DO 801 J=1,NP
      DO 801 JJ=1,NP
  801   PHIINV(J,JJ)=DBLE(PHI(J,JJ))
      CALL MXINT(PHIINV,NP,EIGHT,DET)
      IF(DET.EQ.0.0)  THEN
        PRINT *,'THE CLASSIFICATION MATRIX IS SINGULAR!'
        PRINT *,'-NO UNIQUE COMPOSITION ESTIMATE IS POSSIBLE - ABORT'
        STOP
      END IF
      DO 1 J=1,NP
        SUM=0.0
        DO 2 JJ=1,NP
    2     SUM=SUM+PHIINV(J,JJ)*THETA(JJ,1)
        THETA(J,2)=SUM
    1 CONTINUE
C
```

```
C NOW CALCULATE CONSTRAINED COOK ESTIMATES.  THIS METHOD CAN
C REQUIRE MORE THAN ONE ITERATION, BUT NEVER MORE THAN NP-1
      DO 888 J=1,NP
  888   THETA(J,3)=THETA(J,2)
      DO 10 ITERS=1,NP
        DO 889 J=1,NP
  889     IF(THETA(J,3).LT.0.0) GO TO 9
        RETURN
    9   ATOT=0.0
        ATHTOT=0.0
        DO 11 J=1,NP
          IF(THETA(J,3).GT.0.0) THEN
            A(J)=1.0
          ELSE
            A(J)=0.0
          END IF
          ATOT=ATOT+A(J)
          THETA(J,3)=A(J)*THETA(J,3)
          ATHTOT=ATHTOT+THETA(J,3)
   11   CONTINUE
        DO 12 J=1,NP
   12     THETA(J,3)=THETA(J,3)-A(J)*(ATHTOT-1.0)/ATOT
   10 CONTINUE
      PRINT *,'FAULT WITH CODE FOR COOK''S CONSTRAINED ESTIMATE'
      WRITE(6,700) ITERS,A,((THETA(J,K),K=1,3),J=1,4)
  700 FORMAT(1X,'ITERS=',I2,'  A: ',4F4.1,4(/1X,'THETA: ',3F12.8))
      STOP
      END
C***********************************************************************
      SUBROUTINE MILLARC(NP,N,THETA,X)
C THIS SUBROUTINE OBTAINS THE MAXIMUM LIKELIHOOD CONSTRAINED CORRECTED
C ESTIMATES AS ADVOCATED BY MILLAR.  THE EM (EXPECTATION MAXIMIZATION)
C ALGORITHM IS USED.  SUBROUTINE  EM  BELOW IS THE SAME VERSION USED
C BY WDF TO DO DIRECT MAXIMUM LIKELIHOOD.  THE INPUT FOR  EM  IS
C CREATED BELOW - MUCH OF IT IS SETTING ALGORITHM SPECIFICATIONS.
      INTEGER NP,N,Y(8),IMAX,MAXG,NROWSX
      REAL THETA(8,5),THETA4(8),X(8,8),TOL
      CHARACTER OPT1*3,OPT5*3
      IMAX=100
C GIVE 'EVEN' STARTING VALUES FOR THE ESTIMATES
      DO 4 J=1,NP
    4   THETA4(J)=1.0/FLOAT(NP)
      TOL=1.0E-06
C OPT1 IS A PRINT OPTION.  IF YES THEN INTERMEDIATE RESULTS ARE PRINTED
C EVERY ITERATION.  IT WILL BE SET TO 'NO' BUT COULD BE ALLOWED AS A
C CONTROL FILE OPTION IF DESIRED.
      OPT1='NO'
C OPT5='YES' TURNS ON THE EM ACCELERATION SUBROUTINE
      OPT5='YES'
      MAXG=NP
C CALCULATE CLASSIFICATION FREQUENCIES FROM THE VECTOR (THETA(J,1)) OF
C RAW CLASSIFICATION PROPORTIONS.
      DO 1 J=1,NP
    1   Y(J)=NINT(THETA(J,1)*N)
C NROWSX  IS THE NUMBER OF ROWS IN  X  (THE CLASSIFICATION MATRIX) AS
C REQUIRED BY THE EM SUBROUTINE.
      NROWSX=8
C
C HERE GOES.....
C
```

```
      CALL EM(NP,IMAX,X,THETA4,TOL,OPT1,OPT5,MAXG,Y,N,NROWSX)
      DO 2 J=1,NP
    2   THETA(J,4)=THETA4(J)
      RETURN
      END
C***********************************************************************
      SUBROUTINE ML(NP,N,THETA,X)
C THIS SUBROUTINE OBTAINS THE DIRECT MAXIMUM LIKELIHOOD ESTIMATES.
C THE EM (EXPECTATION MAXIMIZATION) ALGORITHM IS USED.
C SUBROUTINE  EM  BELOW IS THE SAME VERSION USED BY WDF.
C THE INPUT FOR  EM  IS CREATED BELOW -
C MUCH OF IT IS SETTING ALGORITHM SPECIFICATIONS.
      INTEGER NP,N,Y(3000),IMAX,MAXG,NROWSX,G
      REAL THETA(8,5),THETA5(8),X(3000,8),TOL
      CHARACTER OPT1*3,OPT5*3
      IMAX=100
C GIVE 'EVEN' STARTING VALUES FOR THE ESTIMATES
      DO 4 J=1,NP
    4   THETA5(J)=1.0/FLOAT(NP)
      TOL=1.0E-06
C OPT1 IS A PRINT OPTION.  IF YES THEN INTERMEDIATE RESULTS ARE PRINTED
C EVERY ITERATION.
      OPT1='NO'
      OPT5='YES'
      MAXG=N
      DO 1 G=1,MAXG
    1   Y(G)=1
C
C HERE GOES.....
C
      NROWSX=3000
      CALL EM(NP,IMAX,X,THETA5,TOL,OPT1,OPT5,MAXG,Y,N,NROWSX)
      DO 2 J=1,NP
    2   THETA(J,5)=THETA5(J)
      RETURN
      END
C***********************************************************************
      SUBROUTINE RESULTS(NP,THETA,POPNAM,TITLE)
      INTEGER NP
      REAL THETA(8,5)
      CHARACTER POPNAM(NP)*6,TITLE*80
      WRITE(6,TITLE)
      WRITE(6,701)
      DO 888 J=1,NP
  888   WRITE(6,702) POPNAM(J),(THETA(J,K),K=1,5)
      WRITE(6,700)
      RETURN
  700 FORMAT(///)
  701 FORMAT(/,11X,'RAW',6X,'COOK & LORD',
     .6X,'COOK',8X,'MILLAR',6X,'MAXIMUM'/,
     .34X,'CONSTRAINED',2X,'CONSTRAINED',2X,'LIKELIHOOD'//)
  702 FORMAT(1X,A6,F8.4,4(5X,F8.4))
      END
C***********************************************************************
      SUBROUTINE MXINT(A,NRR,NRA,DET)
      DIMENSION ISW(96), A(*)
      DOUBLE PRECISION A,DET

      NA = NRA+1
      ND = NRR*NRA
```

```
      DO 1 I=1,NRR
    1 ISW(I) = I
      NDIAG = 1
      NUP = 1
      DET = 1.
      DO 9 I=1,NRR
         KA = I
         AM = 0.
         LA = NDIAG
         DO 2 J=I,NRR
            IF (DABS(A(LA)).LE.AM) GO TO 2
            KA = J
            AM = DABS(A(LA))
    2    LA = LA+1
         IF (AM.GT.1.0E-20) GO TO 3
         WRITE (6,14)
         DET = 0.
         RETURN
    3    IF (KA.EQ.I) GO TO 5
         MUP = I
         DO 4 MLOW=KA,ND,NRA
            S = A(MUP)
            A(MUP) = A(MLOW)
            A(MLOW) = S
    4    MUP = MUP+NRA
         IS = ISW(KA)
         ISW(KA) = ISW(I)
         ISW(I) = IS
    5    AA = A(NDIAG)
         DET = DET*AA
         DO 6 J=I,ND,NRA
    6    A(J) = A(J)/AA
         A(NDIAG) = 1./AA
         NEL = NUP
         DO 8 J=1,NRR
            IF (J.EQ.I) GO TO 8
            AA = A(NEL)
            A(NEL) = 0.
            KA = I
            DO 7 K=J,ND,NRA
               A(K) = A(K)-AA*A(KA)
    7       KA = KA+NRA
    8    NEL = NEL+1
         NUP = NUP+NRA
    9 NDIAG = NDIAG+NA
      I = 1
   10 IF (ISW(I).NE.I) GO TO 11
      I = I+1
      IF (I.GT.NRR) GO TO 13
   11 J = ISW(I)
      ISW(I) = ISW(J)
      ISW(J) = J
      MLOW = (I-1)*NRA+1
      MUP = (J-1)*NRA+1
      DO 12 K=1,NRR
         S = A(MLOW)
         A(MLOW) = A(MUP)
         A(MUP) = S
         MLOW = MLOW+1
   12 MUP = MUP+1
```

```
      GO TO 10

   13 RETURN
   14 FORMAT (1X,19HPIVOT ZERO IN MXINT)
      END
C***********************************************************************
      SUBROUTINE MXOUT(NP,V,UNIT)
      DIMENSION V(20,20)
      DOUBLE PRECISION V
      INTEGER UNIT

      DO 39 I=1,NP,7
         K=I+6
         IF(K.GT.NP)K=NP
         IF(UNIT.NE.2) THEN
            WRITE(UNIT,33)I,NP,I,K
            WRITE(UNIT,35) (L,L=I,K)
         END IF
         DO 37 J=I,NP
            L=J
            IF(J.LE.K)GO TO 1
            L=K
    1       WRITE(UNIT,36)J,(V(J,M),M=I,L)
   37    CONTINUE
   39    CONTINUE
   33 FORMAT(1H-,'(VARIABLES ',I2,1H-,I2,3H * ,I2,1H-,I2,1H))
   35 FORMAT(2H0 ,7I12)
   36 FORMAT(1H ,I3,7E12.6)
      RETURN
      END
C***********************************************************************
      SUBROUTINE MXMULT(N1,N2,A,B,C)
C CALCULATES  C = AB.  A  IS (N1*N2) AND  B IS (N2*N2)
      DIMENSION A(8,20),B(20,20),C(8,20)
      DOUBLE PRECISION B
      DO 1 I=1,N1
         DO 1 J=1,N2
            S=0.0
            DO 2 K=1,N2
    2          S=S+A(I,K)*B(K,J)
            C(I,J)=S
    1    CONTINUE
      RETURN
      END
C***********************************************************************
      SUBROUTINE EM(NP,IMAX,X,THETA,TOL,OPT1,OPT5,MAXG,Y,N,NROWS)
      REAL LAMBDA(3000),X(NROWS,8),THETA(NP),WORK(8),THETA1(8)
      REAL DIFF1(8),DIFF2(8)
      INTEGER G,Y(MAXG),NROWS,MAXG
      CHARACTER OPT1*3,OPT5*3,POPNAM(8)*6,FLAG
      DATA POPNAM/'POPN 1','POPN 2','POPN 3','POPN 4','POPN 5','POPN 6',
     .            'POPN 7','POPN 8'/
      LAG=1
      FLAG='N'
      DO 72 K=1,NP
   72    DIFF1(K)=0.0
      HOOD=0.0
      AA=1.0

      DO 1 ITER=0,IMAX
```

```
C THE VARIABLES  FLAG AND LAG  CONTROL THE ACCELERATION PROCESS.
C IF FLAG.EQ.'Y' IS TRUE THEN IT INDICATES A SUCCESSFUL ACCELERATION
C IN WHICH CASE  LAMBDA & HOOD  HAVE ALREADY BEEN CALCULATED IN
C SUBROUTINE ACCEL.
         IF(FLAG.EQ.'Y')GO TO 333
         OHOOD=HOOD
         HOOD=0.0
         DO 2 G=1,MAXG
            LAMBDA(G)=0.0
            DO 10 K=1,NP
               LAMBDA(G)=LAMBDA(G)+X(G,K)*THETA(K)
   10       CONTINUE
            IF(LAMBDA(G).LT.1.0E-12.OR.LAMBDA(G).GT.1.0E+12) THEN
              WRITE(6,98)ITER,G,LAMBDA(G)
            END IF
    2    CONTINUE
C TEST TO SEE WHETHER IT IS NECESSARY TO CALCULATE THE LOG LIKELIHOOD
C ON THIS ITERATION
         IF(AA.LT.TOL.OR.ITER.EQ.IMAX)GO TO 21
         IF(OPT5.EQ.'NO'.OR.LAG.GE.3.OR.ITER.LE.8)GO TO 334
   21    DO 22 G=1,MAXG
   22       HOOD=HOOD+Y(G)*ALOG(LAMBDA(G))
  333    IF(AA.LT.TOL.OR.ITER.EQ.IMAX)GO TO 900
C TEST TO SEE IF PRINTOUT REQUIRED EACH ITERATION
  334    IF(OPT1.EQ.'NO') GO TO 14
         WRITE(6,37)ITER
         IF(HOOD.NE.0.0) WRITE(6,38)HOOD
         WRITE(6,48)
         DO 29 K=1,NP
            WRITE(6,39)K,POPNAM(K),THETA(K)
   29    CONTINUE
         WRITE(6,40)AA

   14    DO 11 K=1,NP
   11       WORK(K)=0.0
         DO 3 G=1,MAXG
            QUOTI=Y(G)/LAMBDA(G)
            DO 3 K=1,NP
               WORK(K)=WORK(K)+QUOTI*X(G,K)
    3    CONTINUE
         DO 4 K=1,NP
            THETA1(K)=WORK(K)*THETA(K)/N
            DIFF2(K)=DIFF1(K)
            DIFF1(K)=THETA1(K)-THETA(K)
    4    CONTINUE
         AA=0.0
         DO 20 K=1,NP
            AA=AMAX1(AA,ABS(DIFF1(K)))
            THETA(K)=THETA1(K)
   20    CONTINUE
         IF(OPT5.EQ.'NO'.OR.ITER.LT.10)GO TO 1
         LAG=LAG-1
         FLAG='N'
         IF(LAG.NE.0)GO TO 1
         HOOD=HOOD+2*(HOOD-OHOOD)
         CALL ACCEL(NP,MAXG,DIFF2,DIFF1,THETA,HOOD,LAG,AA,X,LAMBDA,Y,
                   NROWS)
         IF(LAG.EQ.3) THEN
            IF(OPT1.EQ.'YES') WRITE(6,50) ITER
            FLAG='Y'
```

```
         END IF
         IF(LAG.EQ.10.AND.OPT1.EQ.'YES')WRITE(6,51)ITER
    1 CONTINUE
C.....................END OF EM ALGORITHM.............................
  900 IF(OPT1.NE.'YES') RETURN
      WRITE(6,37)ITER
      WRITE(6,38)HOOD
      WRITE(6,48)
      DO 30 K=1,NP
         WRITE(6,39) K,POPNAM(K),THETA(K)
   30 CONTINUE
      WRITE(6,40)AA
      IF(AA.LT.TOL) THEN
         WRITE(6,42)
      ELSE
         WRITE(6,41)
      END IF
      RETURN

   37 FORMAT(///1H ,8('=========='),//,1H ,'ESTIMATED COMPOSITION OF ',
     .'MIXED FISHERY AT ITERATION ',I4)
   38 FORMAT(//1H ,'THE LOGLIKELIHOOD IS ',F12.4)
   48 FORMAT(1H //,10X,10HPOPULATION,10X,12HCONTRIBUTION/)
   39 FORMAT(1H ,I6,5X,A6,13X,F9.6)
   40 FORMAT(//1H ,'CONVERGENCE ',F10.6)
   41 FORMAT(3(/1H ,'*****MAXIMUM NUMBER OF ITERATIONS REACHED*****'))
   42 FORMAT(3(/1H ,'*****CONVERGENCE TO SPECIFIED TOLERANCE******'))
   50 FORMAT(//1H ,'**********************************',
     .          /1H ,'*SUCCESSFUL ACCELERATION COMPLETED*  ITER=',I4,
     .          /1H ,'**********************************')
   51 FORMAT(//1H ,'************************************',
     .          /1H ,'*ACCELERATION ATTEMPTED BUT DECLINED*  ITER=',I4,
     .          /1H ,'************************************')
   98 FORMAT(1H ,'CAUTION: AT ITER ',I4,' LAMBDA(',I4,')=',E9.2)
      END
C**********************************************************************
      SUBROUTINE ACCEL(NP,MAXG,DIFF2,DIFF1,THETA,HOOD,LAG,AA,X,LAMBDA,Y,
     .                 NROWSX)
      REAL LAMBDA(MAXG)
      INTEGER G,NROWSX,Y(MAXG),MAXG
      DIMENSION DIFF2(NP),DIFF1(NP),THETA(NP),X(NROWSX,8),RATIO(8),
     .HALT(8),THETA1(8)
      ALPHA=100.0
      SUM=0.0
      DO 1 K=1,NP
         IF(ABS(DIFF1(K)).GE.AA/20) GO TO 2
         HALT(K)=0.0
         GO TO 1
    2    IF(DIFF1(K).LT.0.0)HALT(K)=-THETA(K)/DIFF1(K)
         IF(DIFF1(K).GT.0.0)HALT(K)=(1-THETA(K))/DIFF1(K)
         HALT(K)=INT(HALT(K))
    1 CONTINUE
      DO 3 K=1,NP
         IF(ABS(DIFF1(K)).LT.AA/20)GO TO 3
         RATIO(K)=DIFF1(K)/DIFF2(K)
         ALPHA=AMIN1(ALPHA,RATIO(K))
    3 CONTINUE
      IF(ALPHA.GT.0.75)GO TO 8
      LAG=10
      RETURN
```

```
    8 IF(ALPHA.EQ.1.0)ALPHA=0.99
      ALPHA=AMIN1(10.0,(1/(1-ALPHA))-1)
      DO 5 K=1,NP
         THETA1(K)=THETA(K)+AMIN1(HALT(K),ALPHA)*DIFF1(K)
         SUM=SUM+THETA1(K)
    5 CONTINUE
      DO 7 K=1,NP
    7    THETA1(K)=THETA1(K)/SUM
C CHECK TO SEE WHETHER ACCELERATION GIVES IMPROVEMENT IN LIKELIHOOD
      HOOD1=0.0
      DO 30 G=1,MAXG
         LAMBDA(G)=0.0
         DO 31 K=1,NP
   31       LAMBDA(G)=LAMBDA(G)+X(G,K)*THETA1(K)
         HOOD1=HOOD1+Y(G)*ALOG(LAMBDA(G))
   30 CONTINUE
      IF(HOOD1.GT.HOOD)GO TO 32
      LAG=10
      RETURN
   32 AA=0.0
      HOOD=HOOD1
      DO 33 K=1,NP
         AA=AMAX1(AA,ABS(THETA1(K)-THETA(K)))
         THETA(K)=THETA1(K)
   33 CONTINUE
      LAG=3
      RETURN
      END
C**********************************************************************
      SUBROUTINE STDDEVS(NP,N,PHI,X,THETA,POPNAM,STSIZE)
      INTEGER G,Y(3000),STSIZE(NP)
      REAL X(3000,8),THETA(8,5),LAMBDA(3000),SDEVS(8,5),PHI(8,8),
     .     S(8),T(8)
      CHARACTER POPNAM(NP)*6,TITLE*80
      DATA TITLE/'(25H STANDARD DEVIATION TABLE)'/
      WRITE(6,700)
C BEGIN WITH PELLA AND ROBERTSON STANDARD DEVIATIONS.
      CALL PANDR(NP,N,STSIZE,THETA,PHI,SDEVS)
C NOW DO MILLLARC INFINITESIMAL JACKKNIFE STANDARD DEVIATIONS.
      DO 1 J=1,NP
    1   Y(J)=NINT(THETA(J,1)*N)
      DO 2 J=1,NP
    2   T(J)=THETA(J,4)
      MAXG=NP
      NROWSX=8
      CALL IJACK(NP,MAXG,N,LAMBDA,T,Y,PHI,S,NROWSX)
      DO 3  J=1,NP
    3   SDEVS(J,4)=S(J)
C NOW DO MAXIMUM LIKELIHOOD INF JACK STD DEVS.
      DO 11 G=1,N
   11   Y(G)=1
      DO 12 J=1,NP
   12   T(J)=THETA(J,5)
      MAXG=N
      NROWSX=3000
      CALL IJACK(NP,MAXG,N,LAMBDA,T,Y,X,S,NROWSX)
      DO 13 J=1,NP
   13   SDEVS(J,5)=S(J)
      CALL RESULTS(NP,SDEVS,POPNAM,TITLE)
      RETURN
```

```
  700 FORMAT(//' PLEASE REMEMBER THAT THE STANDARD DEVIATION ESTIMATES',
     .' FOR THE COOK AND LORD'/' COMPOSITION ESTIMATES INCLUDE BOTH',
     .' BASELINE AND MIXED SAMPLE VARIABILITY.'/' THE OTHER ESTIMATES',
     .' ARE ABLE TO TAKE INTO ACCOUNT ONLY THE MIXED SAMPLE'/,
     .' VARIABILITY AND SO WILL BE SMALLER.'//)
      END
C***********************************************************************
      SUBROUTINE PANDR(NP,N,STSIZE,THETA,PHI,SDEVS)
      INTEGER NP,N,EIGHT,STSIZE(NP)
      REAL A,THETA(8,5),PHI(8,8),SDEVS(8,5),WORK(8)
      DOUBLE PRECISION PHIINV(8,8),SIGMA(8,8),DET
C WANT TO CALCULATE  (PHI INV)*SIGMA*((PHI INV) TRANSPOSE)
      EIGHT=8
      DO 1 J=1,NP
      DO 1 JJ=1,NP
    1   PHIINV(J,JJ)=DBLE(PHI(J,JJ))
      CALL MXINT(PHIINV,NP,EIGHT,DET)

C CALCULATE MIXTURE PART OF  SIGMA
      DO 2 J=1,NP
      DO 2 JJ=1,J
        SIGMA(J,JJ)=-THETA(J,1)*THETA(JJ,1)/N
        SIGMA(JJ,J)=SIGMA(J,JJ)
    2 CONTINUE
      DO 3 J=1,NP
    3   SIGMA(J,J)=THETA(J,1)/N + SIGMA(J,J)

C NOW ADD ON THE CONTRIBUTION FROM STANDARD VARIABILITY
      DO 801 K=1,NP
  801   WORK(K)=THETA(K,2)*THETA(K,2)/FLOAT(STSIZE(K))
      DO 4 J=1,NP
      DO 4 JJ=1,J
        DO 5 K=1,NP
    5     SIGMA(J,JJ)=SIGMA(J,JJ) - WORK(K)*PHI(J,K)*PHI(JJ,K)
        SIGMA(JJ,J)=SIGMA(J,JJ)
    4 CONTINUE
      DO 6 J=1,NP
        DO 7 K=1,NP
    7     SIGMA(J,J)=SIGMA(J,J) + WORK(K)*PHI(J,K)
    6 CONTINUE

      CALL MXMULT2(NP,PHIINV,SIGMA)
      DO 8 J=1,NP
      DO 8 JJ=1,J
        A=PHIINV(J,JJ)
        PHIINV(J,JJ)=PHIINV(JJ,J)
        PHIINV(JJ,J)=A
    8 CONTINUE
      CALL MXMULT2(NP,SIGMA,PHIINV)
      DO 20 J=1,NP
   20   SDEVS(J,2)=SQRT(PHIINV(J,J))
      RETURN
      END
C***********************************************************************
      SUBROUTINE IJACK(NP,MAXG,N,LAMBDA,THETA,Y,X,SDEVS,NROWSX)
      INTEGER G,EIGHT,Y(MAXG)
      REAL LAMBDA(MAXG),SDEVS(NP)
      DIMENSION THETA(NP),X(NROWSX,8)
      DOUBLE PRECISION DET,H(8,8),D(8,8)
C WANT TO COMPUTE  (H INVERSE)*D*((H INVERSE) TRANSPOSE)
```

```
      DO 10 G=1,MAXG
        LAMBDA(G)=0.0
        DO 10 K=1,NP
          LAMBDA(G)=LAMBDA(G)+X(G,K)*THETA(K)
   10 CONTINUE
C
C START BY COMPUTING  H
      DO 4 I=1,NP
         DO 5 K=1,I
            A=0.0
            DO 6 G=1,MAXG
               A=A+Y(G)*X(G,I)*X(G,K)/(LAMBDA(G)*LAMBDA(G))
    6       CONTINUE
            H(I,K)=A*THETA(I)
            H(K,I)=A*THETA(K)
    5    CONTINUE
         A=0.0
         DO 7 G=1,MAXG
            A=A+Y(G)*X(G,I)/LAMBDA(G)
    7    CONTINUE
         H(I,I)=H(I,I) + N-A
    4 CONTINUE
C
C NOW COMPUTE  D
C
      DO 1 I=1,NP
         DO 2 K=1,I
            A=0.0
            DO 3 G=1,MAXG
               A=A+Y(G)*(-1+X(G,I)/LAMBDA(G))*(-1+X(G,K)/LAMBDA(G))
    3       CONTINUE
            D(I,K)=A*THETA(I)*THETA(K)
            D(K,I)=D(I,K)
    2    CONTINUE
    1 CONTINUE
      EIGHT=8
      CALL MXINT(H,NP,EIGHT,DET)
      IF(DET.EQ.0.0)WRITE(6,999)

      CALL MXMULT2(NP,H,D)

      DO 8 I=1,NP
         DO 9 K=1,I
            A=H(I,K)
            H(I,K)=H(K,I)
            H(K,I)=A
    9    CONTINUE
    8 CONTINUE
      CALL MXMULT2(NP,D,H)
      DO 20 J=1,NP
   20   SDEVS(J)=SQRT(H(J,J))
      RETURN
  999 FORMAT(1H0,'*****H IS (NEAR) SINGULAR*****')
      END
C***********************************************************************
      SUBROUTINE MXMULT2(NP,A,B)
      DIMENSION A(8,8),B(8,8),C(8)
      DOUBLE PRECISION A,B,C
      DO 1 K=1,NP
         DO 2 I=1,NP
```

```
        S=0.0
        DO 3  J=1,NP
3          S=S+A(I,J)*B(J,K)
        C(I)=S
2     CONTINUE
      DO 4  J=1,NP
4     B(J,K)=C(J)
1 CONTINUE
  RETURN
  END
```